

УДК 004.42

doi:10.20998/2413-4295.2023.01.05

ПРОЕКТУВАННЯ ТА РОЗРОБКА ЕЛЕКТРОННОГО КАБІНЕТУ СТУДЕНТА ДЛЯ ВИКОРИСТАННЯ НА РІЗНИХ ОПЕРАЦІЙНИХ СИСТЕМАХ

В. О. МІТЕЛІОВ*, О. М. МАРУСЕНКО, М. Д. ВЛАСОВ

кафедра комп'ютерного моделювання процесів та систем, НТУ «ХПІ», Харків, УКРАЇНА
*e-mail: volodymyr.mietielov@khp.edu.ua

АНОТАЦІЯ Спроектовано, розроблено та реалізовано кросплатформений застосунок для користування кабінетом студента будь-де та можливості отримати інформацію щодо змін у розкладі або повідомлення від викладача миттєво. Це дає змогу кожному студенту швидше та зручніше корегувати свій графік сумісно із розкладом занять. Усі користувачі отримують доступ до застосунку за своїми авторизаційними даними, що дозволяє захистити дані користувачів від крадіжок та несанкціонованого доступу. Кросплатформений застосунок надає інформаційну базу, яка відображається у вигляді вікон із новинами, розкладом, повідомленнями, заліковою книжкою. Виконана інтеграція інформації, що передбачає синхронізацію даних у різних програмних застосунках і її консолідацію для подальшої обробки людиною або автоматизованими системами. Як правило, корпоративна інформація представлена у структурованій і неструктурованій формі, причому друга перевищує першу у 3-4 рази і завдання її консолідації часто вкрай важливе. Інтеграцію даних у неструктурованій формі здійснюють системи управління корпоративним контентом і системи управління знаннями. Для повноцінного функціонування застосунку на виділеному сервері розміщено API-частину застосунку та розповсюджено застосунок через магазини застосунків (PlayStore, AppStore). Застосунок може бути використаний на різних операційних системах, оскільки фреймворк React Native є кросплатформеним. При розробці додатку була використана операційна система Windows 10, мова програмування JavaScript та її фреймворки: React Native та NodeJS. У результаті роботи було створено систему, яка виконує описаний функціонал та забезпечує безпеку даних всіх користувачів. Цей застосунок дозволяє студенту переглядати поточний розклад, перевірити персональні дані, обрати необхідну йому дисципліну навчання або побачити поточні оцінки за семестр. Все це робить застосунок необхідним для щоденного користування, аби завжди бути у потоці подій, що відбуваються у навчальному закладі. Швидкість відображення інформації при використанні кросплатформеного застосунку значно вища, аніж перегляд цих даних через браузер користувача.

Ключові слова: операційні системи; кросплатформений застосунок; кабінет студента; React Native; JavaScript; NodeJS; API

DESIGN AND DEVELOPMENT OF THE STUDENT'S ELECTRONIC CABINET FOR USE ON DIFFERENT OPERATING SYSTEMS

V. MIETIELOV*, O. MARUSENKO, M. VLASOV

Department of Computer Modeling of Processes and Systems, NTU "KhPI," Kharkiv, UKRAINE

ABSTRACT In the work designed, developed and implemented a cross-platform application for using the student's office anywhere and being able to receive information about changes in the schedule or a message from the teacher instantly. This allows each student to adjust their schedule more quickly and more conveniently in accordance with the class schedule. All users get access to the application according to their authorization data, which allows you to protect user data from theft and unauthorized access. The cross-platform application provides an information base that is displayed in the form of windows with news, schedule, messages, scorebook. The integration of information, which involves the synchronization of data in various software applications and its consolidation for further processing by a person or automated systems, has been carried out. As a rule, corporate information is presented in structured and unstructured form, and the second exceeds the first by 3-4 times, and the task of its consolidation is often extremely important. Data integration in an unstructured form is carried out by corporate content management systems and knowledge management systems. For the full functioning of the application, the API part of the application is placed on a dedicated server, and the application is distributed through application stores (PlayStore, AppStore). The application can be used on different operating systems, since the React Native framework is cross-platform. The application was developed using the Windows 10 operating system, the JavaScript programming language and its frameworks: React Native and NodeJS. As a result of the work, a system was created that performs the described functionality and ensures data security for all users. This application allows the student to view the current schedule, check personal data, choose the discipline of study he needs or see the current grades for the semester. All this makes the application necessary for daily use in order to always be in the flow of events happening in the educational institution. The speed of displaying information when using a cross-platform application is much higher than viewing this data through the user's browser.

Keywords: operating systems; cross-platform application; student's office; React Native; JavaScript; NodeJS; API

Вступ

Натепер світова спільнота вступила у нову фазу свого розвитку, яка визначається неймовірно швидким зростанням попиту до інформаційних ресурсів у самих різних сферах діяльності. Рівень розвитку інформаційних ресурсів та характер їх використання у значній мірі впливає на добробут і

безпеку суспільства у цілому, а також його окремих елементів, включаючи державні структури.

Інформація стає таким самим стратегічним ресурсом, як і традиційні – матеріальні та енергетичні. Головні позиції будь-якої організації визначаються у першу чергу її можливостями по розумному використанню передових досягнень у сфері інформаційних технологій. Особливу роль при

цьому відіграє сфера освіти, де формується інтелектуальний потенціал країни, який споживає і створює її інформаційні ресурси.

Застосунок був реалізований задля покращення життя студентів та викладачів. Даний програмний продукт буде розвиватися і надалі для втілення побажань користувачів та введення нових можливостей.

Людині XXI віку важко уявити своє життя без високотехнологічних пристроїв, як-то смартфони, планшетні ПК, бо саме ці науково-технічні новинки дозволяють не тільки підтримувати зв'язок з людьми із різних куточків світу, а й допомагають у формуванні та розвитку сучасного інформаційного простору.

Сфера ІТ прогресує кожного дня. Розробки у сфері кросплатформених застосунків стають дедалі різноманітнішими [1–7], бо навіть побутова техніка (телевізори, холодильники тощо) зараз керується під різними операційними системами. Сьогодні за допомогою смартфона можна не тільки робити високоякісні зображення, а й переглянути інформацію стосовно будь-кого або будь-чого (сторінки у соціальних мережах, блогах тощо).

Наявність сучасних технологій дає нам змогу не тільки полегшити життя, а ще й економить наш час. Інформація та дані все частіше розглядаються як життєво важливі ресурси, які повинні бути організовані таким чином, щоб ними можна було легко користуватися. Саме тому розробка нового інформаційного застосунку для студентів є досить актуальною. Наявність збалансованого розширення для мобільних пристроїв може стати у нагоді не тільки студентам, а й університетам, які зможуть простіше доносити необхідну інформацію [8–11].

Мета роботи

Мета цієї роботи – проектування, розробка та реалізація мобільного застосунку для користування кабінетом студента будь-де та мати змогу отримати інформацію щодо змін у розкладі або повідомлення від викладача миттєво. Це дає змогу кожному студенту швидше та зручніше корегувати свій графік сумісно із розкладом занять.

Головною задачею було розробити зручний та надійний кросплатформений застосунок для користування студентами та викладачами. Надалі цей застосунок зможе стати невід'ємною частиною студентського життя та дозволить спростити перші роки навчання і спілкування із викладачами.

При розробці застосунку була використана операційна система MS Windows 10, мова програмування JavaScript та її фреймворки: React-Native [12–14] та NodeJS [15].

Застосунок може бути використаний на різних операційних системах, оскільки фреймворк React-Native є кросплатформеним.

Виклад основного матеріалу

При запуску застосунку головний клас App формує його інтерфейс за допомогою виклику функції render.

При натисканні кнопки «Sing in» викликається метод `signInLogIn`, який відмальовує клас `signInLogIn` у якому відображається вікно входу до облікового запису.

На сторінці входу до облікового запису студента працює кнопка «Забули пароль» – «Forgot your password?» (рис. 1), натиснувши яку користувач має змогу відправити запит до ІТ-відділу університету на поновлення даних для входу.

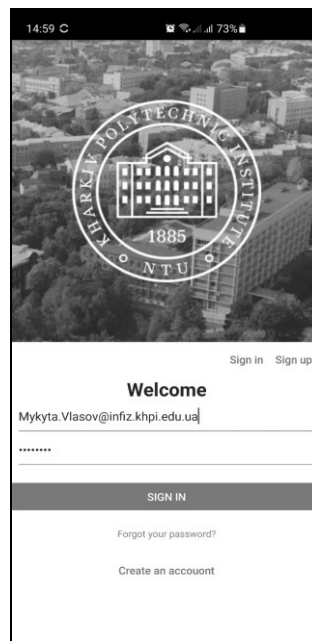


Рис. 1 – Вікно входу до застосунку

При натисканні кнопки відбувається процес аутентифікації – відправлення запиту до захищеного сховища даних для перевірки введених даних та очікування відповіді у застосунку. Якщо дані вірні, користувач переходить до головної сторінки з новинами, а якщо дані невірні, пропонується перевірити введені дані та спробувати ще раз.

Користувач має можливість вибрати один із п'яти основних пунктів меню (головна сторінка новин, розклад занять, повідомлення, залікова книжка та профіль студента) та перейти до кожного з них.

На сторінці новин користувач може або ж відкрити новину для детального перегляду, або вибрати інший пункт меню. Панель з головними сторінками відображається на будь-якій сторінці. Також користувач може викликати бокове меню із додатковими сторінками, просто витягнувши його з лівого боку екрану свого пристрою (рис. 2).

Для переходу на сторінку із розкладом необхідно натиснути другу кнопку нижнього меню (рис. 3). На сторінці розкладу студент може вибрати один із п'яти днів та побачити поточний розклад, що оновлюється при зміні. Надалі можна натиснути на аудиторію та побачити на карті де саме розташовується корпус із необхідною аудиторією.

Якщо користувачу надійшло повідомлення від деканату, він може його побачити, натиснувши третю

кнопку нижнього меню, після чого застосунок відкриє сторінку із повідомленнями (рис. 4).

На сторінці повідомлень студент може лише побачити та прочитати повідомлення, що надійшли до нього.

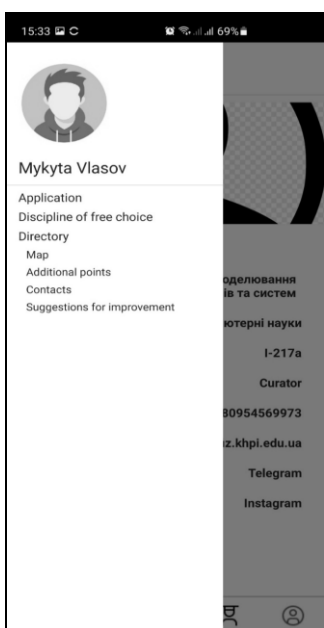


Рис. 2 – Бокове меню із додатковими пунктами



Рис. 3 – Розклад занять студента

Якщо натиснути четверту кнопку, у застосунку відкриється сторінка залікової книжки (рис. 5). Відображення залікової книжки та усі необхідні дані отримуються із серверу та є актуальними для кожного студента на момент відображення. Користувач може вибрати які саме оцінки відображати у його кабінеті, від 1 до 5 курсу (1 або 2 семестр).

При натисканні на п'яту кнопку відобразиться особистий профіль студента. На сторінці особистого профіля студент може змінити деякі налаштування, такі як соціальні мережі та номер телефона, а також може змінити фото свого профіля.

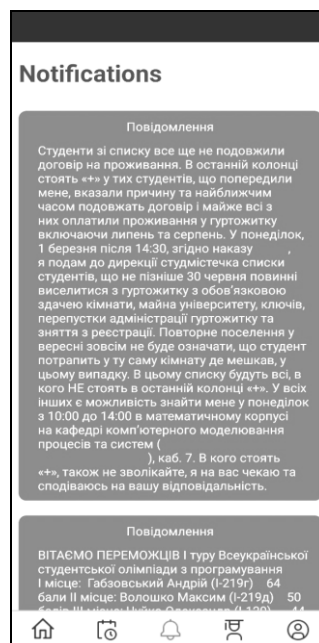


Рис. 4 – Сторінка повідомлень для студента

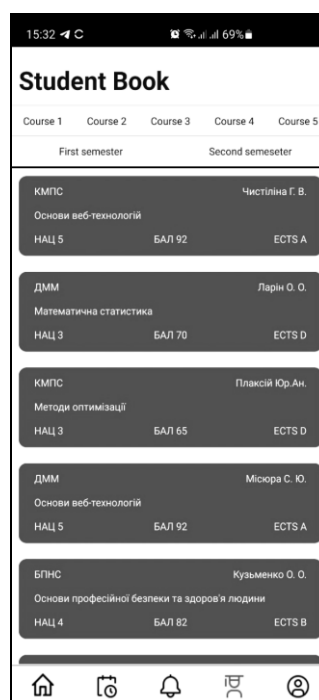


Рис. 5 – Залікова книжка студента

У боковому меню студент може натиснути на кнопку вибору дисциплін вільного вибору та отримати детальну інформацію про вподобані ним дисципліни (рис. 6).

При натисканні на обрану програму студент отримує детальну інформацію про курс із даними про викладача, його тривалість та навчальну програму.

Якщо студенту необхідно замовити довідку у деканаті, він може вибрати із бокового меню пункт із зверненнями до деканату (рис. 7).

При необхідності отримати довідку, студент може натиснути на необхідну йому та отримати повний опис про довідку, а також шаблон і зразок заповнення бланку-заяви на довідку.

Натиснувши на документ, студенту буде запропоновано завантажити його на свій пристрій та відкрити у зручному PDF переглядачі.

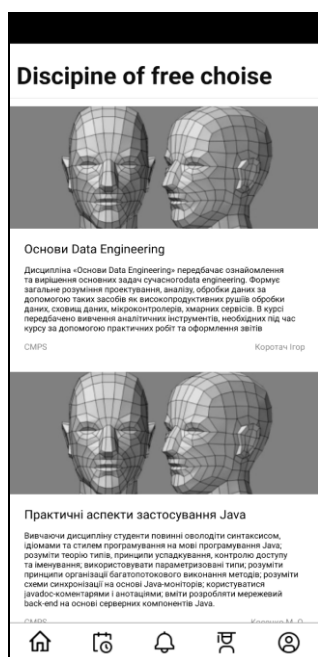


Рис. 6 – Сторінка дисципліни вільного вибору

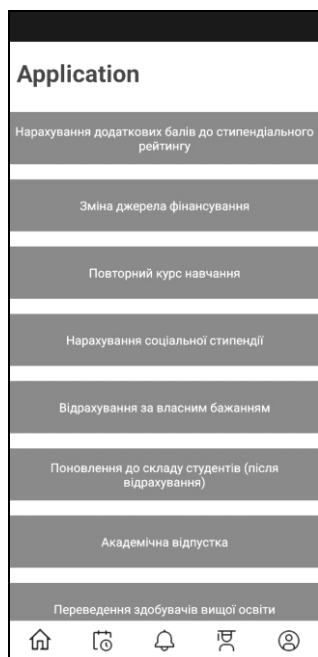


Рис. 7 – Сторінка звернень до деканату

Для повноцінного функціонування застосунку необхідно розмістити на виділеному сервері API-частину застосунку та розповсюдити застосунок через магазини застосунків (PlayStore, AppStore).

Обговорення результатів

Для перевірки роботи кросплатформеного застосунку були проведені тести продуктивності та стрес тести.

При виконанні тесту на продуктивність у мобільному застосунку була використана вбудована можливість моніторингу продуктивності.

Для перевірки роботи застосунку були проведені такі дії:

- Відкрито застосунок із увімкненим монітором продуктивності;

- Швидкий перехід до основних сторінок навігації;

- Перевірка показників моніторингу;

- Перехід на більшу кількість сторінок та виконання мережевих запитів;

- Перевірка показників моніторингу;

- Завершення тесту.

Час проведення тесту склав 2 хвилини.

За даними проведеного тесту (табл. 1) можна визначити, що додаток працює стабільно для користувача та є необхідність оптимізації роботи виконання запитів до серверу.

Таблиця 1 – Результати тесту продуктивності

	Тест 1	Тест 2
UI fps	Max (60), Min (58.5)	Max (60), Min (54.3)
JS fps	Max (59.8), Min (48.9)	Max (59.8), Min (29.0)
Dropped frames	7	12

Для виконання стрес тестування застосунку був використаний список критеріїв, який відображений у розділі тестування.

Для перевірки застосунку на стрес тестування були виконані такі дії:

- Відкриття додатку із чистим менеджером застосунків на пристрої;

- Спроба багаторазового введення пошти та паролю;

- Отримання результатів;

- Швидке перемикання із сторінки на сторінку;

- Отримання результатів;

- Дуже повільне перемикання із сторінки на сторінку;

- Отримання результатів;

- Багаторазове повторне відкриття сторінки з оцінками;

- Отримання результатів;

- Перемикання між декількома застосунками;

- Отримання результатів;

- Завершення проведення стрес тестування.

Таблиця 2 – Результати стрес тесту

	Тест пройдено
Введення даних	Так
Швидке перемикання	Так
Повільне перемикання	Так
Багаторазове відкриття	Так
Перемикання між застосунками	Так

За даними проведеного тесту видно що застосунок витримує усі навантаження та не закривається аварійно або зависає.

У роботі реалізовано кросплатформений застосунок для користування кабінетом студента, використовуючи смартфон із будь-якою операційною системою (Android, iOS). Завдяки застосунку кожен студент без винятку зможе користуватись зручним застосунком та швидко отримувати новини та всю інформацію про свої освітні результати.

Висновки

У роботі було створено кросплатформений застосунок для користування студентами та викладачами, який забезпечує безпеку даних всіх користувачів. Розроблений застосунок дозволяє студенту переглядати поточний розклад, перевірити персональні дані, обрати необхідну йому дисципліну навчання або побачити поточні оцінки за семестр.

Усе це робить застосунок необхідним для щоденного користування, аби завжди бути у потоці подій, що відбуваються у навчальному закладі. Швидкість відображення інформації при використанні мобільного застосунку значно вища, аніж перегляд цих даних через веб браузер.

Кросплатформений застосунок електронного кабінету студента вирішує проблему швидкого та організованого доступу до поточних даних студентського життя.

Список літератури

1. Blanco J. Z., Lucrédio D. A Holistic Approach for Cross-Platform Software Development. *Journal of Systems and Software*. 2021. Vol. 179. P. 110985. doi: 10.1016/j.jss.2021.110985.
2. Dorfer Th., Demetz L., Huber S. Impact of Mobile Cross-Platform Development on CPU, Memory and Battery of Mobile Devices When Using Common Mobile App Features. *Procedia Computer Science*. 2020. Vol. 175. P. 189–196. doi: 10.1016/j.procs.2020.07.029.
3. El-Kassas W. S., Abdullah B. A., Yousef A. H., Wahba A. M. Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal*. 2017. Vol. 8. no. 2. P. 163–190. doi: 10.1016/j.asej.2015.08.004.
4. Rieger Ch., and Majchrzak T. A. Towards the Definitive Evaluation Framework for Cross-Platform App Development Approaches. *Journal of Systems and Software*. 2019. Vol. 153. P. 175–199. doi: 10.1016/j.jss.2019.04.001.
5. Miller Ch. Cross-platform Localization for Native Mobile Apps with Xamarin. 1st ed. 2017. Apress: Imprint: Apress. 2017. 113 p. doi: 10.1007/978-1-4842-2466-3.

6. Wang Zh. A Web-Component-Based Cross-Platform Mobile Application Development Environment for Ordinary Users. *Service-Oriented Computing – IC3SOC 2019 Workshops*, edited by Sami Yangui et al. Springer International Publishing. 2020. P. 266–272. doi: 10.1007/978-3-030-45989-5_22.
7. Latif M., Lakhrissi Y., Nfaoui El H., Es-Sbai N. Review of Mobile Cross Platform and Research Orientations. *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*. 2017. P. 1–4. doi: 10.1109/WITS.2017.7934674.
8. Yukhymenko-Lescroart M. A., Donnelly-Hermosillo D. F., Cowan C. C., Berrett B. D. A Latent Profile Analysis of University Faculty Subtypes for Mobile Technology Integration. *Computers and Education Open*. 2021. Vol. 2. 100052 p. doi: 10.1016/j.caeo.2021.100052.
9. Kwasitsu L., Chiu A. M. Mobile Information Behavior of Warner Pacific University Students. *Library & Information Science Research*. 2019. Vol. 41, no. 2. P. 139–150. doi: 10.1016/j.lisr.2019.04.002.
10. Shada G. S., Ayu M. A. Designing Android User Interface for University Mobile Library. *2018 International Conference on Computing, Engineering, and Design (ICCED)*. 2018. P. 224–229. doi: 10.1109/ICCED.2018.00051.
11. Rataj M., Wojcik J. Are We Ready to Use Mobile Devices in Higher Education? A Case Study from a Central-Eastern European University. *2020 IEEE Frontiers in Education Conference (FIE)*. 2020. P. 1–7. doi: 10.1109/FIE44824.2020.9274102.
12. Kadrija Sh., Memeti A., Luma-Osmani Sh. Development of Mobile App through React Native Hybrid Framework. *2022 11th Mediterranean Conference on Embedded Computing (MECO)*. 2022. P. 1–6. doi: 10.1109/MECO55406.2022.9797173.
13. Azizah A. H., Faidah S. Z., Ulum M. B., Handayani P. Exploration of React Native Framework in Designing a Rule-Based Application for Healthy Lifestyle Education. *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*. 2021. Vol. 1. P. 391–394. doi: 10.1109/ICCSAI53272.2021.9609763.
14. Brito H., Gomes A., Santos A., Bernardino J. JavaScript in Mobile Applications: React Native vs Ionic vs NativeScript vs Native Development. *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*. 2018. P. 1–6. doi: 10.23919/CISTI.2018.8399283.
15. Manduchi G., Luchetta A., Moro G., Garola A. R. Web-Based Streamed Waveform Display Using MDSplus Events and Node.js. *Fusion Engineering and Design*. 2020. Vol. 157. 111625 p. doi: 10.1016/j.fusengdes.2020.111625.

References (transliterated)

1. Blanco J. Z., Lucrédio D. A Holistic Approach for Cross-Platform Software Development. *Journal of Systems and Software*, 2021, vol. 179, p. 110985, doi: 10.1016/j.jss.2021.110985.
2. Dorfer Th., Demetz L., Huber S. Impact of Mobile Cross-Platform Development on CPU, Memory and Battery of Mobile Devices When Using Common Mobile App Features. *Procedia Computer Science*, 2020, vol. 175, pp. 189–196, doi: 10.1016/j.procs.2020.07.029.
3. El-Kassas W. S., Abdullah B. A., Yousef A. H., Wahba A. M. Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal*,

- 2017, vol. 8, no. 2, pp. 163–190, doi: 10.1016/j.asej.2015.08.004.
4. Rieger Ch., and Majchrzak T. A. Towards the Definitive Evaluation Framework for Cross-Platform App Development Approaches. *Journal of Systems and Software*, 2019, vol. 153, pp. 175–199, doi: 10.1016/j.jss.2019.04.001.
 5. Miller Ch. Cross-platform Localization for Native Mobile Apps with Xamarin. 1st ed. 2017, Apress. Imprint: Apress, 2017. 113 p. doi: 10.1007/978-1-4842-2466-3.
 6. Wang Zh. A Web-Component-Based Cross-Platform Mobile Application Development Environment for Ordinary Users. *Service-Oriented Computing – ICSSOC 2019 Workshops*, edited by Sami Yangui et al., Springer International Publishing, 2020, pp. 266–272, doi: 10.1007/978-3-030-45989-5_22.
 7. Latif M., Lakhri Y., Nfaoui El H., Es-Sbai N. Review of Mobile Cross Platform and Research Orientations. *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2017, pp. 1–4, doi: 10.1109/WITS.2017.7934674.
 8. Yukhymenko-Lescroart M. A., Donnelly-Hermosillo D. F., Cowan C. C., Berrett B. D. A Latent Profile Analysis of University Faculty Subtypes for Mobile Technology Integration. *Computers and Education Open*, 2021, vol. 2, 100052 p, doi: 10.1016/j.caeo.2021.100052.
 9. Kwasitsu L., Chiu A. M. Mobile Information Behavior of Warner Pacific University Students. *Library & Information Science Research*, 2019, vol. 41, no. 2, pp. 139–150, doi: 10.1016/j.lisr.2019.04.002.
 10. Shada G. S., Ayu M. A. Designing Android User Interface for University Mobile Library. *2018 International Conference on Computing, Engineering, and Design (ICCED)*, 2018, pp. 224–29, doi: 10.1109/ICCED.2018.00051.
 11. Rataj M., Wojcik J. Are We Ready to Use Mobile Devices in Higher Education? A Case Study from a Central-Eastern European University. *2020 IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–7, doi: 10.1109/FIE44824.2020.9274102.
 12. Kadrija Sh., Memeti A., Luma-Osmeni Sh. Development of Mobile App through React Native Hybrid Framework. *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022, pp. 1–6, doi: 10.1109/MECO55406.2022.9797173.
 13. Azizah A. H., Faidah S. Z., Ulum M. B., Handayani P. Exploration of React Native Framework in Designing a Rule-Based Application for Healthy Lifestyle Education. *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, 2021, vol. 1, pp. 391–394, doi: 10.1109/ICCSAI53272.2021.9609763.
 14. Brito H., Gomes A., Santos A., Bernardino J. JavaScript in Mobile Applications: React Native vs Ionic vs NativeScript vs Native Development. *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, pp. 1–6, doi: 10.23919/CISTI.2018.8399283.
 15. Manduchi G., Luchetta A., Moro G., Garola A. R. Web-Based Streamed Waveform Display Using MDSPplus Events and Node.js. *Fusion Engineering and Design*, 2020, vol. 157, 111625 p, doi: 10.1016/j.fusengdes.2020.111625.

Відомості про авторів (About authors)

Метельов Володимир Олександрович – кандидат технічних наук, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; ORCID: <https://orcid.org/0000-0002-2633-6296>; e-mail: volodymyr.mietielov@khp.edu.ua.

Mietielov Volodymyr – Ph. D., Associate Professor of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; ORCID: <https://orcid.org/0000-0002-2633-6296>; e-mail: volodymyr.mietielov@khp.edu.ua.

Марусенко Олексій Миколайович – Національний технічний університет «Харківський політехнічний інститут», асистент кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; ORCID: <https://orcid.org/0000-0001-6911-2500>; e-mail: Oleksii.Marusenko@khp.edu.ua.

Marusenko Oleksii – Assistant of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; ORCID: <https://orcid.org/0000-0001-6911-2500>; e-mail: Oleksii.Marusenko@khp.edu.ua.

Власов Микита Дмитрович – Національний технічний університет «Харківський політехнічний інститут», студент кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; e-mail: dgygurdenok@gmail.com.

Vlasov Mykyta – Student of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; e-mail: dgygurdenok@gmail.com.

Будь ласка, посилайтесь на цю статтю наступним чином:

Метельов В. О., Марусенко О. М., Власов М. Д. Проектування та розробка електронного кабінету студента для використання на різних операційних системах. *Вісник Національного технічного університету «ХПІ». Серія: Нові рішення в сучасних технологіях.* – Харків: НТУ «ХПІ». 2023. № 1 (15). С. 36-41. doi:10.20998/2413-4295.2023.01.05.

Please cite this article as:

Mietielov V., Marusenko O., Vlasov M. Design and development of the student's electronic cabinet for use on different operating systems. *Bulletin of the National Technical University "KhPI". Series: New solutions in modern technology.* – Kharkiv: NTU "KhPI", 2023, no. 1(15), pp. 36-41, doi:10.20998/2413-4295.2023.01.05.

*Надійшла (received) 09.01.2023
Прийнята (accepted) 08.02.2023*