Development of the programming language learning skills using gamification elements

Taras Basyuk[1,*,†], Andrii Vasyliuk[1,*,†]

[1] *Lviv Polytechnic National University, Bandera str.12, 79013, Lviv, Ukraine*

Abstract

The article analyzes existing methods and approaches used in the process of developing programming language learning skills using gamification elements. It is shown that the application of gamification elements increases motivation, engagement, and productivity in solving technical tasks. A system analysis of the research object was conducted, resulting in the creation of a goal tree, whose primary objective is to develop a system for acquiring skills in learning programming languages. The next stage involved determining the functional requirements for the system, which were divided into the following categories: user management, educational materials and resources, assessment and progress, interactive features and communication, administrative functions, and technical requirements. Functional roles and responsibilities of different parties involved in the process were defined for the designed system. The subsequent stage involved designing the software system using an object-oriented approach and representing the created diagrams using UML language. The article presents use case diagrams, class diagrams, and activity diagrams, which provided the necessary apparatus for its further construction. The database schema of the designed system was presented using the MySQL Workbench software tool. An application software system for learning programming languages with gamification elements has been developed, which currently operates as a prototype.

Further research will focus on testing and improving the system, resolving conflicts, and expanding functionality in accordance with the defined requirements.

Keywords

programming language, learning, object-oriented design, gamification

1. Introduction

In the context of rapid digital technology development and significant integration of information systems into all areas of life [1], programming skills have become one of the most important asset for a successful career and personal development [2]. The main reasons for the relevance of this field are [3]: the growing demand for programmers (according to various studies, programming is one of the most sought-after skills in the labor market [4], necessitating effective training of new programmers and upskilling of existing professionals [5]), technological progress (modern information technologies offer new tools and methods for learning programming languages [6], making education more accessible and efficient [7]),

SCIA-2024: 3rd International Workshop on Social Communication and Information Activity in Digital Humanities, October 31, 2024, Lviv, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

Taras.M.Basyuk@lpnu.ua (T. Basyuk); Andrii.S.Vasyliuk@lpnu.ua (A. Vasyliuk)

0000-0003-0813-0785 (T. Basyuk); 0000-0002-3666-7232 (A. Vasyliuk)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

globalization and knowledge accessibility [8] (the internet and modern information technologies provide access to educational materials and resources from all over the world [9], opening new opportunities for self-learning and professional development regardless of location [10]), and continuous learning (programming is a constantly evolving field due to the emergence of new languages, frameworks, and tools, requiring continuous learning throughout one's career [11], and modern information technologies enable ongoing self-improvement [12]).

However, like with any form of education, knowledge gained through learning must be of a high quality. A crucial step in this process is selecting a mentor who can find a suitable educational approach for the learner and provide relevant materials. Since we live in the 21st century, in the world where technology prevails, the most convenient way of learning is through interactive systems, online manuals, video courses, and audiobooks. This entails searching for information portals, preferably free of charge or with a trial period, and exploring search engine's result pages [13] to obtain relevant content. However, data reliability is not always considered as well as its integrity, usefulness, and interactivity [14].

Analyzing the segment of student youth, conclusion can be made that the most popular type of interactivity today is the use of gamification elements [15]. This way, a platform can be created that combines learning with gameplay and provides users with a sense of competition [16]. This approach can improve the education level by encouraging students to complete assigned tasks in higher education institutions, allowing them to express their creative ideas and unconventional solutions [17]. Additionally, involving senior students can help implement and organize a continuous learning process, ensuring communication and knowledge exchange [18]. Given these trends, developing an information system for developing programming language learning skills using gamification approaches is a relevant task.

2. Related works

The growing demand for programmers and the rapid development of information technologies stimulate scientific research in the field of programming related education. The main areas of focus include:

- Open online courses - according to a study conducted by Harvard University, MOOCs [19] contribute to the democratization of education by making it accessible to a wide audience. MOOC platforms such as Coursera, edX, and Udacity allow users worldwide to learn from leading universities, providing access to high-quality educational materials and programming courses [20].
- Interactive platforms - research [21] has shown that interactive platforms like Codecademy and FreeCodeCamp significantly improve material retention and motivation [22]. They allow users to complete practical tasks in real-time, receive instant feedback, and track their progress, making learning more effective [23].
- Application of gamification methods - article [24] note that the use of gamification (e.g., platforms like CodeCombat and HackerRank) enhances student engagement and success [25]. Gamification, which includes game elements such as points, levels, and rewards, boosts motivation and makes the learning process more engaging [26].
- Use of simulations and virtual labs - a study [27] demonstrated that the use of simulations and virtual labs (e.g., Cisco Packet Tracer) provides practical experience

without the need for physical equipment, allowing users to experiment with code and configurations in a safe environment, leading to a deeper understanding of programming concepts.

- Personalized learning and adaptive learning systems - according to research [28], adaptive learning systems (e.g., Knewton and Smart Sparrow) offer a personalized approach to programming education by tailoring materials and tasks to individual user needs [29]. Personalized learning allows users to progress at their own pace, focusing on their weaknesses, thereby enhancing learning efficiency.
- Intelligent tutoring systems - research [30] has shown that intelligent tutoring systems (e.g., Python Tutor) can automatically provide recommendations and explanations that help learners understand complex programming concepts. Additionally, the use of intelligent tutoring systems enables individual support and feedback, improving programming skills.

Additionally, within the scope of the study, well-known software systems were analyzed, including the Ukrainian portal “Algotester”[31], and platforms “Codewars”[32], “HackerRank” [33], and “Exercism”[34]. The first three systems operate on the principle of a ready-made platform – these are websites that allow users to write code for a given task directly in the browser, compile the program, and check the result immediately. In contrast, the Exercism platform operates somewhat differently: users need to follow specific steps from the task instructions on the website to set up the environment locally on their computer. As a result, the compilation and checking occur on the user’s side rather than the system’s side. This provides expanded capabilities (tracking code execution, checking input and output data), offering more flexibility in task execution compared to real-time solutions.

To highlight the uniqueness of each system and identify issues such as imperfections and the lack of desired functionality, let’s take a closer look at each one.

“Algotester” system - also known as the “College of Algorithmic Programming,” it is characterized by the following features: site statistics, user guide, personal account, a set of various tasks, the ability to view the profile and statistics of another user, create a team, participate in regularly held tournaments, and access to classroom sessions at the Ivan Franko National University of Lviv. The main advantages are: multilingual support (English and Ukrainian); user rating table; a significant list of tasks; extended navigation instructions; task solution results table; ability to create your own tasks, and more. The disadvantages include the lack of solutions visualizations and tournament competitions, and the ranking table data loads as a single block rather than in parts, although it is visually presented as such.

System “Codewars” - this online platform allows amateurs and professional developers to participate in solving various programming tasks. These discrete coding exercises develop various programming language skills and allow to consolidate them in an integrated online development environment. Codewars was founded by Jake Hoffner and Nathan Doctor in November 2012. The advantages include: a wide variety of competitions; rating system; detailed instructions with examples; flexible selection of tasks according to skills and abilities; connection to social networks; the ability to compete with real users. The disadvantages of Codewars are: unstable mobile version of the site; long response time; if the task page was reloaded both code and data may be lost.

System “HackerRank” - this system is designed to create competitive tasks in the field of programming for both regular users and businesses, where developers compete by trying to program according to given specifications. The programming tasks from HackerRank can be solved using various programming languages, including Java, C++, PHP, Python, SQL, JavaScript, and cover different areas of information technology. The advantages of HackerRank are: a sufficient selection of competitions; rating system; detailed instructions with examples; connection to social networks; mobile version of the site; changes in the editor are saved even after the page is reloaded; quick system response to user requests. The disadvantages of HackerRank are: no “favorites” option if a task is liked; somewhat unstable mobile version of the site; significantly fewer programming languages to choose from compared to Codewars.

System “Exercism” - an open-source online programming platform that offers code practice and mentorship in seventy different programming languages. The website differs from other platforms by requiring users to download exercises via the command line client, solve the code on their computers, and then submit the solutions for feedback. Exercism’s codebase is open-sourced, consisting of dozens of repositories containing code for tasks in various programming languages. The advantages of Exercism are: wide selection of competitions; significant choice of programming languages to learn; detailed instructions with examples; stable mobile version of the site; quick system response to user requests. The disadvantages of Exercism are: no “favorites” option if a task is liked; no sparring option with users; no competitors rating table; integration with limited number of social networks.

The conducted analysis of known bodies of research indicate significant success in the programming language learning promotion. However, the lack of problem-oriented software solutions makes further study a relevant task.

2.1. Effectiveness of applying gamification methods

Gamification in education is a powerful tool for increasing motivation, engagement, and learning effectiveness, especially in the context of learning programming. The application of game mechanics such as points, badges, rankings, and competitions contributes to making learning more engaging and interesting for users, particularly young people who are accustomed to interactive and dynamic content.

During the development and testing of a system for teaching programming, similar software methodologies and existing empirical studies on the effectiveness of these methods were analyzed. One study involved 100 respondents who used the learning system for four weeks. The main metrics for evaluating effectiveness were: user engagement level (frequency of system logins, session duration), number of completed tasks, and students' self-assessment of learning motivation [35].

The study results showed that gamification significantly impacted user engagement. In particular, the use of rankings and competitions between users contributed to a 30% increase in the number of completed tasks compared to traditional teaching methods without gamification elements. Points and badges awarded for achieving certain levels of competence stimulated users to return to the system and continue performing tasks even after reaching initial goals. Leaderboards and competitions had the greatest impact on motivation. Participants who took part in competitions showed 40% higher productivity in solving tasks compared to those who studied individually. The psychological factor of competition and the ability to compare one's results with those of other users created an additional incentive to improve programming skills.

However, not all gamification elements proved equally effective for all users. For example, some participants expressed concern about excessive pressure that arises when performing tasks within time-limited competitions. This indicates the need for a careful balance between task complexity and motivational elements to avoid demotivating users, especially beginners [36].

Existing studies have also shown that the integration of gamification methods effectively supports long-term learning. Badges and levels that reflect user progress help maintain motivation throughout the learning period. Students who actively received these accolades completed an average of 25% more tasks compared to those who ignored these elements.

Thus, the results of existing studies confirmed that gamification has a significant positive impact on the process of learning programming, increasing both user engagement and their motivation to achieve learning goals.

2.2. Main research objectives and their significance

The goal of this study is to develop a web client for an educational portal with gamification elements, allowing users to try their hand at solving algorithmic problems. The conducted research will provide the means to create software based on it, incorporating game elements that enable users to grasp basic algorithms, approaches, and patterns taught in programming-oriented courses. Additionally, the created portal will include task instructions, a personal account to view progress, and tools for creating tournaments to evaluate users' programming skills level.

To achieve this goal, the following tasks would need to be accomplished: analysis of the existing approaches, research, and software tools used in developing programming skills; identification of the main challenges that arise in this process; system design using an object-oriented approach; prototype system implementation that enables programming skills development with the use of gamification elements.

The results of the study address a relevant scientific and practical problem of developing programming skills using modern information technologies.

3. Results and discussion

3.1. Application of the systems analysis methodology for studying the subject area

At the beginning of the research, a decision was made to employ the system analysis methodology for the subject area investigation. System analysis is a crucial approach for understanding, modeling, and optimizing complex systems, and one of its significant applications is the development of programming skills using modern information technologies. System analysis for this subject area can be divided into several stages [37].

Goal definition and requirements - establishing the final learning objectives, such as mastering a specific programming language, creating specific types of programs, or achieving a certain level of competence.

Analysis and modeling of learning - creating models of the educational process that consider various aspects such as teaching methods (online courses, interactive exercises, project-based learning), use of educational materials (books, videos, manuals), and feedback (tests, assignments, projects). An important step here involves using diagrams to model the educational process and identify potential improvements.

To implement the first stage, a goal tree was constructed to visualize and structure the tasks and sub-goals necessary to achieve the main objective. The goal tree of the designed system is presented in Figure 1.

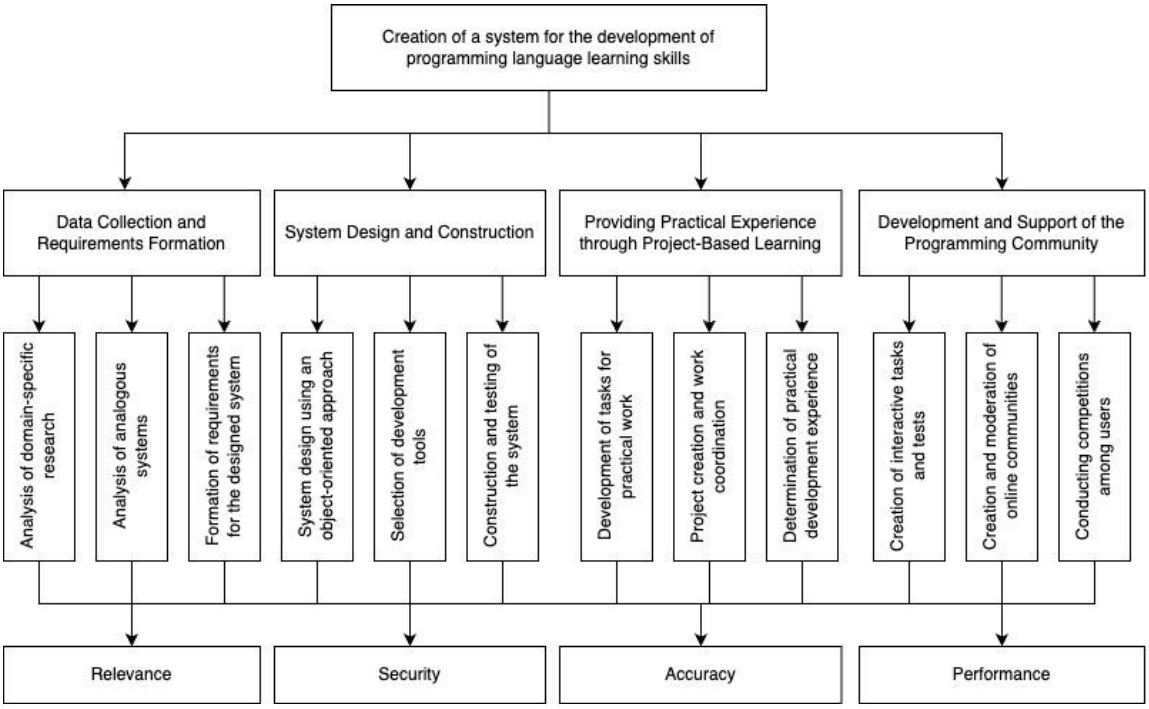

Creation of a system for the development of programming language learning skills
Data Collection and Requirements Formation
System Design and Construction
Providing Practical Experience through Project-Based Learning
Development and Support of the Programming Community
Analysis of domain-specific research
Analysis of analogous systems
Formation of requirements for the designed system
System design using an object-oriented approach
Selection of development tools
Construction and testing of the system
Development of tasks for practical work
Project creation and work coordination
Determination of practical development experience
Creation of interactive tasks and tests
Creation and moderation of online communities
Conducting competitions among users
Relevance
Security
Accuracy
Performance

Figure 1: Goal tree

Overall Goal is to create a system for developing programming language learning skills. To achieve this, several sub-goals need to be accomplished:

Data collection and requirements definition

- Analysis of research in the subject area - collecting and analyzing contemporary research on programming education methodologies is a crucial stage in understanding which methods are most effective. This subgoal involves studying scientific articles and various research findings. It is also important to pay attention to recent trends and innovations in programming education, such as the use of interactive platforms, gamification of learning processes, virtual and augmented reality, and so on. Identifying key aspects that impact the effectiveness of learning will help establish a foundation for further system development.
- Analysis of similar systems - this subgoal includes identifying existing programming education systems. Analyzing the strengths and weaknesses of these systems will help understand which elements are most effective and which need improvement. It is also important to study user feedback on these systems to determine their needs and

expectations. This will help avoid mistakes made by competitors and create a system that meets users' requirements to the maximum extent possible.

- Requirements definition for the designed system - based on the analysis of the subject area and analog systems, it is necessary to define the functional requirements for the designed system. This includes a list of functions that the system should perform, such as interactive lessons, automated assessment, feedback mechanisms, and others. It is also important to establish non-functional requirements such as security, performance, scalability, and system reliability. Technical specifications are then formulated based on these requirements, which will be used during the system development process.

System design and construction

- System design using object-oriented approach - involves developing a system based on objects that represent real-world entities. It includes creating a set of diagrams that relatively simply describe the characteristics required to build flexible and easily scalable systems.
- Selection of the development tools - at this stage, it's essential to choose the development tools that will be used to create the system. This includes selecting programming languages, frameworks, and database management systems. The choice should be based on the system requirements, its functionality, and scalability capabilities.
- System construction and testing - after selecting the development tools, the system construction process begins. It involves writing code, configuring database management systems, integrating different system components, and creating the user interface. Once development is complete, the system undergoes testing to identify and rectify errors. Additionally, it's crucial to conduct testing to ensure the system meets security, performance, and reliability requirements.

Providing practical experience through project-based learning

- Development of tasks for practical assignments - to ensure practical experience, it is necessary to develop tasks for practical assignments that encompass various aspects of programming. These tasks will include coding exercises, algorithm development, database work, and more. It is important that the tasks are diverse and cater to different levels of complexity, enabling users to progressively enhance their skills.
- Project creation and coordination of work - n addition to individual tasks, it is crucial to facilitate project work that encompasses all stages of software development. Coordinating project work helps in gaining skills to work in teams, utilize version control systems, and integrate various software components.
- Defining practical development experience - at this stage, it's important to determine the specific skills and knowledge that users should gain while working on practical tasks and projects. This may include programming, testing, database management, server configuration, working with APIs, code optimization, and more.

- Creating interactive exercises and tests - to support active learning, it is important to create interactive exercises and tests that help learners solidify their knowledge. These will include automated tests that assess theoretical knowledge and practical tasks that require coding. Interactive exercises should be diverse and engaging to maintain motivation for learning.
- Creating and moderating online communities - establishing online communities will facilitate knowledge sharing, asking questions, receiving assistance, and sharing achievements. In the designed system, this will include forums and chats. Moderating these communities will help maintain order, ensure safety, and encourage active participation from all members.
- Organizing user competitions - hosting competitions/tournaments among users will promote skill development, increase motivation, and foster friendly competition.

3.2. Defining functional requirements for the system

In accordance with the defined sub-goals in the formulated goal tree (Figure 1), the next stage of the research was to identify functional requirements. Functional requirements should encompass various aspects of the system's operation, ensuring effective learning and development of programming skills among users [38] (Table 1).

Table 1
Functional requirements of the system

Requirement	Description
1. User management	
user registration	the system should allow new users to register by providing necessary information such as name, email address, password, etc.
Authentication and authorization	The system should ensure secure user authentication and control access to various functions based on roles.
User profile management	Users should be able to view and edit their profiles, including personal information and account settings.
2. Educational materials and resources	
Knowledge base	The system should include various learning materials, such as text-based tasks, videos, interactive code examples, etc.
Categories and topics	Materials should be organized into categories and topics to simplify navigation and search.
Interactive tasks	The system should provide interactive exercises and tasks for practicing programming with the ability to check results.
3. Assessment and progress tracking	
Quizzes and tests	The system should provide the ability to take quizzes and tests to assess users' knowledge
Grades and scores	The system should track users' scores and ratings based on completed tasks and tests.

Progress reports	Users should have access to progress reports, including completed tasks, earned scores, and recommendations for further learning.
4. Interactive features and communication	
Forums and discussions	The system should have forums or other means for users to discuss topics and questions.
Chat	Real-time interactive communication for assistance and support.
Rating and feedback	Users should be able to leave feedback and rate educational materials and instructors.
5. Technical requirements	
Support for various programming languages	The system should support educational materials and exercises for popular programming languages (Python, Java, C++, JavaScript, etc.).
Data backup	The system should perform regular data backups to prevent data loss.
Security	The system should ensure a high level of security for user data, including protection against unauthorized access and breaches.

Defined functional requirements aim to create a comprehensive and effective system for developing programming skills using gamification elements. For the planned programming skills development system, it is important to define the functional roles and responsibilities of various stakeholders involved in the process. Below are the main roles and their responsibilities (Table 2).

Table 2
Functional roles and stakeholders responsibilities

Executor	Responsibilities	Client	Responsibilities
Backend developers	API, tests, tester's libraries	Customer	Provide clear system requirements, agree on deadlines
Frontend developers	Client-side development, tests, request manager	Tester	Verify the quality of work and functionality of the system, report defects, and document them
Database architect	Database design		
Project manager	Coordination of terms and tasks, communication with the client and team	Business Analyst	Define appropriate business logic aligned with market requirements, coordinate decisions, and assess the feasibility of assigned tasks
Technical specialist	Database design, API, tester's request manager, tester's libraries		

3.3. System design using object-oriented approach

The next step was to design the system in accordance with the object-oriented approach using the UML language [39]. The initial diagram in the design process was the use case diagram [40], depicted in Figure 2.

The main actors are: the user, the task administrator, the forum moderator and the solutions service. This diagram reflects the main purpose of the system - to provide high-quality tasks for practical training, to provide a full-fledged code editor, the ability to submit coded solution and receive feedback.

The following diagram is a class diagram used to visualize the structure of classes and their relationships. An extended description of the class diagram application includes the following aspects: requirements analysis (used to analyze system requirements, namely, understanding the structure of data and components allows you to analyze which classes are needed to implement the functionality of the system and how they are related to each other); communication with project participants (allows developers, architects, managers and other interested parties to understand the structure of the system and its components); identifying potential problems (can help identify potential problems and flaws in the system design, such as circular dependencies between classes, excessive complexity, or insufficient modularity); documentation (serves as an important tool of documenting the project, namely providing clear and specific information about the structure of the system for future developers and maintenance personnel).

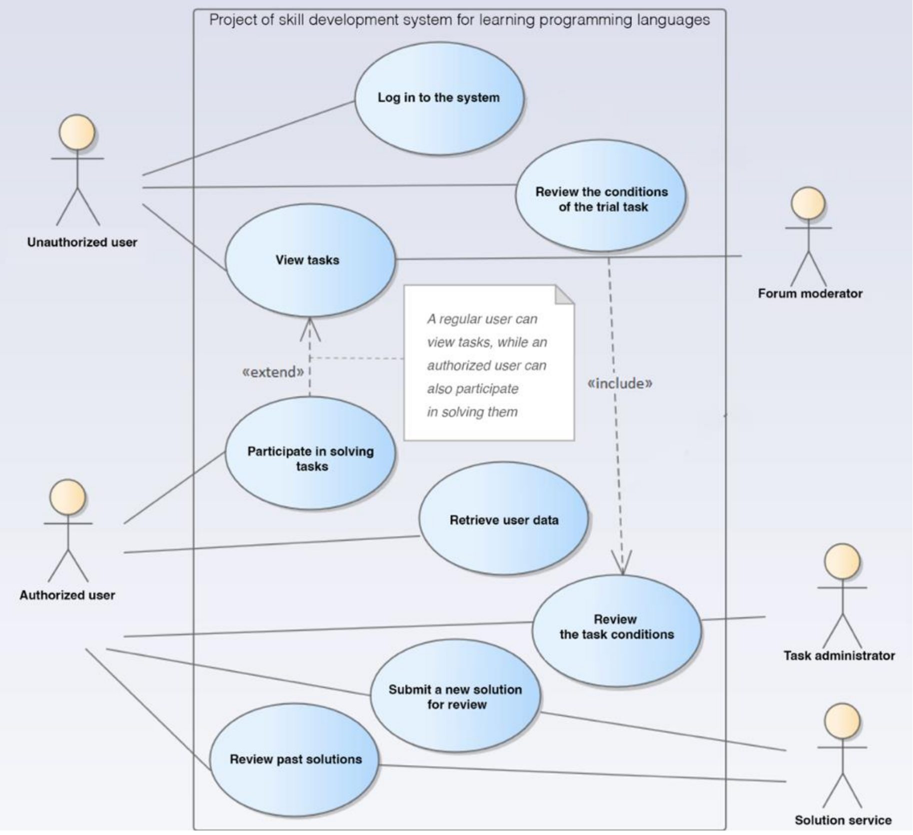

Project of skill development system for learning programming languages
Log in to the system
Review the conditions of the trial task
View tasks
Unauthorized user
Forum moderator
A regular user can view tasks, while an authorized user can also participate in solving them
«extend»
«include»
Participate in solving tasks
Retrieve user data
Authorized user
Review the task conditions
Task administrator
Submit a new solution for review
Review past solutions
Solution service

Figure 2: Use case diagram

In Figure 3. a class diagram of the system is presented.

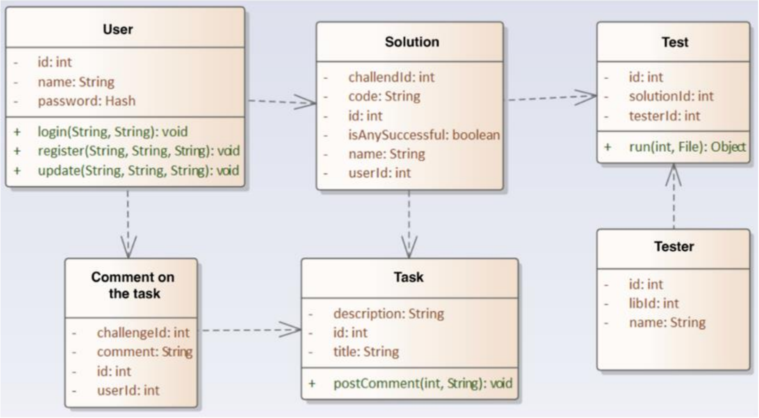


Figure 3: Class diagram

With the aim of illustrating the user authentication process within the system, an activity diagram was employed. This diagram serves as a valuable tool for developers, analysts, and other stakeholders to achieve the following objectives: visualization of the process (the diagram provides a clear representation of the steps sequence that comprises user authentication, helping with identification of the potential bottlenecks, delays, or inefficiencies in the flow), improvement in team's general project understanding (all project participants, including developers, analysts, and clients, can better comprehend the process), identification and resolution of issues (the diagram facilitates easy identification of potential issues or bottlenecks in the process), verification and testing (testers can use the diagram to create test cases that cover all aspects of the authentication process). The activity diagram for the user authentication process in the system is depicted in Figure 4.

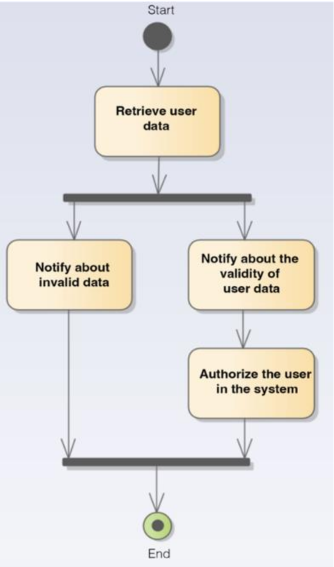

Start
Retrieve user data
Notify about invalid data
Notify about the validity of user data
Authorize the user in the system
End

Figure 4: Activity diagram for the user authentication process

The authentication process involves obtaining user-entered data and it's subsequent processing. At this stage, the system verifies the login and password, which are stored as hashed values in the data repository. If the entered data does not match, the user is notified of the incorrect input. Otherwise, an authentication token is generated and sent to the user to continue communication in a secure mode.

To implement a system for developing programming language skills, a data storage solution needs to be implemented. As a result of conceptual and logical design of the repository, a data model was developed using the software tool MySQL Workbench [41] (Figure 5).

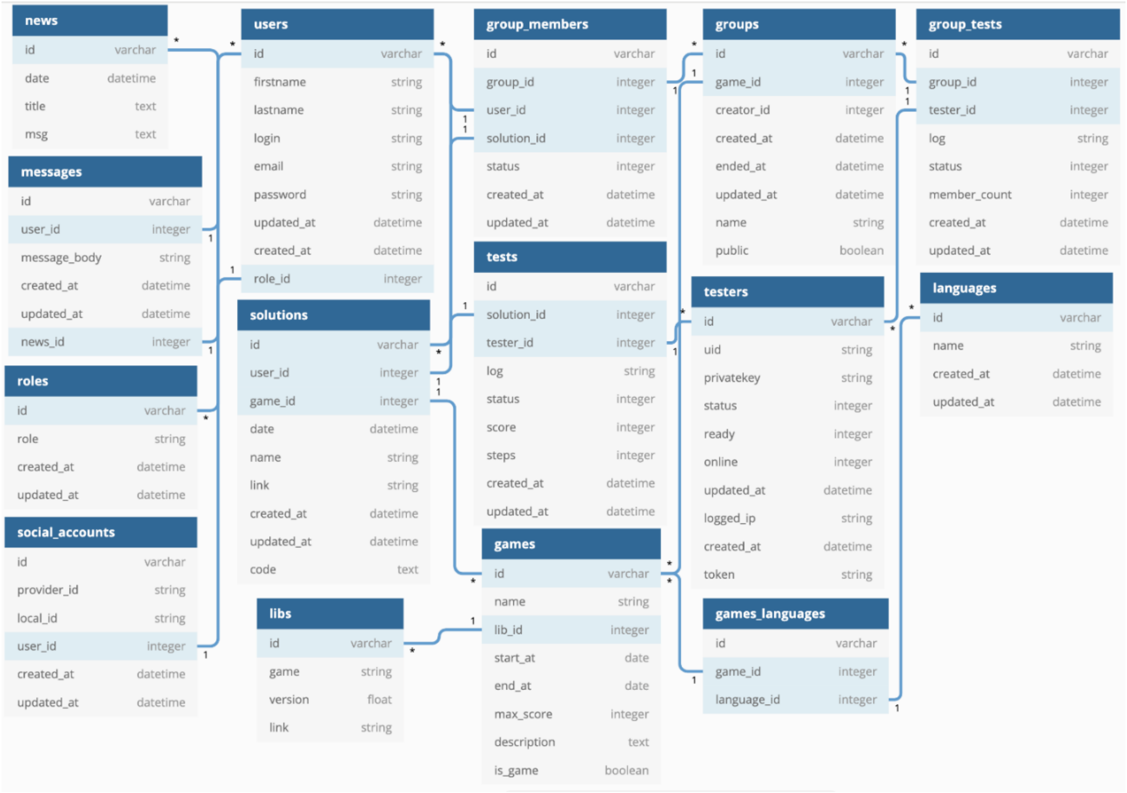


Figure 5: Database scheme of the system

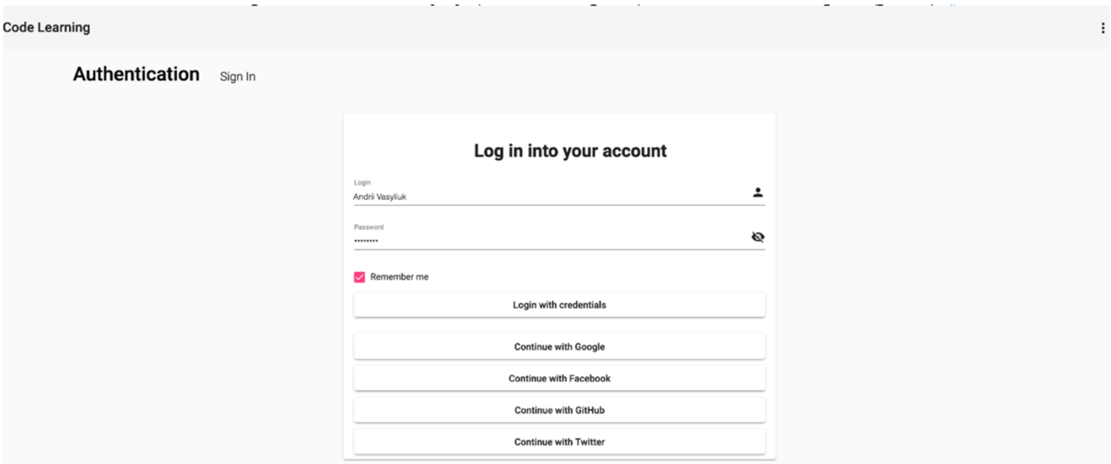
During the design of the gamified learning system, several challenges were encountered. One of the primary issues was ensuring that the gamification elements—such as points, badges, and leaderboards—were motivating without overwhelming or frustrating users. To address this, we adopted an iterative approach, testing different game mechanics with small user groups to balance difficulty levels. Another significant challenge was the integration of real-time feedback within the code editor. This required the development of custom APIs that could handle multiple languages, while maintaining low latency. We also faced scalability issues related to the system's performance when handling simultaneous users, which was mitigated by optimizing server-side processing and database queries. Finally, ensuring user engagement over time proved difficult, and this was addressed by introducing periodic competitions and collaborative challenges to sustain interest.

3.4. Construction of a system with gamification elements

Based on specified requirements and functional dependencies, a prototype of programming languages learning skills development system that uses gamification elements was implemented [42]. In the context of system development, gamification elements play a crucial role in enhancing learning efficiency and user's engagement. The main application methods for this system include:

- Utilization of levels and scores - allows users to track their progress and compare it with others, thereby stimulating task completion.
- Leaderboards - encourages competition and the desire to excel, thereby maintaining interest in continuous improvement.
- Competitions - regular challenges and competitions can be organized to foster competitiveness and test gained knowledge.
- Progressive stories and scenarios - educational materials are structured as progressive scenarios where the user acts as the main character, progressing through different stages and challenges.

To start using the system, users need to pass through the authentication stage, which can be done in two ways: through traditional access (username and password) and via social networks (Figure 6).



Code Learning

Authentication Sign In

Log in into your account

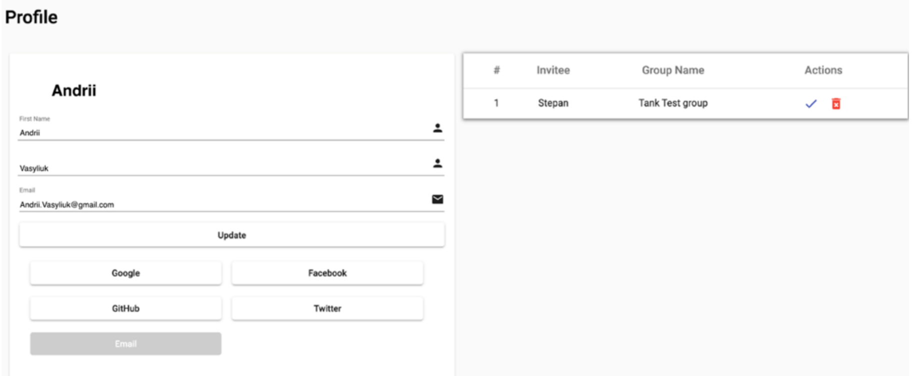
Login
Andrii Vasyliuk

Password
••••••••

☑ Remember me

Figure 6: Authentication page

User profile displays their editable data and a table of group invitations. Here, the user can accept or decline incoming invitations and optionally connect other social networks, even if they have registered via email (Figure 7).



Profile

Andrii

First Name: Andrii

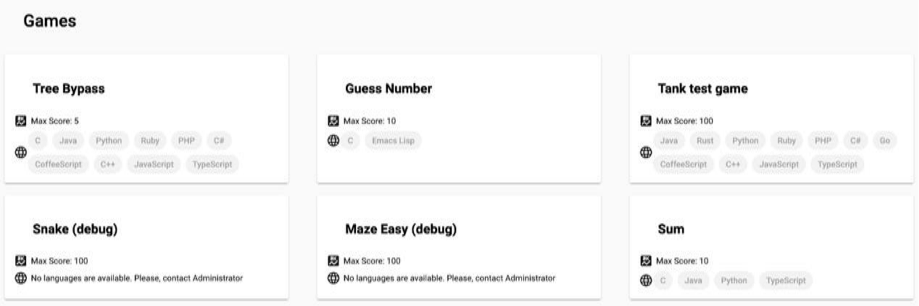
Vasyliuk

Email: Andrii.Vasyliuk@gmail.com

#	Invitee	Group Name	Actions
1	Stepan	Tank Test group	✓ 🗑

Figure 7: User profile with group invitations

Task components presented as gamified scenarios (Figure 8).



Games

Tree Bypass
Max Score: 5
C, Java, Python, Ruby, PHP, C#, CoffeeScript, C++, JavaScript, TypeScript

Guess Number
Max Score: 10
C, Emacs Lisp

Tank test game
Max Score: 100
Java, Rust, Python, Ruby, PHP, C#, Go, CoffeeScript, C++, JavaScript, TypeScript

Snake (debug)
Max Score: 100
No languages are available. Please, contact Administrator

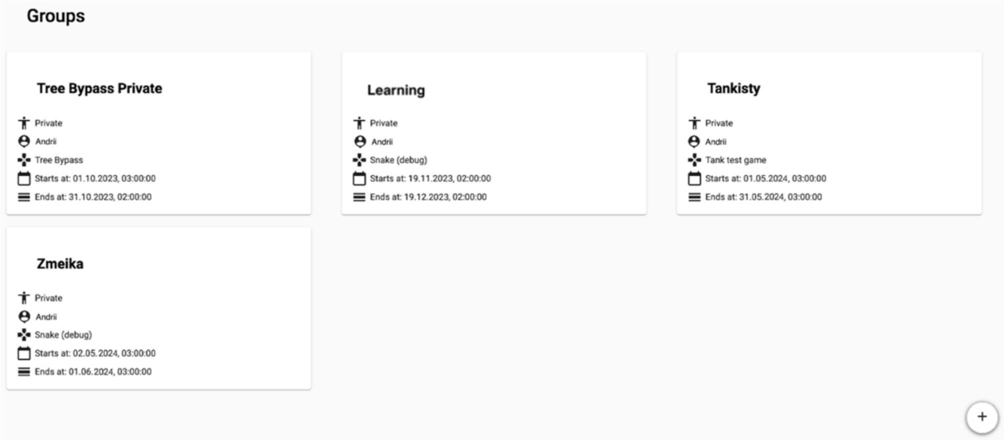
Maze Easy (debug)
Max Score: 100
No languages are available. Please, contact Administrator

Sum
Max Score: 10
C, Java, Python, TypeScript

Figure 8: Task list

After selecting the scenario, the corresponding code editor is launched, providing tools for coding according to the given task [43]. Users can choose a programming language from the available options [44] and also change the editor theme: light, dark, or high contrast [45].

A separate feature of the prototype is the ability to view the list of groups that a user belongs to or has created. Each item includes information such as the name, creator, game type, join period, and whether the group is private or public. In the case of private groups, the creator must manually add users since private groups are not visible by default (Figure 9).



Groups

Tree Bypass Private
Private
Andrii
Tree Bypass
Starts at: 01.10.2023, 03:00:00
Ends at: 31.10.2023, 02:00:00

Learning
Private
Andrii
Snake (debug)
Starts at: 19.11.2023, 02:00:00
Ends at: 19.12.2023, 02:00:00

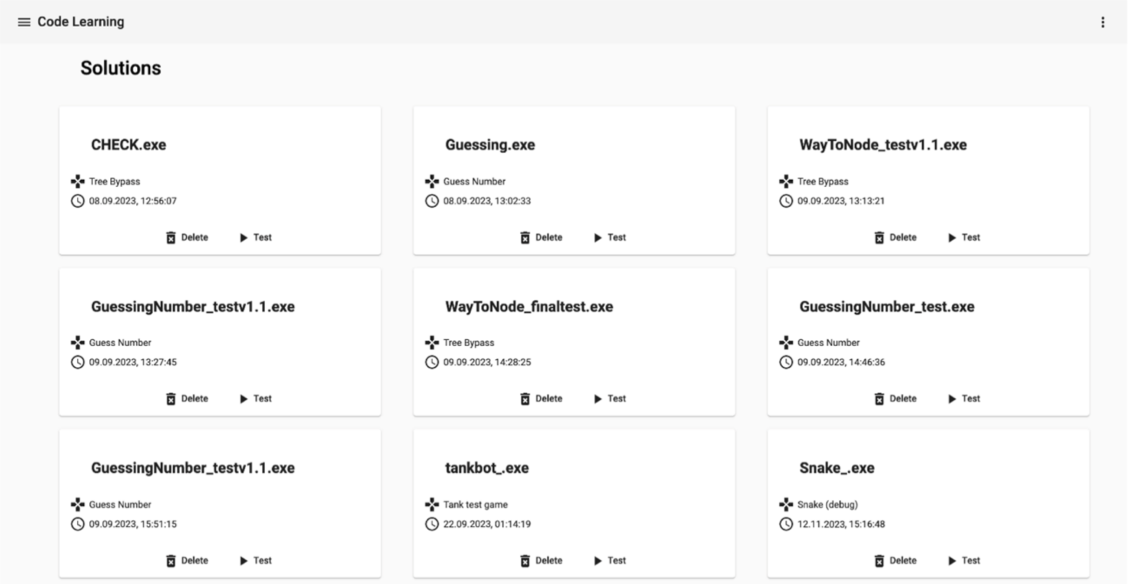
Tankisty
Private
Andrii
Tank test game
Starts at: 01.05.2024, 03:00:00
Ends at: 31.05.2024, 03:00:00

Zmeika
Private
Andrii
Snake (debug)
Starts at: 02.05.2024, 03:00:00
Ends at: 01.06.2024, 03:00:00

Figure 9: The user’s group list

In the constructed system, a section of solutions has been developed where users can manage their learning elements such as tasks, tests, and other educational materials. This section includes the following functional capabilities, such as deleting and launching tests (Figure 10):

- Deleting tasks - users can select a task from the list and click the 'Delete' button to remove it from the system.
- Launching tests - users can click the 'Start testing' button to initiate testing of the selected task.


Code Learning

Solutions

CHECK.exe
Tree Bypass
08.09.2023, 12:56:07
Delete ▶ Test

Guessing.exe
Guess Number
08.09.2023, 13:02:33
Delete ▶ Test

WayToNode_testv1.1.exe
Tree Bypass
09.09.2023, 13:13:21
Delete ▶ Test

GuessingNumber_testv1.1.exe
Guess Number
09.09.2023, 13:27:45
Delete ▶ Test

WayToNode_finaltest.exe
Tree Bypass
09.09.2023, 14:28:25
Delete ▶ Test

GuessingNumber_test.exe
Guess Number
09.09.2023, 14:46:36
Delete ▶ Test

GuessingNumber_testv1.1.exe
Guess Number
09.09.2023, 15:51:15
Delete ▶ Test

tankbot_.exe
Tank test game
22.09.2023, 01:14:19
Delete ▶ Test

Snake_.exe
Snake (debug)
12.11.2023, 15:16:48
Delete ▶ Test

Figure 10: List of solutions

The developed system for learning programming languages is flexible and versatile—it can be used for personal training as well as for learner's skills improvement. Additionally, the system provides unique tasks with processes visualization (games). When creating a game, the administrator has the discretion to choose from the available programming languages within the system that can be used to solve the task.

Currently, the system supports various programming languages, including Python, Java, C++, and JavaScript. However, the integration of gamification elements is planned to be adapted to the specific characteristics of each language in the future. In particular, for Python, gamified tasks are planned to be developed in a way that focuses on rapid problem-solving and iterations, as the simplicity of the language allows for faster code development and execution. Conversely, in C++-related gamification tasks, it is planned to add challenges related to memory management, pointer usage, and code efficiency optimization, which are key issues in mastering the language.

4. Conclusion

The relevance of developing programming language skills using gamification elements is driven by increasing demand for programmers, technological progress, global access to knowledge, innovative teaching methods, and the necessity for continuous learning. Modern information technologies significantly simplify the learning process, making it more effective and accessible to a wide range of people. Through research, existing methods and approaches used in programming language learning were analyzed. A systematic analysis of the research object was conducted, followed by the project requirements identification and analysis. Functional roles and responsibilities were defined and documented. The next stage involved designing the software system using an object-oriented approach and reflecting the created diagrams according to UML standards. A prototype of the application software system that implements the programming language learning process was developed. Currently, the software solution operates as a prototype.

Future research will focus on testing and refining the system, resolving conflicts, and expanding functionality in accordance with the defined requirements. Moreover, there are plans to focus on studying the long-term effects of gamification and their impact on the retention of programming skills and cognitive load. It is anticipated that an analysis will be conducted on how the motivational benefits of gamification translate into deeper learning and skill retention over time. A separate direction could be tracking users' progress over six months to one year, comparing their skill levels. If the results are satisfactory, it is planned to use the system in various educational contexts.

References

[1] F. Han, J. Cai, X. Wang, The development of computer communication technology and its application in electronic information engineering. In 2021 10th International Conference on Internet Computing for Science and Engineering (ICICSE 2021). Association for Computing Machinery, New York, NY, USA, 2021, pp.56–59. https://doi.org/10.1145/3485314.3485333.

[2] A. Dyro, Adaptive Learning In Education: The Next Gen Of Educational eContent, 2021. URL: https://elearningindustry.com/adaptive-learning-in-education-next-gen-econtent

[3] N. Dabbagh, R. Marra, J. Howland, Meaningful online learning: Integrating strategies, activities, and learning technologies for effective designs. Routledge.. TechTrends 64, 2020, pp. 931-933. https://doi.org/10.1007/s11528-020-00547-8.

[4] C. Simone, L. Howles, Designing the Online Learning Experience: Evidence-Based Principles and Strategies, Stylus Publishing, 2021. P.212.

[5] C. Petzold, Code: The Hidden Language of Computer Hardware and Software, Microsoft Press, 2022. P.480.

[6] N. Dabbagh, M. Marra, J. Howland, Meaningful Online Learning. 1st ed. Taylor and Francis, 2019. P.224.

[7] T. Basyuk, A. Vasyliuk, Peculiarities of an Information System Development for Studying Ukrainian Language and Carrying out an Emotional and Content Analysis // CEUR Workshop Proceedings. – 2023. – Vol. 3396: Computational Linguistics and Intelligent Systems 2023: Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. Volume II: Computational Linguistics Workshop, Kharkiv, Ukraine, April 20-21, 2023.pp. 279–294.

[8] M. Liu, D. Yu, Towards intelligent E-learning systems. Educ Inf Technol 28, 2023, pp.7845–7876. https://doi.org/10.1007/s10639-022-11479-6

[9] T. Green, L. Donovan, Learning anytime, anywhere through technology. In The Wiley Handbook of Teaching and Learning; Wiley: Hoboken, NJ, USA, 2018, pp. 225–256.

[10] T. Litmanen, I. Autio, Intelligent tutoring in online learning environment. In Proceedings of the 10th International Technology, Education and Development Conference, Valencia, Spain, 7–9 March 2016; pp. 6988–6995.

[11] H. El-Sabagh, Adaptive e-learning environment based on learning styles and its impact on development students' engagement. Int. J. Educ. Technol. High. Educ. 2021, pp.18-53.

[12] Y. Jing, L. Zhao, K. Zhu, H. Wang, C. Wang, Q. Xia, Research Landscape of Adaptive Learning in Education: A Bibliometric Study on Research Publications from 2000 to 2022. Sustainability 2023, Vol.15. https://doi.org/10.3390/su15043115.

[13] T. Basyuk, Innerlinking website pages and weight of links. Proceedings of the 12th International Scientific and Technical Conference «Computer Science and Information Technologies CSIT-2017». Lviv, Ukraine, September 12–15, 2017, pp. 12–15.

[14] S Linckels, K. Yves, R. Reuter, C. Dording, C. Weber, C. Meinel, Teaching with information and communication technologies: preliminary results of a large scale survey. In Proceedings of ACM SIGUCCS fall conference: communication and collaboration (SIGUCCS '19). Association for Computing Machinery, New York, NY, USA, 157–162. https://doi.org/10.1145/1629501.1629530

[15] A. Yasin, A. Abbas, Role of gamification in engineering education: A systematic literature review. IEEE Global Engineering Education Conference, EDUCON, 2021, pp. 210–213. https://doi.org/10.1109/EDUCON46332.2021.9454038

[16] J. Majuri, J. Koivisto, J. Hamari, Gamification of Education and Learning: A Review of Empirical Literature. In Proceedings of the 2nd International GamiFIN Conference, Pori, Finland, 21–23 May 2018, pp. 11–19.

[17] J. Swacha, State of Research on Gamification in Education: A Bibliometric Survey. Educ. Sci. 2021, Vol. 11. https://doi.org/10.3390/educsci11020069.

[18] M. Kalogiannakis, S. Papadakis, A. Zourmpakis, Gamification in Science Education. A Systematic Review of the Literature. Educ. Sci. 2021, Vol. 11. https://doi.org/10.3390/educsci11010022.

[19] J. Hansen, J. Reich, Democratizing education? Examining access and usage patterns in massive open online courses. Science, Vol. 350(6265), 2018, pp.1245–1248.

[20] R. Levin, R. How are people globally benefiting from online learning, 2017. URL: https://blog.coursera.org/online-learners-around-world-report-benefits-ranging-advancing-careers-discovering-field-study-gaining-confidence/2017.

[21] S. Wang, C. Christensen, W. Cui, R. Tong, L. Yarnall, L. Shear, M. Feng, When adaptive learning is effective learning: Comparison of an adaptive learning system to teacher-led instruction. Interact. Learn. Environ. 2020, Vol. 31, pp.793–803.

[22] H. Al-Chalabi, A. Hussein, U. Apoki, An Adaptive Learning System Based on Learner's Knowledge Level. In Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 1–3 July 2021; pp. 1–4.

[23] T. Basyuk, A. Vasyliuk, V. Lytvyn, O. Vlasenko, Features of designing and implementing an information system for studying and determining the level of foreign language proficiency// CEUR Workshop Proceedings. – 2023. – Vol. 3312: Modern Machine Learning Technologies and Data Science Workshop (MoMLeT&DS 2022): Proceedings of the Modern Machine Learning Technologies and Data Science Workshop, Leiden, The Netherlands, November 25-26, 2022. pp. 212-225.

[24] A. Ahmad, F. Zeshan, M. Khan, R. Marriam, A. Ali, The Impact of Gamification on Learning Outcomes of Computer Science. Claims ACM Transactions on Computing Education (TOCE), Volume 20, Issue 2, Article No.: 16, pp 1 - 25.

[25] R. Alsawaier, Research trends in the study of gamification. Int. J. Inf. Learn. Technol. 2019, Vol. 36, pp. 373–380.

[26] M. Arnold, B. North, H. Fischer, J. Mueller, M. Diab, Game-Based Learning in Vet Schools: A Learning Architecture for Educators in Vocational Education. INTED2021 Proceedings, 1 April 2021, pp. 3297–3303.

[27] I. Veza, A. Sule, N. Putra, M. Idris, I. Ghazali, P. Irianto, M. Ulka, A. Gipin, Virtual Laboratory for Engineering Education: Review of Virtual Laboratory for Students Learning. Engineering Science Letter. 2022, Vol. 1, pp. 41-46.

[28] P. Saraiva, R. Carmo, J. Gomes, V. João, V. Windson, Adaptive learning in computer science education: A scoping review. Education and Information Technologies. 2023, Vol. 29, pp.1-50.

[29] A. Vasyliuk, T. Basyuk, V. Lytvyn, Design and Implementation of a Ukrainian-Language Educational Platform for Learning Programming Languages// CEUR Workshop Proceedings. - 2023. - Vol. 3426: Modern Machine Learning Technologies and Data Science Workshop (MoMLeT&DS 2023): Proceedings of the Modern Machine Learning Technologies and Data Science Workshop, Lviv, Ukraine, June 3, 2023. pp. 406-420.

[30] T. Crow, A. Luxton-Reilly, B. Wünsche, Intelligent tutoring systems for programming education: a systematic review. The 20th Australasian Computing Education Conference, 2018, pp. 53-62. http://dx.doi.org/10.1145/3160489.3160492.

[31] Platform Algotester, 2024. URL: https://algotester.com/uk

[32] Codewars - Achieve mastery through coding practice and developers, 2023. URL: https://www.codewars.com/

[33] HackerRank - Online Coding Tests and Technical Interviews, 2024. URL: https://www.hackerrank.com/

[34] Platform Exercism, get really good at programming, 2024. URL: https://exercism.org/

[35] R. Huang, A. D. Ritzhaupt, Sommer, J. Zhu, A. Stephen, N. Valle, J. Li. The impact of gamification in educational settings on student learning outcomes: A meta-analysis. Educational Technology Research and Development, 68(4), 2020. pp.1875-1901. https://doi.org/10.1007/s11423-020-09807-z

[36] A. P. Markopoulos, A. Fragkou, P.D. Kasidiaris, & J.P. Davim. Gamification in engineering education and professional training. International Journal of Mechanical Engineering Education, 50(2), 2022. pp. 197-219. https://doi.org/10.1177/0306419015591324

[37] A. Dennis, B. Wixom, D. Tegarden, Systems Analysis and Design: An Object-Oriented Approach with UML, Wiley, 2020.

[38] J. Valacich, J. George, J. Hoffer, Modern Systems Analysis and Design, Pearson, 2020. P.521.

[39] A. Rababah, Assessing the Effectiveness of UML Models in Software System Development. International Journal of Applied Science and Research. 2024. pp.13-24. https:/doi.org/10.56293/IJASR.2024.5703

[40] B. Shamile, Software Development with UML Diagrams, Independently published. 2022. P.81.

[41] J. Murach, Murach's MySQL (3rd Edition), Mike Murach & Associates; 3rd edition, 2019. P.650.

[42] A. Behl, N. Jayawardena, V. Pereira, N. Islam, M. Giudice, J. Choudrie, Gamification and e-learning for young learners: A systematic literature review, bibliometric analysis, and future research agenda. Technological Forecasting and Social Change, 2022, vol. 176. http://dx.doi.org/10.1016/j.techfore.2021.121445

[43] B. Forouzan, R. Gilberg, C++ Programming: An Object-Oriented Approach. McGraw Hill; 1st edition, 2019. P. 960.

[44] C. Horstmann, Java: Advanced Features, Volume 2 (Oracle Press Java, Addison Wesley; 12th edition. 2022. P.1040.

[45] E. Norex, Mastering Dynamic Programming in Python. Independent Creating Platform. 2024. P.219.