

УДК 004.422

DOI <https://doi.org/10.32782/IT/2024-3-3>

**Катерина ГОРИШНЯ**

студентка кафедри програмної інженерії, Харківський національний університет радіоелектроніки, пр. Науки, 14, м. Харків, Україна, 61166

ORCID: 0009-0001-9032-4249

**Ірина АФАНАСЬЄВА**

кандидат технічних наук, доцент кафедри програмної інженерії, Харківський національний університет радіоелектроніки, пр. Науки, 14, м. Харків, Україна, 61166

ORCID: 0000-0003-4061-0332

**Костянтин ОНИЩЕНКО**

старший викладач кафедри програмної інженерії, Харківський національний університет радіоелектроніки, пр. Науки, 14, м. Харків, Україна, 61166

ORCID: 0000-0002-7746-4570

**Наталія ГОЛЯН**

кандидат технічних наук, доцент кафедри програмної інженерії, Харківський національний університет радіоелектроніки, пр. Науки, 14, м. Харків, Україна, 61166

ORCID: 0000-0002-1390-3116

**Віра ГОЛЯН**

кандидат технічних наук, доцент кафедри програмної інженерії, Харківський національний університет радіоелектроніки, пр. Науки, 14, м. Харків, Україна, 61166

ORCID: 0000-0002-7196-5286

**Бібліографічний опис статті:** Горішня, К., Афанасьєва, І., Онищенко, К., Голян, Н., Голян, В. (2024). Проектування програмної системи для бронювання квитків. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 23–32, doi: <https://doi.org/10.32782/IT/2024-3-3>

## ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ БРОНЮВАННЯ КВИТКІВ

Дана робота присвячена розробці та впровадженню платформи для бронювання квитків, яка відповідає сучасним вимогам користувачів та організаторів заходів.

**Мета роботи.** Метою даної роботи є проектування та розробка програмної системи для бронювання та покупки квитків на заходи «Ticketer». Основним завданням є створення зручного та ефективного інструменту, який забезпечить користувачам швидкий доступ до інформації про події, а також надасть організаторам заходів сучасні інструменти для управління продажами.

**Методологія.** У процесі проектування платформи було використано поетапний підхід, який включав аналіз ринку, прототипування, визначення бізнес-правил та проектування архітектури системи. Основні елементи інтерфейсу та функціональні можливості були спочатку розроблені у вигляді прототипу за допомогою інструменту Figma, що дозволило оптимізувати користувацький досвід. Крім того, були створені діаграми сценаріїв використання, CRC-карти, об'єктні діаграми та діаграми класів, що допомогли структуровано визначити архітектуру системи та забезпечити узгодженість її компонентів. Особлива увага приділялася забезпеченню надійності та масштабованості платформи, що дозволить її подальший розвиток та інтеграцію з іншими сервісами.

**Наукова новизна.** Наукова новизна проекту полягає в інтеграції сучасних функцій, таких як реальні оновлення доступності квитків, персоналізовані рекомендації та багатоплатформена доступність. Це дозволяє підвищити ефективність роботи платформи та задовольнити змінювані потреби сучасних користувачів. Використання прототипування на ранніх стадіях розробки та поетапного підходу до проектування архітектури забезпечує високий рівень гнучкості та адаптивності системи.

**Висновки.** В результаті дослідження було створено повнофункціональну платформу для бронювання квитків «Ticketer», яка забезпечує зручний, безпечний і персоналізований досвід для користувачів, а також надає організаторам заходів ефективні інструменти для управління продажами. Платформа відповідає сучасним вимогам ринку і має високий потенціал для подальшого розвитку та масштабування.

**Ключові слова:** бронювання квитків, інтерфейс користувача, платформа, архітектура системи, масштабованість.

**Kateryna GORISHNIA**

Student at the Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave., Kharkiv, Ukraine, 61166, [kateryna.horishnia@nure.ua](mailto:kateryna.horishnia@nure.ua)

ORCID: 0009-0001-9032-4249

**Iryna AFANASIEVA**

Ph.D., Associate Professor at the Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave., Kharkiv, Ukraine, 61166, [iryna.afanasieva@nure.ua](mailto:iryna.afanasieva@nure.ua)

ORCID: 0000-0003-4061-0332

**Kostiantyn ONYSHCHENKO**

Senior Lecturer at the Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave., Kharkiv, Ukraine, 61166, [kostiantyn.onyshchenko@nure.ua](mailto:kostiantyn.onyshchenko@nure.ua)

ORCID: 0000-0002-7746-4570

**Natalia GOLIAN**

Ph.D., Associate Professor at the Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave., Kharkiv, Ukraine, 61166, [natalia.golian@nure.ua](mailto:natalia.golian@nure.ua)

ORCID: 0000-0002-1390-3116

**Vira GOLAN**

Ph.D., Associate Professor at the Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave., Kharkiv, Ukraine, 61166, [vira.golan@nure.ua](mailto:vira.golan@nure.ua)

ORCID: 0000-0002-7196-5286

**To cite this article:** Gorishnia, K., Afanasieva, I., Onyshchenko, K., Golian, N., Golan, V. (2024). Proektuvannia prohramnoi systemy dlia broniuвання kvytktiv [Design of a software system for ticket booking]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 23–32, doi: <https://doi.org/10.32782/IT/2024-3-3>

## DESIGN OF A SOFTWARE SYSTEM FOR TICKET BOOKING

*This paper is dedicated to the development and implementation of a ticket booking platform that meets the modern requirements of users and event organizers.*

**Aim of the study.** *The aim of this research is the design and development of a software system for booking and purchasing tickets for events, called «Ticketer». The primary objective is to create a convenient and efficient tool that provides users with quick access to event information while offering event organizers modern tools for sales.*

**Methodology.** *The process of designing the platform followed a phased approach, including market analysis, prototyping, defining business rules, and system architecture design. The main interface elements and functionalities were initially developed as a prototype using the Figma tool, which allowed for the optimization of the user experience. In addition, use case diagrams, CRC cards, object diagrams, and class diagrams were created to structurally define the system architecture and ensure the consistency of its components. Special attention was given to ensuring the platform's reliability and scalability, allowing for further development and integration with other services.*

**Scientific novelty.** *The scientific novelty of the project lies in the integration of modern features such as real-time ticket availability updates, personalized recommendations, and multi-platform accessibility. This enhances the platform's efficiency and meets the changing needs of modern users. The use of prototyping in the early stages of development and a phased approach to architecture design ensure a high level of system flexibility and adaptability.*

**Conclusions.** *As a result of the research, a fully functional ticket booking platform «Ticketer» was created, providing a convenient, secure, and personalized experience for users while offering event organizers effective tools for sales management. The platform meets modern market demands and has high potential for further development and scaling.*

**Key words:** *ticket booking, user interface, platform, system architecture, scalability.*

**Актуальність проблеми.** У сучасному світі проведення різноманітних заходів, таких як концерти, театральні вистави, спортивні події, стає все більш популярним. Однак, процес купівлі квитків на ці заходи часто супроводжується

численними труднощами, такими як довгі черги, обмежена доступність квитків та фрагментована інформація. Це призводить до незадовільного користувацького досвіду та втрати можливостей для організаторів заходів максимізувати

продаж квитків. У зв'язку з цими викликами, нова платформа для продажу квитків на заходи має на меті революціонізувати індустрію квитків, забезпечуючи безперервний та ефективний процес бронювання як для користувачів, так і для організаторів.

#### **Аналіз останніх досліджень і публікацій.**

Існуючі платформи для продажу квитків, такі як Ticketsbox та Kagabas, надають користувачам базові можливості для бронювання та купівлі квитків. Однак, вони часто не задовольняють змінювані потреби користувачів та організаторів. Дослідження показують, що сучасні користувачі потребують реальних оновлень доступності квитків, персоналізованих рекомендацій та мобільної доступності (Smith, Jones, 2020). Крім того, організатори заходів потребують інструментів для аналізу переваг клієнтів, що дозволяє оптимізувати стратегії продажу та маркетингу (Lee, 2019).

**Мета дослідження.** Метою даного дослідження проектування, розробка та впровадження нової платформи для бронювання та купівлі квитків «Ticketer», яка надасть користувачам зручний та швидкий доступ до інформації про заходи, а також забезпечить організаторів заходів ефективними інструментами для управління продажами та маркетингом. Платформа буде оснащена сучасними функціями, такими як реальні оновлення доступності, персоналізовані рекомендації та багатоплатформна доступність, що дозволить підвищити задоволеність користувачів та збільшити продажі квитків.

**Виклад основного матеріалу дослідження.** Проєкт «Ticketer» було розроблено на основі детально визначеного документа «Vision & Scope», який окреслює головну мету та стратегію розвитку платформи (Wieggers, Beatty, 2013, International Organization for Standardization, 2008). Відповідно до документа, основною метою платформи є надання користувачам зручного та сучасного інструменту для бронювання та покупки квитків на різні події, включаючи театральні вистави, кіносеанси та спортивні заходи. Проєкт орієнтований на покращення користувацького досвіду через персоналізовані рекомендації, які надаються на основі вподобань користувача, та забезпечення безпеки даних за допомогою двофакторної аутентифікації.

Наступним етапом було створено прототип системи «Ticketer» у Figma (Vilppu, 2019), який став ключовим етапом у процесі розробки. Цей прототип дозволив візуалізувати та тестувати користувацький інтерфейс на ранніх

стадіях проекту, що забезпечило ефективну оцінку зручності використання та ідентифікацію потенційних проблем до етапу повної реалізації.

Головна сторінка застосунку для бронювання квитків Ticketer містить стратегічно розміщені елементи навігації та переліку доступних подій, що дозволяє користувачеві оперативно знаходити необхідну інформацію. Для підвищення ефективності пошуку подій, на сторінці «Каталог» реалізовано інтегрований фільтр та сортування, що дозволяє користувачам швидко знаходити події відповідно до їхніх уподобань, зокрема за датою, місцем проведення та типом заходу (рис. 1).

На сторінці «Подія» (рис. 2) користувач може отримати доступ до детальної інформації про конкретну подію, включаючи її опис, дату, місце проведення та інші важливі деталі, що забезпечує інформованість і сприяє прийняттю обґрунтованого рішення щодо бронювання квитка. Окрім того, користувач має змогу додати подію до списку бажань, що підвищує персоналізованість та адаптивність сервісу (Kotler, Keller, 2016).

Насупним етапом в проектуванні програмної системи для забезпечення повноцінної взаємодії користувачів із системою «Ticketer», важливе значення має діаграма сценаріїв використання, відома як Smart Use Case Diagram (Max Zosim, 2024). Ця діаграма (рис. 3) ілюструє основні процеси взаємодії користувачів із системою, включаючи такі ключові функції, як реєстрація, пошук подій, бронювання квитків та керування обліковим записом.

Система «Ticketer» побудована на основі кількох класів, які були визначені за допомогою CRC-карт (Class-Responsibility-Collaborator). Ці карти дають можливість чітко визначити відповідальність кожного класу, а також їхню співпрацю з іншими класами (Ambler, 2019). За результатами виконаного аналізу, для забезпечення відповідності системи вимогам користувачів та бізнес-цілям проекту також була побудована діаграма класів (Class Diagram), яка відображає структуру системи на рівні класів та їх взаємозв'язки (Max Zosim, 2024).

На діаграмі класів (рис. 4) можна побачити взаємозв'язки між основними компонентами системи «Ticketer». Зокрема, класи User, Ticket, Payment та інші пов'язані між собою через асоціації, що відображає їх взаємодію в рамках системи. Клас User містить атрибути, такі як id, email, password, що визначають основні характеристики користувача. Клас Ticket включає атрибути id, paymentId, startTime, endTime, що

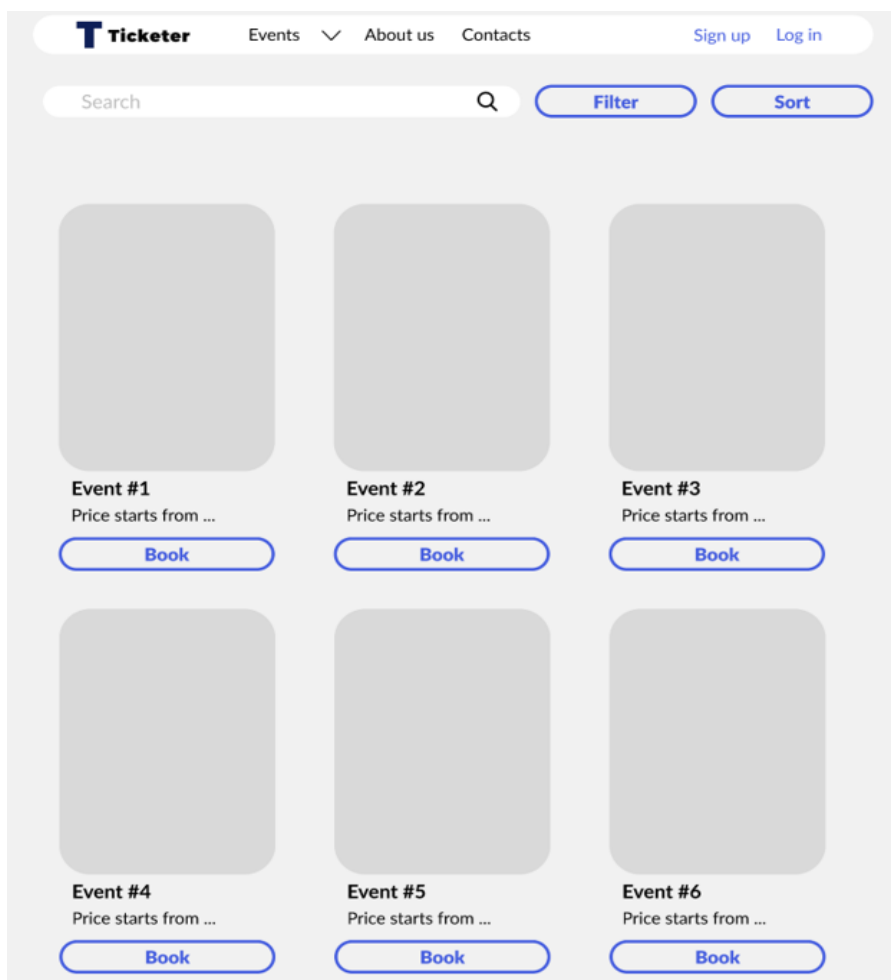


Рис. 1. Фрагменти сторінки «Каталог» з переліком подій

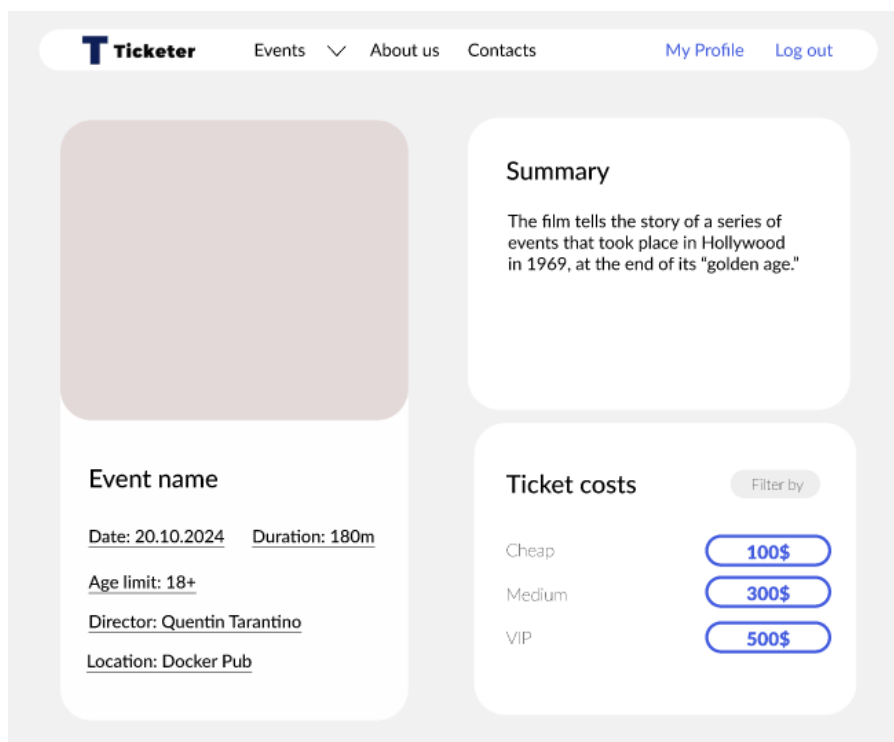


Рис. 2. Фрагменти сторінки «Подія»

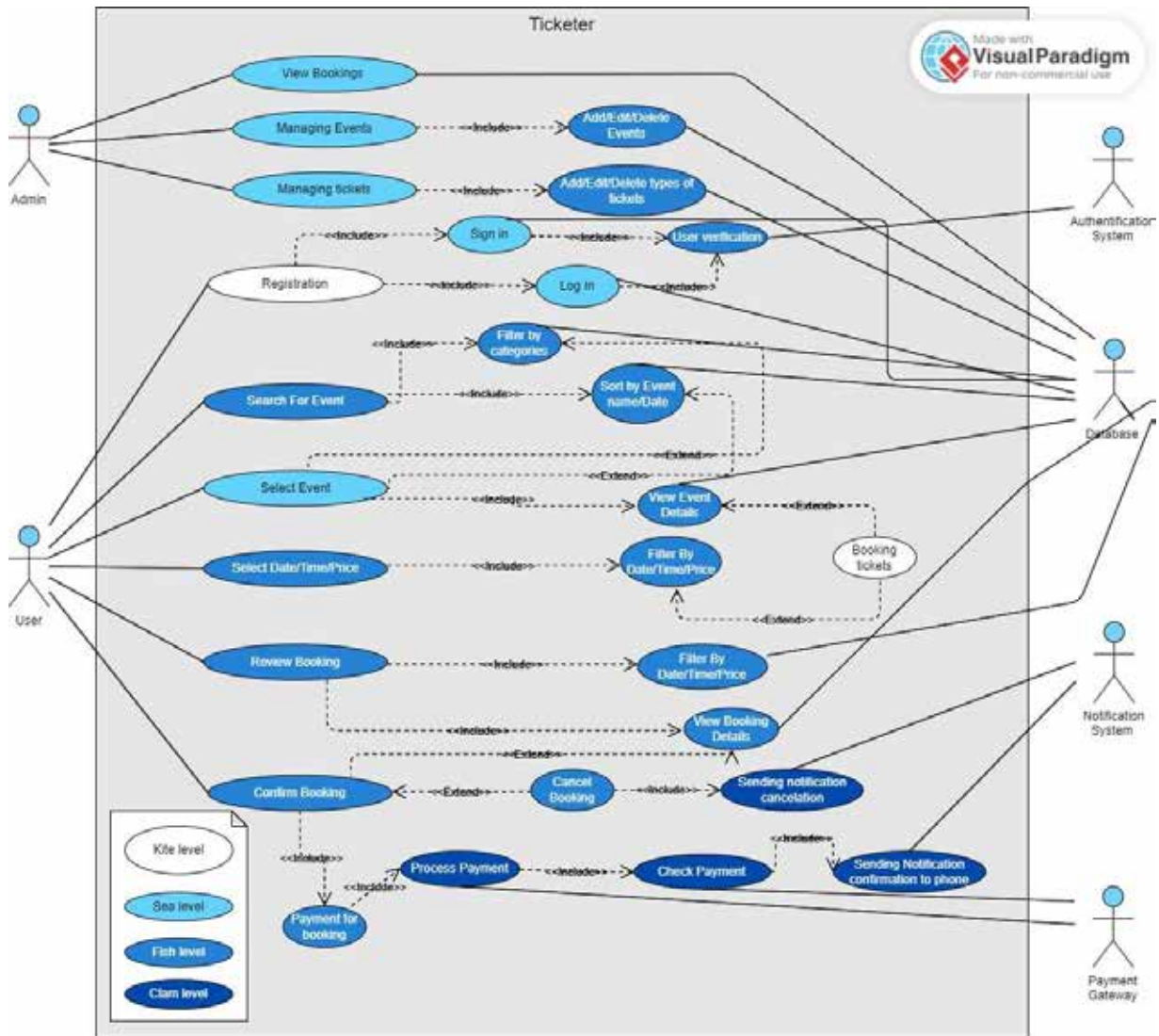


Рис. 3. Smart Use Case diagram проекту «Ticketer»

описують інформацію про квиток, який може бути придбаний користувачем.

Важливим елементом є клас Payment, який містить атрибути moneyPaid, date, userId, eventId, що забезпечують управління процесом оплати квитків. Взаємодія між класами реалізується через методи, такі як buyTicket, refundTicket, createEvent, createPayment, які відповідають за виконання основних функцій системи. Це забезпечує узгоджену роботу різних компонентів системи, що є ключовим для її стабільного функціонування.

Наступним етапом в проектуванні програмної системи було розроблено діаграма об'єктів (Object Diagram), що надає більш детальне уявлення про те, як класи інстанціюються під час виконання програми. Ця діаграма (рис. 5) ілюструє конкретні приклади взаємодії об'єктів класів у реальному часі, що дозволяє краще зрозуміти динаміку роботи системи і виявити

проблеми ще на етапі проектування, що, в свою чергу, допомагає уникнути помилок у процесі розробки (Max Zosim, 2024).

Щоб ефективно відслідковувати стан системи в різних сценаріях, було реалізовано діаграма станів (StateChart). Вона відображає, як система переходить з одного стану в інший залежно від дій користувачів або інших подій (Max Zosim, 2024).

На діаграмі станів (рис. 6) можна побачити послідовність станів, через які проходить система під час виконання різних процесів. Наприклад, процес вибору події починається зі стану «Start», після чого користувач може переглядати каталог подій, вибирати конкретну подію та отримувати детальну інформацію про неї. Далі система переходить до авторизації користувача, де він може увійти до свого облікового запису або зареєструвати новий, якщо ще не має облікового запису.

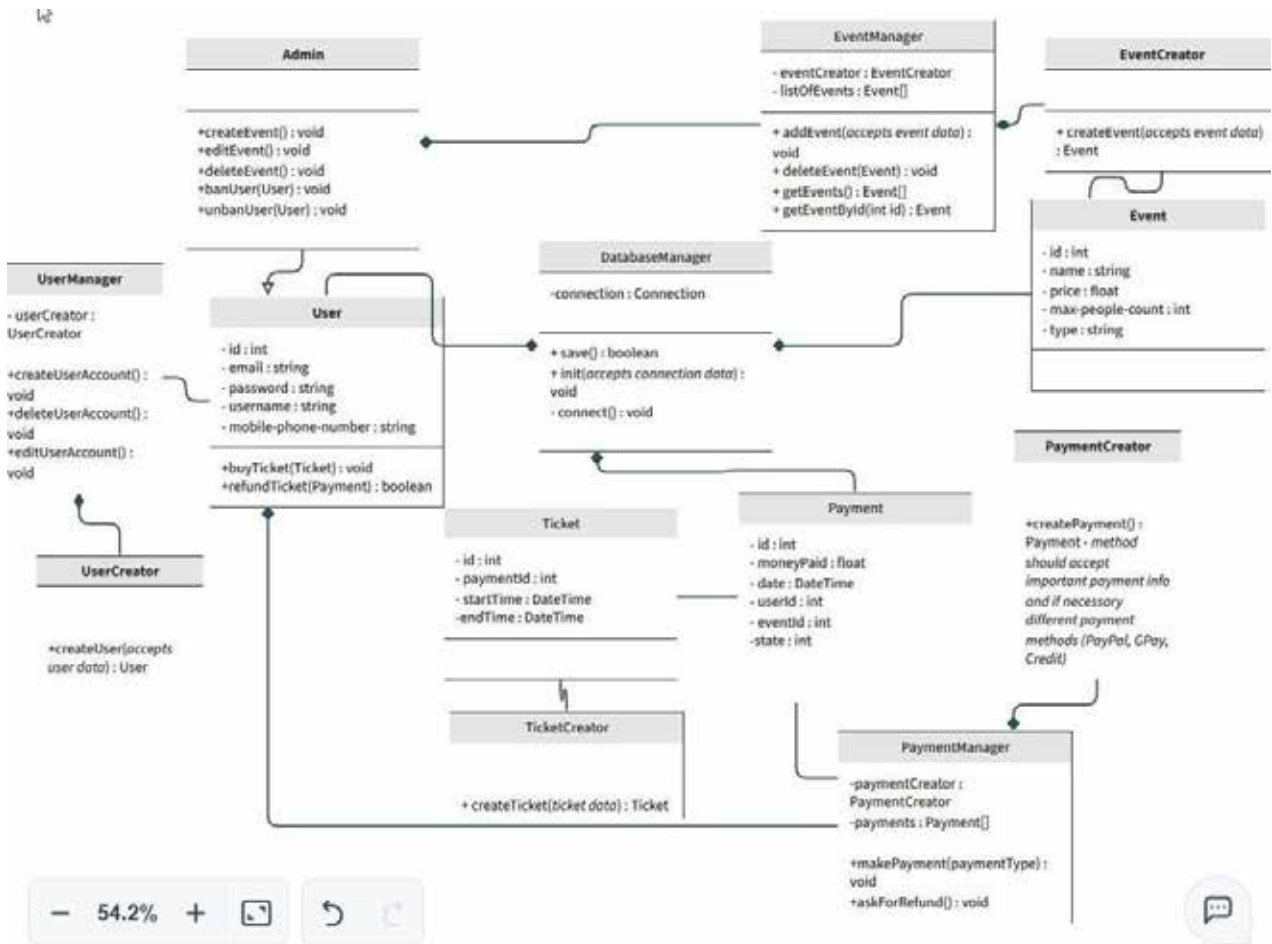


Рис. 4. Class diagram проекту «Ticketer»

Після авторизації користувач може оформити замовлення, яке включає створення замовлення, підтвердження та процес оплати. Система забезпечує перевірку валідності платежу, після чого квиток надсилається користувачу. Ця діаграма дозволяє розробникам краще зрозуміти, як система реагує на дії користувача в різних сценаріях, що допомагає забезпечити її надійність та стабільність в умовах реального використання.

Для відображення розгортання системи у виробничому середовищі було розроблено діаграму розгортання (Deployment Diagram). Вона демонструє, як різні компоненти системи розподілені на серверах та інших апаратних засобах, що забезпечує їхню доступність і масштабованість (Agile Modeling, 2024).

На цій діаграмі (рис. 7) можна побачити розподіл різних компонентів системи «Ticketer» на декілька серверів, таких як Backend Server, Frontend Server, Database Server, та інші. Компоненти, такі як Backend application, Frontend application, Database, Redis, Nginx, і Storage, розгорнуті на відповідних серверах, що забезпечує оптимальну взаємодію між ними.

Наприклад, Backend Server обробляє бізнес-логіку і взаємодіє з Database Server для зберігання даних, тоді як Frontend Server відповідає за рендеринг інтерфейсу для кінцевих користувачів. Використання серверів, таких як Nginx для управління HTTP-запитами та Redis для кешування даних, допомагає підвищити продуктивність та масштабованість системи (NGINX, 2024).

Остаточним результатом проекту «Ticketer» є повнофункціональний продукт, який включає всі перераховані вище елементи і забезпечує користувачам зручний та безпечний спосіб бронювання квитків на різні події (рис. 8-9). Завдяки ретельному плануванню і тестуванню на всіх етапах розробки, платформа «Ticketer» здатна задовольнити потреби користувачів і відповідає сучасним вимогам ринку, що робить її конкурентоспроможною та ефективною на ринку подібних послуг.

**Висновки.** Підсумовуючи, проект «Ticketer» пройшов через ретельний і поетапний процес проектування, за життєвим циклом розробки програмної системи, який охопив всі ключові аспекти від початкового планування до

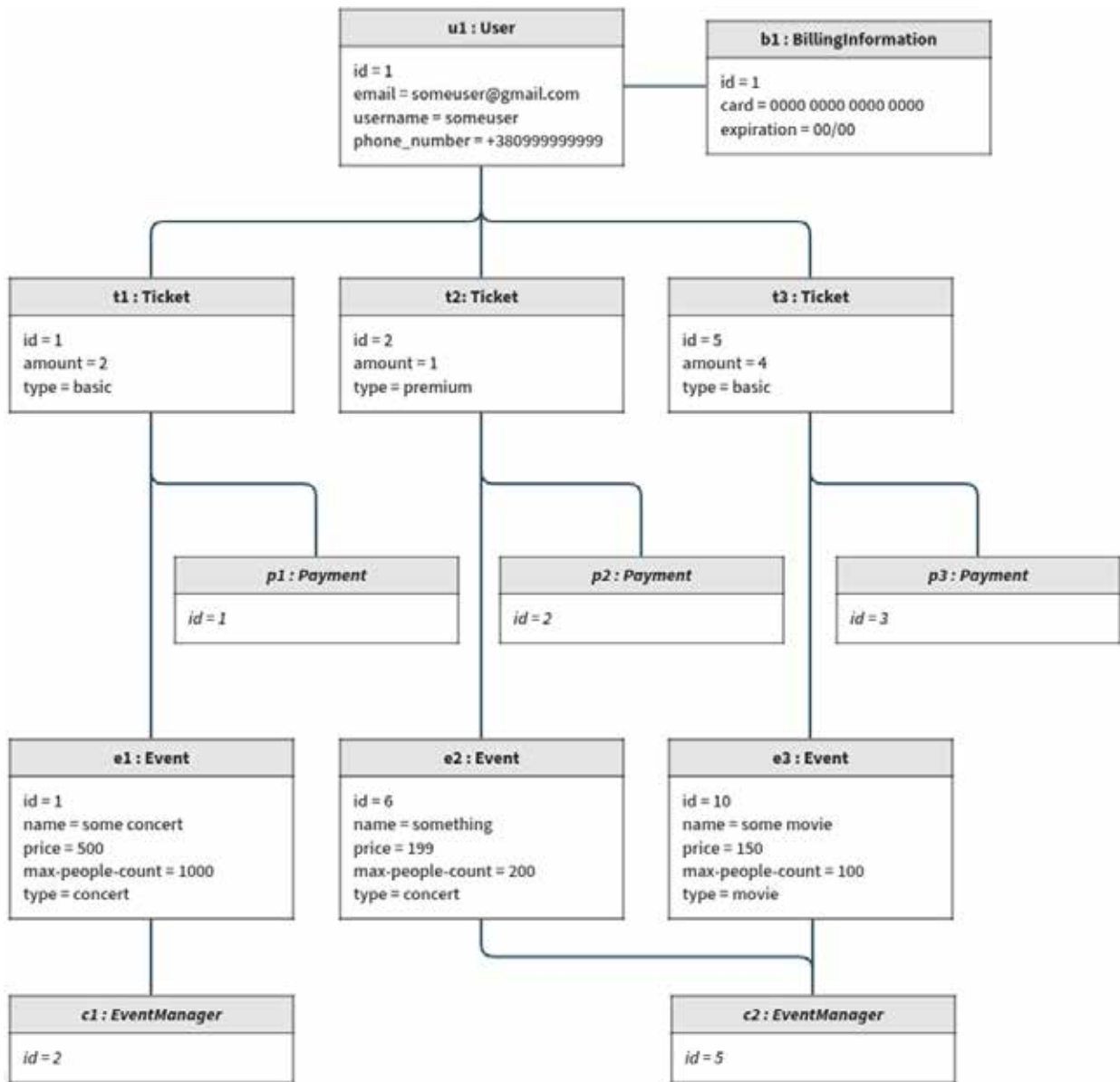


Рис. 5. Object diagram проекту «TICKETER»

остаточного розгортання платформи. Завдяки використанню сучасних інструментів прототипування, чіткому визначенню бізнес-правил і детальному проектуванню архітектури, вдалося створити платформу, яка не тільки задовольняє потреби користувачів, але й відповідає сучасним вимогам ринку.

Платформа «TICKETER» забезпечує зручний і безпечний спосіб бронювання квитків на різні події, що робить її конкурентоспроможною

і здатною зайняти провідну позицію на ринку подібних послуг. Всі виклики, з якими ми стикалися під час розробки, були успішно подолані завдяки комплексному підходу до планування і реалізації проекту. Цей досвід наочно демонструє, що при правильній організації процесу розробки і уважному ставленні до деталей можна створити високоякісний продукт, який буде затребуваним і успішним на ринку.

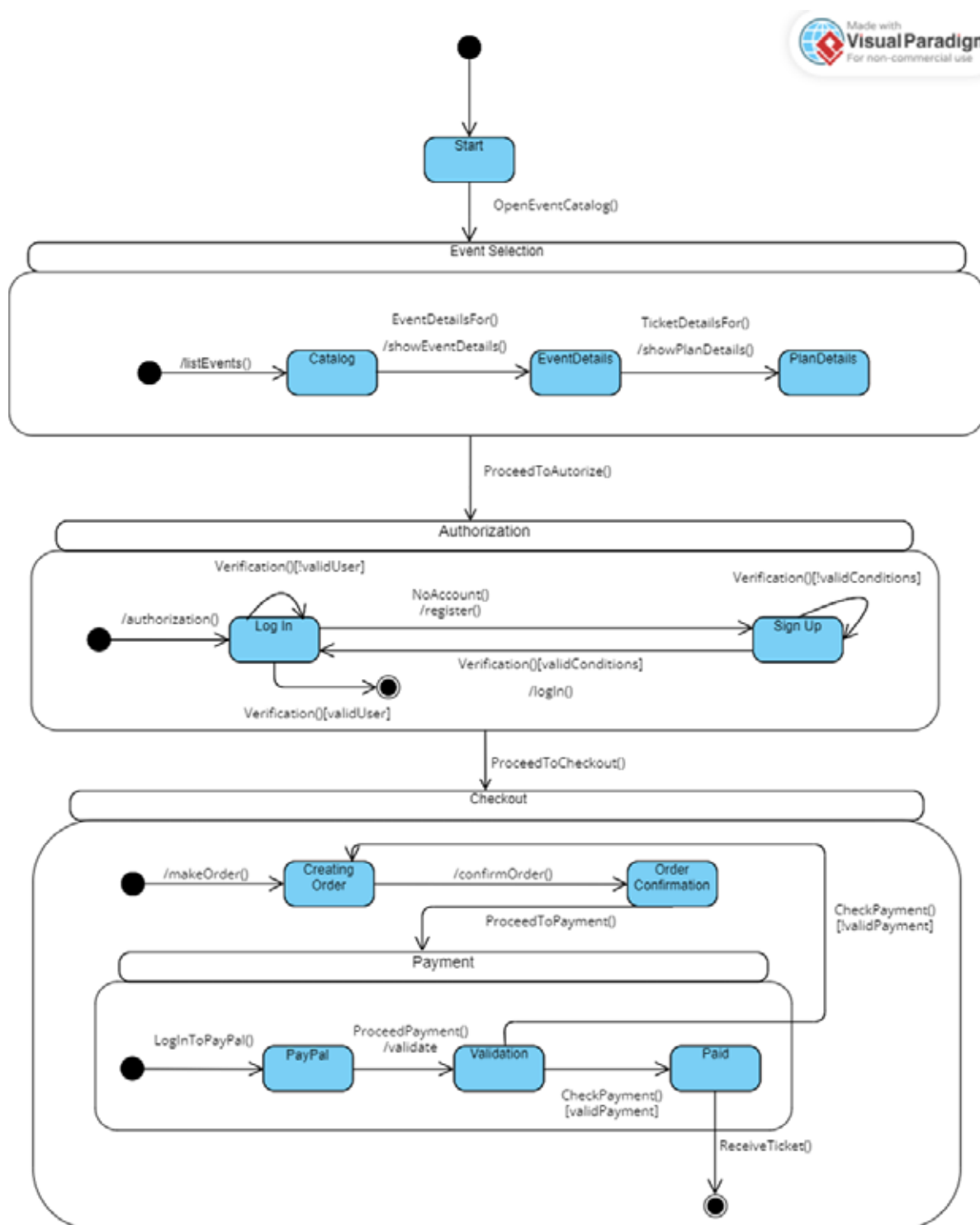


Рис. 6. State Chart проекту «TICKETER»



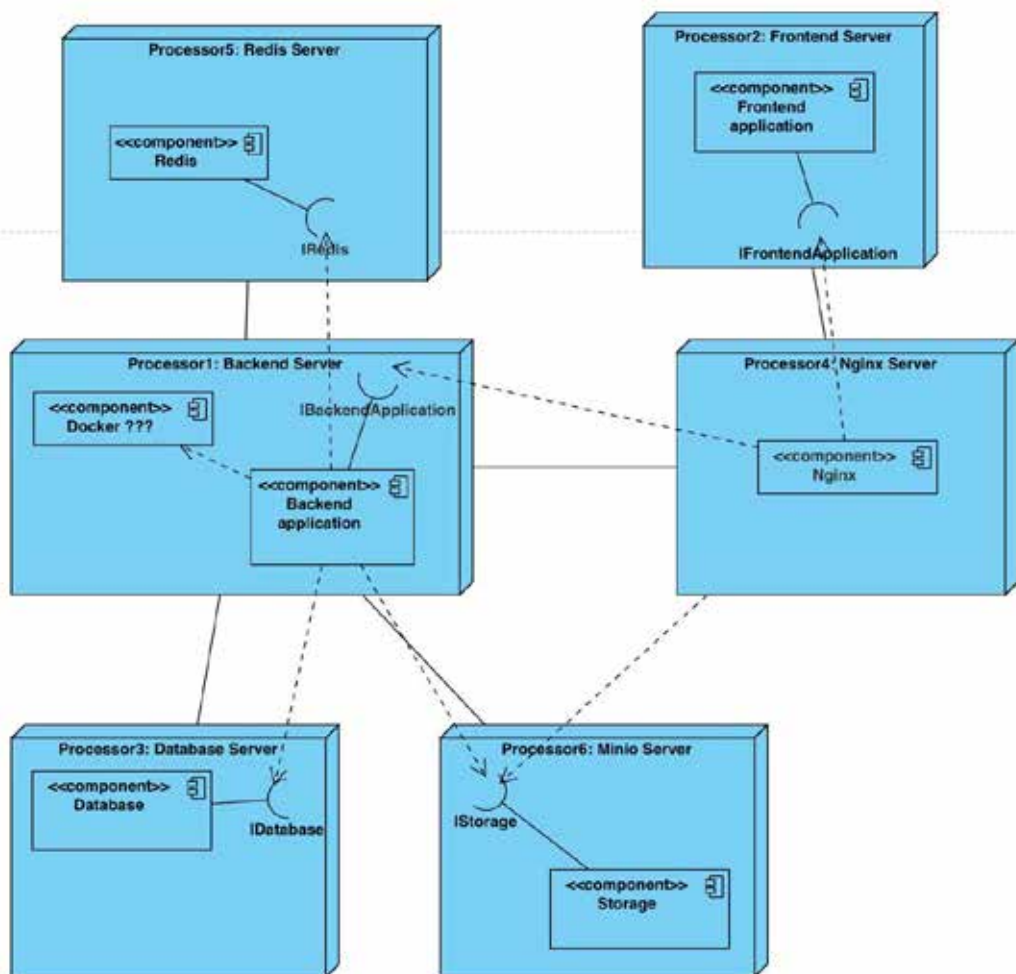


Рис. 7. Deployment Diagram проекту «Ticketer»

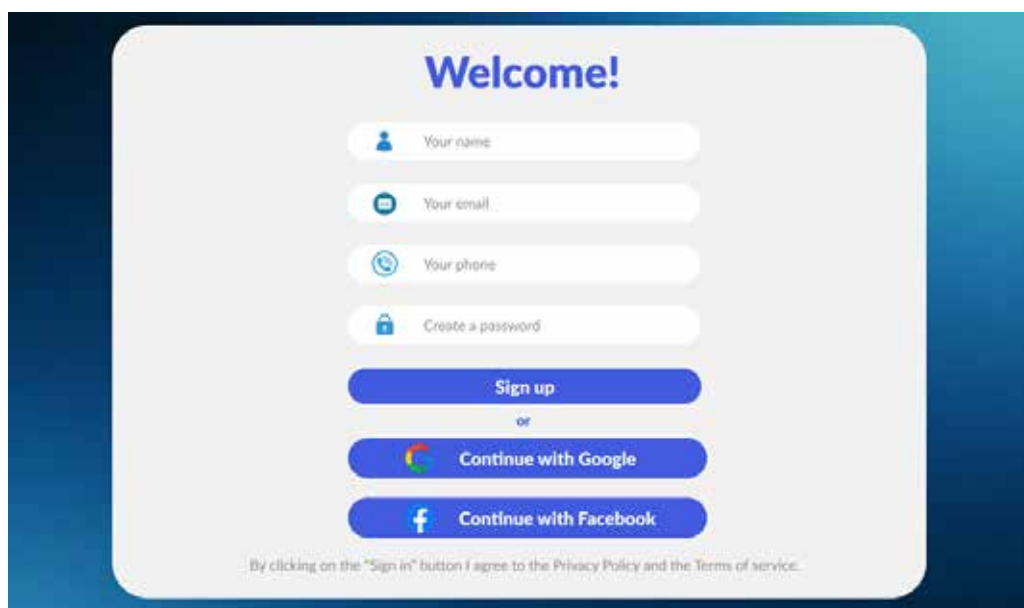


Рис. 8. Сторінка реєстрації (або входу) на платформу «Ticketer»

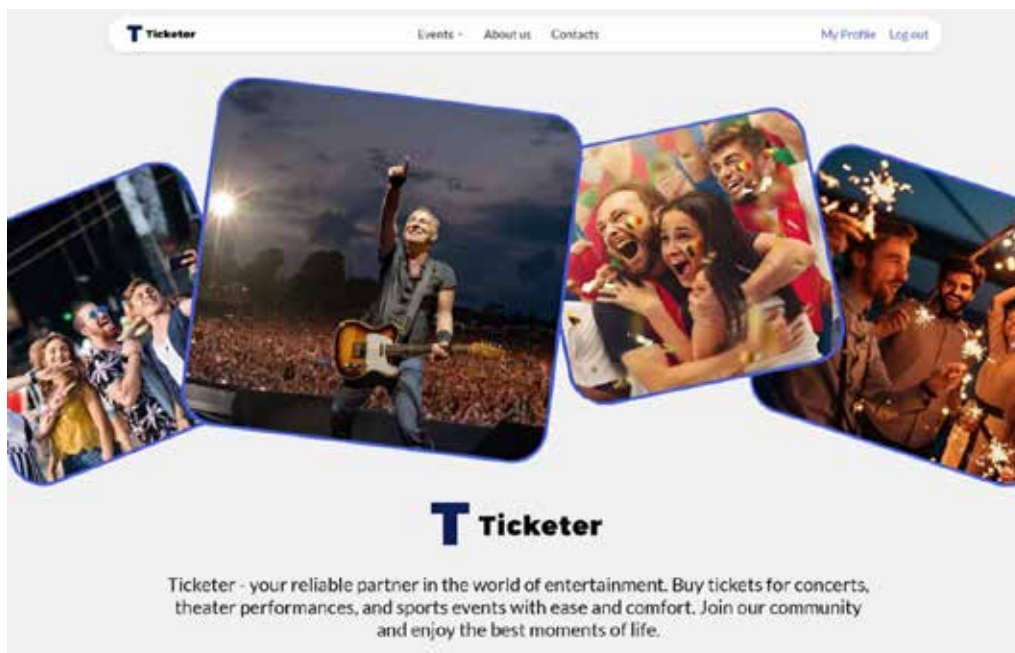


Рис. 9. Головна сторінка платформи «Ticker»

#### ЛІТЕРАТУРА:

1. Smith A., Jones B. Digital Transformation in Business: Strategy and Practice. Routledge, 2020. 320 p.
2. Lee J. Digital Business and E-commerce Management. Pearson, 2019. 450 p.
3. Wieggers K., Beatty J. Software Requirements. 3rd ed. Microsoft Press, 2013. 590 p.
4. Systems and software engineering – Software Life Cycle Processes. ISO 12207:2008. [Чинний від 2008-02-01]. 122 с. (Міжнародний стандарт).
5. Vilppu H. Prototyping for Designers: Developing the Best Digital and Physical Products. O'Reilly Media, 2019. 290 p.
6. Kotler P., Keller K. L. Marketing Management. 15th ed. Pearson, 2016. 812 p.
7. Unified Modeling Language. URL: <https://www.maxzosim.com/unifikovana-mova-modeluvannia/> (дата звернення: 30.08.2024).
8. Ambler S. W. The Object Primer: Agile Model-Driven Development with UML 2.5. Cambridge University Press, 2019. 400 p.
9. Agile Modeling Artifacts. URL: <https://agilemodeling.com/artifacts/crcmodel.htm> (дата звернення: 30.08.2024).
10. NGINX. NGINX Web Server Guide. URL: <https://docs.nginx.com/nginx/admin-guide/web-server/> (дата звернення: 30.08.2024).

#### REFERENCES:

1. Smith, A., & Jones, B. (2020). Digital transformation in business: Strategy and practice. Routledge.
2. Lee, J. (2019). Digital business and e-commerce management. Pearson.
3. Wieggers, K., & Beatty, J. (2013). Software requirements (3rd ed.). Microsoft Press.
4. International Organization for Standardization. (2008). ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes. Retrieved from: <https://www.iso.org/standard/43447.html>
5. Vilppu, H. (2019). Prototyping for designers: Developing the best digital and physical products. O'Reilly Media.
6. Kotler, P., & Keller, K. L. (2016). Marketing management (15th ed.). Pearson.
7. Max Zosim. (2024). Unified modeling language. Retrieved from: <https://www.maxzosim.com/unifikovana-mova-modeluvannia/>
8. Ambler, S. W. (2019). The object primer: Agile model-driven development with UML 2.5. Cambridge University Press.
9. Agile Modeling. (2024). Agile modeling artifacts. Retrieved from: <https://agilemodeling.com/artifacts/crcmodel.htm>
10. NGINX. (2024). NGINX web server guide. Retrieved from: <https://docs.nginx.com/nginx/admin-guide/web-server/>