Автоматизація технологічних і бізнес-процесів Volume 17, Issue 2/ 2025
http://www.atbp.ontu.edu.ua/
УДК 378.4:004.378.5

MANAGEMENT SYSTEMS AS A TOOL FOR DIGITAL TRANSFORMATION OF HIGHER EDUCATION INSTITUTIONS

СИСТЕМИ УПРАВЛІННЯ ЯК ІНСТРУМЕНТ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ

[1]Pavlo Lomovtsev, [2]Iurii Kornienko, [3]Svitlana Boltach, [4]Tamila Zihura, [5]Serhii Zubrytskii
[1]Павло Ломовцев, [2]Юрій Корнієнко, [3]Світлана Болтач, [4]Таміла Зігура, [5]Сергій Зубрицький
[1,2,3,4,5]Одеський національний технологічний університет, м. Одеса, Україна
[1,2,3,4,5]Odesa National University of Technology, Odesa, Ukraine
ORCID: [1]https://orcid.org/0000-0002-8145-9993, [2]https://orcid.org/0000-0002-0973-3927, [3]https://orcid.org/0000-0003-1902-2405
E-mail: [2]yurikkorn@gmail.com, [3]boltach.svetlana@gmail.com,

DOI: 10.15673/atbp.v17i2.3159

Анотація. *У сучасних умовах стрімкого розвитку інформаційно-комунікаційних технологій цифрова трансформація стає пріоритетним напрямом розвитку закладів вищої освіти (ЗВО). Управлінські системи, які включають програмні засоби для автоматизації адміністративних, навчальних та організаційних процесів, відіграють ключову роль у підвищенні ефективності функціонування ЗВО. Вони дозволяють інтегрувати різноманітні процеси, зокрема управління навчальними планами, кадровим потенціалом, фінансовими ресурсами, комунікацією зі студентами та зовнішніми партнерами, а також моніторинг якості освіти і акредитаційні процедури. У статті проведено аналіз основних переваг використання управлінських систем для цифрової трансформації, серед яких підвищення прозорості процесів, скорочення часу на адміністрування, покращення прийняття управлінських рішень на основі аналітичних даних, а також забезпечення гнучкості та адаптивності організаційної структури ЗВО. Окремо розглядаються виклики впровадження таких систем, зокрема необхідність підвищення кваліфікації персоналу, інтеграція з існуючими інформаційними ресурсами, питання безпеки даних та захисту інформації. Визначено, що стратегічне застосування управлінських систем сприяє не лише оптимізації внутрішніх процесів, а й підвищенню конкурентоспроможності та інноваційного потенціалу вищих навчальних закладів у контексті цифрової економіки. Запропоновано рекомендації щодо ефективної реалізації цифрової трансформації на основі сучасних управлінських систем, що дозволяють забезпечити сталий розвиток ЗВО, підвищення якості освітніх послуг та інтеграцію з глобальними освітніми трендами.*

Abstract. *In the modern context of rapid development of information and communication technologies, digital transformation has become a priority direction for the development of higher education institutions (HEIs). Management systems, which include software tools for automating administrative, educational, and organizational processes, play a crucial role in enhancing the operational efficiency of HEIs. These systems enable the integration of diverse processes such as curriculum management, human resource administration, financial resource allocation, communication with students and external partners, as well as monitoring education quality and accreditation procedures. The article analyzes the primary advantages of using management systems for digital transformation, including increased process transparency, reduction of administrative time, improved decision-making based on data analytics, and ensuring organizational flexibility and adaptability. Challenges in implementing these systems are also addressed, including the need for staff training, integration with existing information resources, data security, and information protection issues. It is concluded that the strategic application of management systems contributes not only to optimizing internal processes but also to enhancing the competitiveness and innovative potential of higher education institutions in the context of the digital economy. Recommendations are provided for the effective implementation of digital transformation based on modern management systems, which enable sustainable development of HEIs, improvement in the quality of educational services, and integration with global educational trends.*

Key words: automated design, software development, software life cycle, modeling, information and analytical activities, analysis

Ключові слова: автоматизоване проєктування, розробка програмного забезпечення, життєвий цикл програмного забезпечення, моделювання, інформаційно-аналітична діяльність, аналіз.

Introduction

The COVID-19 pandemic has profoundly disrupted educational systems worldwide, compelling institutions to adopt rapid and extensive digital transformations. Traditional face-to-face instruction was abruptly suspended due to stringent quarantine measures aimed at safeguarding public health, necessitating an urgent transition to remote, online learning modalities. This shift not only altered pedagogical practices but also accelerated the integration of automation technologies within education, marking a pivotal moment in the ongoing digitalization of academic environments.

Automation in education encompasses the deployment of advanced digital tools and information systems designed to optimize administrative workflows, enhance instructional delivery, and improve data-driven decision-making processes. Education Management Information Systems (EMIS), in particular, have emerged as critical infrastructure, facilitating efficient data collection, analysis, and management across various educational functions such as admissions, student progress monitoring, and curriculum development.

Despite significant advancements, disparities remain in the adoption and sophistication of such systems globally. According to recent UNESCO reports, a substantial proportion of countries continue to rely on traditional paper-based methods, while others have embraced online platforms to varying degrees, revealing a heterogeneous landscape of digital maturity. In Ukraine, these challenges are compounded by the ongoing conflict, which imposes additional barriers to consistent access and operational stability within educational institutions.

Moreover, effective implementation of automated systems faces obstacles including limited digital literacy among educators, suboptimal user interface design, and organizational resistance. Addressing these factors is essential to harness the full potential of automation as a means to enhance educational quality and resilience.

This article aims to explore the multifaceted role of automation in transforming educational management amidst crisis conditions, with a focus on the Ukrainian context. It investigates technological, pedagogical, and administrative dimensions, proposing strategies to overcome existing barriers and promote sustainable digital integration in higher education.

Theoretical part

During the COVID-19 pandemic, strict quarantine restrictions were introduced worldwide, as even routine daily activities posed a threat to public health and safety. This situation necessitated a radical transformation in the operation of educational institutions, which were forced to suspend traditional in-person instruction and switch to a remote (online) learning format. This transition triggered a large-scale digital transformation in the education sector, marking the onset of what is now referred to as the “automation era.”

In the educational context, automation involves the implementation of modern digital technologies to optimize administrative and learning processes, enhance efficiency, reduce routine workloads, and enable flexible interaction among stakeholders. With the help of automated tools, educators can focus on delivering high-quality learning content, while students can better manage their time and academic performance.

Education Management Information Systems (EMIS) have become a key component of the digital infrastructure of educational institutions. These systems facilitate the collection, analysis, and use of data to support admission processes, monitor student progress, administer academic programs, and inform decision-making. According to a UNESCO report [1], more than half of countries (53%) still rely on traditional paper-based methods for educational data management, while 51% use online interfaces, and 36% apply other forms of technology. This highlights the uneven implementation of digital solutions and the significant potential for further development in many countries.

In Ukraine, the development and implementation of such systems remains crucial even beyond the pandemic, particularly under the conditions of ongoing war. Educational institutions face challenges such as temporary suspension of in-person learning, relocation of teachers and students, and disruptions in electricity supply. Under such conditions, digital solutions become vital not only for ensuring the continuity of the educational process but also for maintaining stability in other critical areas of public life.

However, several barriers hinder the effective adoption of automated systems. Many existing platforms are burdened by complex logic, non-intuitive interfaces, or inadequate functionality. Limited digital literacy among educators and administrative staff often results in a preference for outdated practices such as paper documentation, manual data entry, or complete avoidance of digital tools.

An additional limitation is the insufficient attention paid to user experience in system design. To ensure effective interaction, systems must meet the needs of not only teachers but also students and prospective applicants. Applying UX/UI design principles — such as clear structural hierarchy, minimalist layout, and mobile-friendly navigation — significantly improves usability. In many cases, interfaces created solely for desktops perform poorly on mobile devices, complicating the completion of even basic tasks.

Moreover, institutional resistance is a major obstacle to digitalization. Without administrative support, it is difficult to secure resources, provide staff training, and ensure the systematic integration of new technologies. As a result, automation is implemented inconsistently, reducing its effectiveness and creating gaps between system components.

The successful implementation of automated educational platforms requires a comprehensive approach — from technical excellence to organizational commitment. It is essential to ensure tool accessibility, interface usability, user-centered design, data security, and continuous digital training for staff and students. Only under these conditions will automation serve not as a burden, but as a catalyst for qualitative improvements in education.

Practical part

At all stages of software product development, modern tools are employed to ensure efficiency, productivity, and high quality of the final deliverable. From design through testing and deployment, specialized environments facilitate the work of developers and designers, enable rapid iteration, and guarantee seamless integration among system components.

One such tool is the web-based application Figma. Figma's feature set focuses on user interface (UI) and user experience (UX) design, providing vector graphic editors and prototyping utilities [2]. A key advantage of Figma is its emphasis on real-time collaboration, allowing all project participants to work concurrently on designs, leave comments, and provide feedback without delay. Figma operates within a cloud environment where the latest versions of drafts are stored online, making design review and editing accessible from any internet-connected device. The application offers not only a web version but also desktop and mobile clients. Additionally, Figma supports an extensive library of plugins and fonts, and allows creation of reusable components and style guides. Its capability to export layouts in various raster and vector formats, alongside interoperability with other platforms such as Sketch, makes it indispensable in modern web application development workflows.

The primary tool for any software developer is the integrated development environment (IDE). Contemporary IDEs typically provide functionalities for code editing, compiling, deployment, and debugging. For this project, WebStorm by JetBrains will be utilized. WebStorm is a development environment tailored for JavaScript/TypeScript, HTML, and CSS, with support for major frameworks and libraries in web development [3]. It extends standard IDE capabilities with features such as intelligent code completion, real-time error detection, debugging tools, and testing utilities. Furthermore, WebStorm integrates version control systems including Git, GitHub, and GitLab, facilitating effective team collaboration, change tracking, and rollback capabilities. The environment can be customized to individual developer preferences and workflows. Code editing is streamlined by a user-friendly editor, hotkey support, and efficient file navigation.

Thus, leveraging Figma for design and WebStorm for code creation and testing significantly enhances project productivity. Combining these tools in front-end development promotes a more structured, rapid, and error-resistant workflow, ensuring consistency between design specifications and implementation.

React, as previously mentioned, is employed for building web applications. However, client-side rendering inherent to React can pose challenges, such as security vulnerabilities, SEO limitations, and incompatibility with browsers that disable JavaScript. To address these issues, the Next.js framework was developed [4]. Next.js enables server-side rendering, generating HTML content prior to delivery to the client browser, thereby improving SEO by allowing search engines to index fully rendered pages and accelerating initial page load performance. Another benefit of Next.js is automatic routing: within the main app directory, each subfolder corresponds to a route accessible via a specific URL. A route requires a layout.js file (defining page structure) or page.js file (defining page content). Next.js also supports dynamic and parallel routes. The framework optimizes standard HTML tags—for example, replacing the <img /> tag with the <Image /> component from next/image ensures consistent image quality across devices and prevents layout shifts during loading. Similarly, the <Link /> component improves navigation over the conventional <a /> tag by enabling client-side routing. Overall, Next.js contributes to faster, more secure, and user- and SEO-friendly websites.

Modern websites are primarily evaluated by users on visual appeal. The CSS language has long been the foundation for styling web content. However, with evolving requirements, traditional methods often fall short, prompting the creation of numerous libraries, frameworks, and preprocessors. Tailwind CSS is one such modern utility-first CSS framework [5]. Its core advantage lies in using utility classes directly in HTML, obviating the need for separate style declarations. Instead of defining separate classes for components and writing CSS rules elsewhere, developers apply pre-built utility classes that perform specific styling functions.

Projects typically incorporate auxiliary libraries that facilitate development, maintenance, and scalability. ESLint is a JavaScript/TypeScript linter that detects potential errors and enforces code style conventions. Prettier complements this by automatically formatting code to maintain stylistic consistency. Axios is a lightweight library for performing HTTP requests and interacting with APIs asynchronously, supporting error handling and request interception. TanStack offers a suite of libraries addressing common complex development challenges: TanStack Query manages server state synchronization, TanStack Form enables creation of typed, flexible, and validated forms, and TanStack Table provides customizable data tables with filtering, pagination, and sorting capabilities.

A noted limitation of Tailwind CSS is the repetition of similar utility classes. This issue is mitigated by daisyUI, a plugin that builds on Tailwind's utilities to define common component classes such as btn, link, and input for corresponding UI elements.

Together, these libraries form a cohesive and robust development ecosystem that enhances user experience and facilitates project scalability and error detection for developers.

Prior to product development, key project requirements are identified, typically divided into functional and non-functional categories. Functional requirements specify what the system must do, while non-functional requirements describe the qualities the system must exhibit. Table 1 presents the primary client demands identified for this project.

Table 1. Main Requirements for the User Interface

Functional Requirements	Non-Functional Requirements
Should have a convenient, logical, and intuitive layout of elements	Must display correctly in all modern browsers
Must support display on various devices	Must comply with WCAG accessibility standards
The site should include interactive elements	The structure should be easy to update, extend, and maintain
Separate forms should be created for registration/login	Page load and rendering times should be minimal
Separate CRUD interface for customer relationship management	

Another method for defining requirements is the creation of a use case diagram. This approach allows for a clear identification of the capabilities and actions of both regular users and administrators. These users act as actors in the diagram who interact with the system. Figure 1 illustrates the general principle of website usage from a student's perspective.

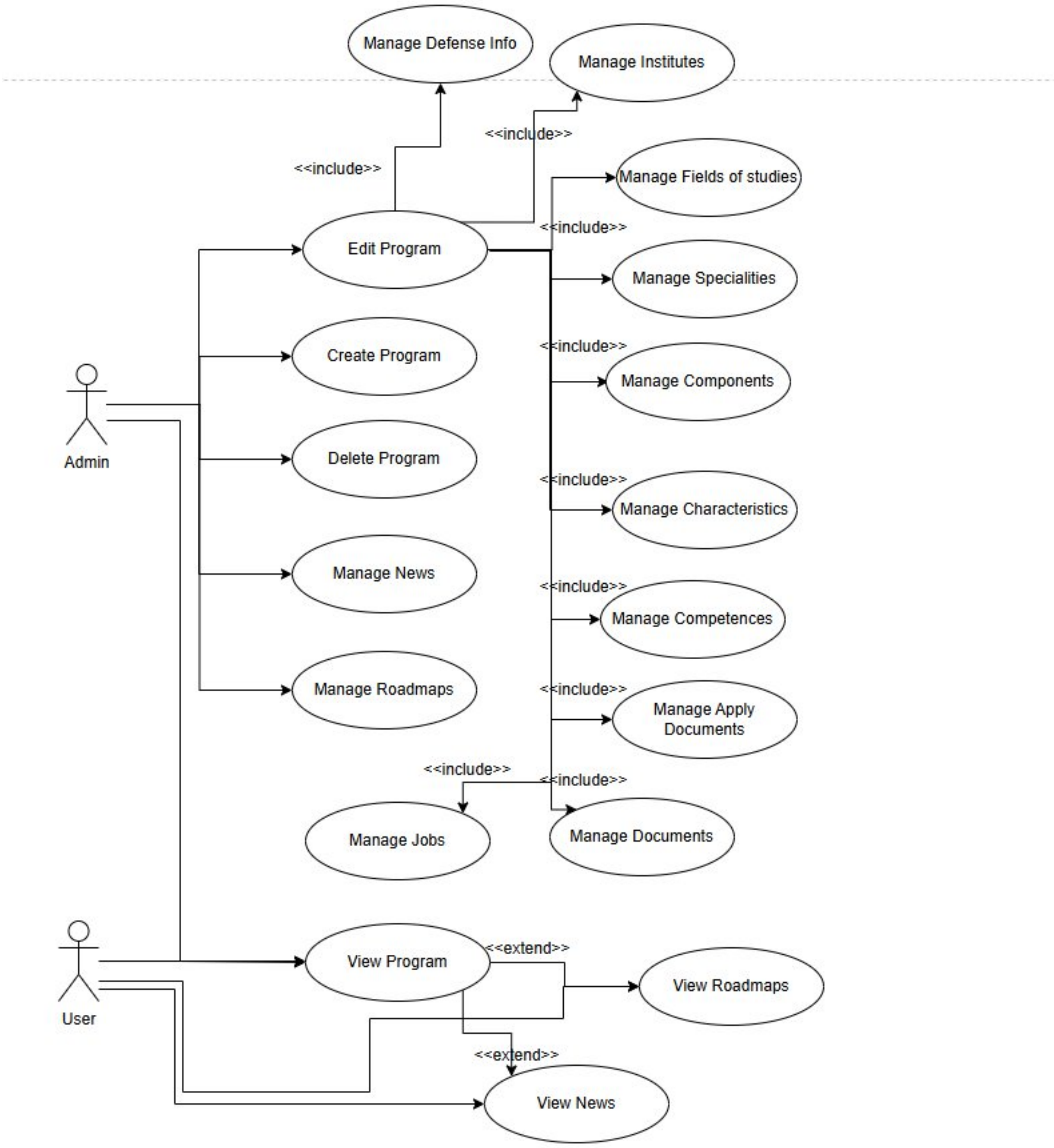

Fig. 1 – Use Case Diagram

The system consists of client-side and server-side components. The client-side is responsible for rendering information received from the server. Regular users have access to all web pages with navigation capabilities between them. Administrators are granted additional access to a control panel, through which any publicly available data can be modified. The authorization module is partially implemented on the client-side as a login form that interacts with the server to verify user identity.

The server-side handles data storage, processes requests from the client, manages data operations, and ensures secure interactions between users and the database. A REST API functions as an intermediary between the server and client. All retrieval, modification, addition, or deletion of data from the database is performed via API requests.

A distinct module manages authentication and authorization. In this project, token-based methods were employed for implementation. This approach uses two tokens: an access token and a refresh token. Both tokens are stored in HTTP-only cookies, which are inaccessible to JavaScript, thereby providing protection against cross-site scripting (XSS) attacks. The refresh token has a long expiration period, whereas the access token is short-lived and is used to reauthorize the user through the refresh token.

Conclusion

The development of the system integrates modern client-server architecture, leveraging a clear separation of responsibilities to optimize performance, security, and usability. The client-side ensures intuitive information presentation and user navigation, while administrators are empowered with advanced control through a dedicated management panel. The server-side robustly handles data storage, request processing, and security via RESTful API endpoints, facilitating seamless interaction with the database.

Implementing token-based authentication and authorization mechanisms with HTTP-only cookies significantly enhances system security by mitigating common web vulnerabilities such as XSS attacks. The use of both access and refresh tokens balances security and user experience by enabling short-lived access credentials complemented by longer-lived refresh tokens.

Overall, this architecture and security approach create a scalable, secure, and user-friendly environment that meets both functional and non-functional requirements, ensuring reliable operation under varying user roles and scenarios.

Список використаних джерел

1. UNESCO. (2023). *The State of Education Management Information Systems (EMIS)*. [Online]. Available: https://www.unesco.org/en/articles/state-education-management-information-systems-emis-2023
2. Wikipedia – Figma [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Figma
3. WebStorm: The JavaScript and TypeScript IDE, by JetBrains [Електронний ресурс]. – Режим доступу: https://www.jetbrains.com/webstorm/
4. Next.js [Електронний ресурс]. – Режим доступу: https://nextjs.org/docs
5. Tailwind CSS [Електронний ресурс]. – Режим доступу: https://tailwindcss.com/docs

Reference

1. UNESCO. (2023). The state of education management information systems (EMIS). https://www.unesco.org/en/articles/state-education-management-information-systems-emis-2023
2. Wikipedia contributors. (n.d.). Figma. Wikipedia. Retrieved July 15, 2025, from https://en.wikipedia.org/wiki/Figma
3. JetBrains. (n.d.). WebStorm: The JavaScript and TypeScript IDE. Retrieved July 15, 2025, from https://www.jetbrains.com/webstorm/
4. Vercel. (n.d.). Next.js documentation. Retrieved July 15, 2025, from https://nextjs.org/docs
5. Tailwind Labs. (n.d.). Tailwind CSS documentation. Retrieved July 15, 2025, from https://tailwindcss.com/docs

Отримана в редакції 20.05.2025. Прийнята до друку 10.06.2025. Received 21 May 2025. Approved 10 June 2025. Available in Internet 30 June 2025