

DOI: <https://doi.org/10.36910/6775-2524-0560-2026-63-01>

УДК 004.896, 004.91, 004.031.4, 004.622, 004.514

Сомар Артем Ігорович¹, студент магістратури

Ковівчак Ярослав Васильович¹, к.т.н., доцент

<https://orcid.org/0000-0003-3562-4924>

Дубук Василь Іванович², к.т.н., доцент

<https://orcid.org/0000-0002-6339-1032>

¹Національний університет «Львівська політехніка», м. Львів, Україна

²Львівський районний територіальний центр комплектування та соціальної підтримки, м. Львів, Україна

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ПОБУДОВИ ДІАГРАМ ВАРІАНТІВ ВИКОРИСТАННЯ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ

Сомар А.І., Ковівчак Я.В., Дубук В.І. Розробка автоматизованої системи побудови діаграм варіантів використання з елементами штучного інтелекту. Стаття присвячена розробці автоматизованої системи для побудови діаграм варіантів використання на основі технічного опису функціонування системи з використанням великих мовних моделей. Розглянуто існуючі способи, підходи та засоби, які уможливають побудову діаграм варіантів використання під час проектування систем. Обґрунтовано актуальність розробки автоматизованої системи побудови діаграм варіантів використання. Розроблено концептуальну модель системи на основі використання великих мовних моделей. Приведено структурну схему та алгоритм роботи системи. Розроблено діаграму варіантів використання та функціональну модель системи з різними рівнями її деталізації. Побудовано діаграму потоків даних. Приведено діаграму основних компонентів системи. Розроблено діаграму класів та інтерфейс системи. Здійснено реалізацію всіх компонентів системи. Проведено тестування, як основних компонентів, так і системи загалом. Запропонована автоматизована система для побудови діаграм варіантів використання на основі технічного опису функціонування додатку може знайти своє застосування на етапі проектування систем різного призначення, оскільки дає змогу оперативно генерувати UML діаграму прецедентів, зменшуючи трудомісткість ручного аналізу вимог і підвищуючи ефективність проектування інформаційних систем.

Ключові слова: автоматизована система, діаграми варіантів використання, побудова діаграм, нейронні мережі, штучний інтелект, автоматизація процесів.

Somar A., Kovivchak Ya., Dubuk V. Development of an automated system for building use case diagrams with elements of artificial intelligence. The article is devoted to the development of an automated system for building use case diagrams based on a technical description of the application's operation using Large Language Models. Existing methods, approaches and tools that allow to build use case diagrams during systems design are considered. The relevance of developing of an automated system for building use case diagrams is substantiated. A conceptual model of the system is developed based on the use of large language models. A structural diagram and algorithm of the system operation are presented. A use case diagram and a functional model of the system with various levels of detail have been developed. A data flow diagram is constructed. A diagram of the main system components is provided. A class diagram and system interface are developed. All system components are implemented. Testing of both the main components and the system as a whole was carried out. The proposed automated system for building a use case diagram based on a technical description of the application's operation can find its application at the design stage of systems for various purposes, as it allows for the prompt generation of a UML use case diagram, reducing the laboriousness of manual requirements analysis and increasing of effectiveness of information systems design.

Keywords: automated system, use case diagrams, diagram construction, neural networks, artificial intelligence, process automation.

Постановка задачі. Для створення комп'ютерних програмних засобів використовують певні моделі розробки програмного забезпечення. Вибір моделі залежить від складності розробки, функціоналу додатку, необхідного часу та коштів на розробку, а також ряду інших факторів. Кожна із існуючих моделей розробки програмного забезпечення передбачає послідовне виконання відповідних етапів розробки. Незалежно від вибраної моделі, під час розробки додатків, необхідно виконувати етап проектування системи. На етапі проектування здійснюють побудову різних моделей. Це дає змогу розробити та оптимізувати структуру системи, функціональні можливості засобу та вибрати оптимальну взаємодію з користувачами. Проектування здійснюється на основі відповідних загальноприйнятих стандартів. Один із стандартів передбачає використання для побудови моделей уніфікованої мови моделювання UML (Unified Modeling Language), універсальної мови графічного опису програмних систем [1]. За допомогою UML можна здійснити побудову різних типів діаграм, які дають змогу розкрити і спроектувати необхідні аспекти роботи системи, а саме її структуру, алгоритми роботи, основні процеси і потоки даних, визначити та узгодити функції системи для кожного типу користувачів [2].

Одним із ключових інструментів UML є діаграми варіантів використання (use case diagrams). Ці діаграми дають змогу змодельовати та відобразити поведінку системи стосовно її користувачів, а

саме: які користувачі та у якій послідовності мають доступ до функцій системи. За допомогою таких діаграм визначають функціональні можливості майбутнього продукту з точки зору користувачів. Вони є важливим інструментом для візуалізації та опису взаємодії між користувачами (акторами) і системою, що уможлиблює на початкових етапах розробки чітко визначити функціональні вимоги до майбутнього програмного продукту. Діаграми варіантів використання також є основою для розробки технічної документації продукту та тестових сценаріїв на наступних етапах [3].

У більшості випадків, діаграми варіантів використання створюють вручну за допомогою CASE-засобів. Це накладає певні обмеження на виконання етапу проектування, а саме – збільшення затрат часу та ресурсів на побудову необхідних діаграм. Автоматизація процесу побудови діаграм варіантів використання допомогла б спростити аналіз функціональних і нефункціональних вимог до системи і пришвидшити їх побудову, а також підвищити ефективність виконання етапу проектування загалом. На сьогодні, уніфікованого загальноприйнятого рішення щодо автоматичної побудови діаграм варіантів використання, поки що, не існує.

Тому, актуальною є розробка автоматизованої системи для побудови діаграм варіантів використання на основі технічного опису функціонування засобу з елементами штучного інтелекту.

Аналіз останніх досліджень і публікацій. Для аналізу вибрано наступні найбільш розповсюджені системи-аналогі: LucidChart – традиційний графічний інструмент з елементами автоматизації [4]; DiagramGPT – спеціалізоване ШІ-рішення для автоматичної генерації діаграм та ChatGPT [5]; PlantUML – передбачає гібридний підхід на основі поєднання мовних моделей і заданого текстового опису діаграм [6]. Один із результатів застосування засобу LucidChart для генерації діаграми варіантів використання приведено на рис. 1.

Рис. 1 – Приклад побудови діаграми прецедентів за допомогою автоматизованих інструментів Нвс/

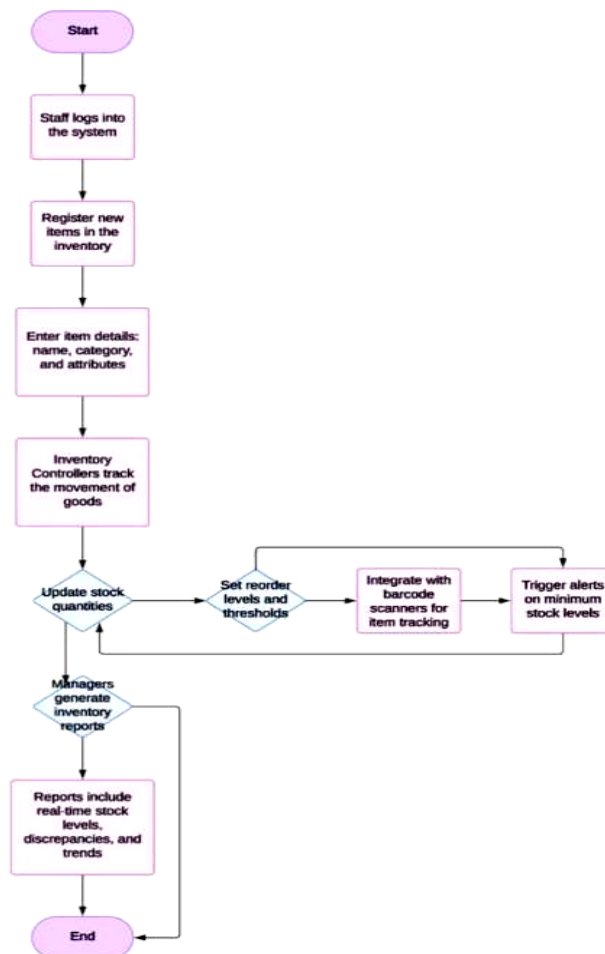


Рис. 1 – LucidChart

LucidChart – програмний засіб, який призначений для візуалізації даних і побудови діаграм. Він широко використовується у бізнес-аналітиці, проектуванні програмного забезпечення та освіті [7]. За допомогою цього засобу можна здійснити побудову блок-схем, ментальних карт, UML-

діаграм, прототипів програмних продуктів. Однак, у даному засобі лише зараз розробники намагаються впроваджувати елементи штучного інтелекту для автоматизації процесів побудови діаграм, і саме тому, діаграма, приведена на рис. 1, не відповідає вимогам стандарту мови UML до діаграм прецедентів.

Засіб LucidChart містить засоби автоматизації побудови діаграм з наступними недоліками:

- засоби штучного інтелекту використовуються лише для побудови простих блок-схем, діаграм класів і діаграм послідовностей;
- обмежені можливості імпорту даних для автоматичного формування діаграм;
- наявні лише елементарне автоматичне вирівнювання і форматування діаграми;
- у випадку побудови більш складних діаграм, таких, як діаграм варіантів використання,

необхідно здійснювати ручне введення всіх елементів.

До переваг даного засобу можна віднести:

- зручний інструменти редагування діаграм;
- наявні інструменти для автоматичного форматування;
- можливість одночасної побудови різних діаграм.

Програмний засіб DiagramGPT є інноваційним інструментом, що використовує великі мовні моделі для автоматичної побудови діаграм на основі їх текстового опису. Він дає змогу користувачам отримати готову діаграму за введеним текстовим описом системи або процесу [8]. Однак, незважаючи на перспективність такої технології, засіб DiagramGPT має суттєві обмеження щодо якості і точності згенерованих діаграм, особливо при спеціалізованому вузькопрофесійному його використанні. Якщо згенерувати діаграму варіантів використання за допомогою DiagramGPT, отримаємо діаграму, яка також не відповідає вимогам UML щодо діаграм прецедентів (див. приклад на рис. 2).

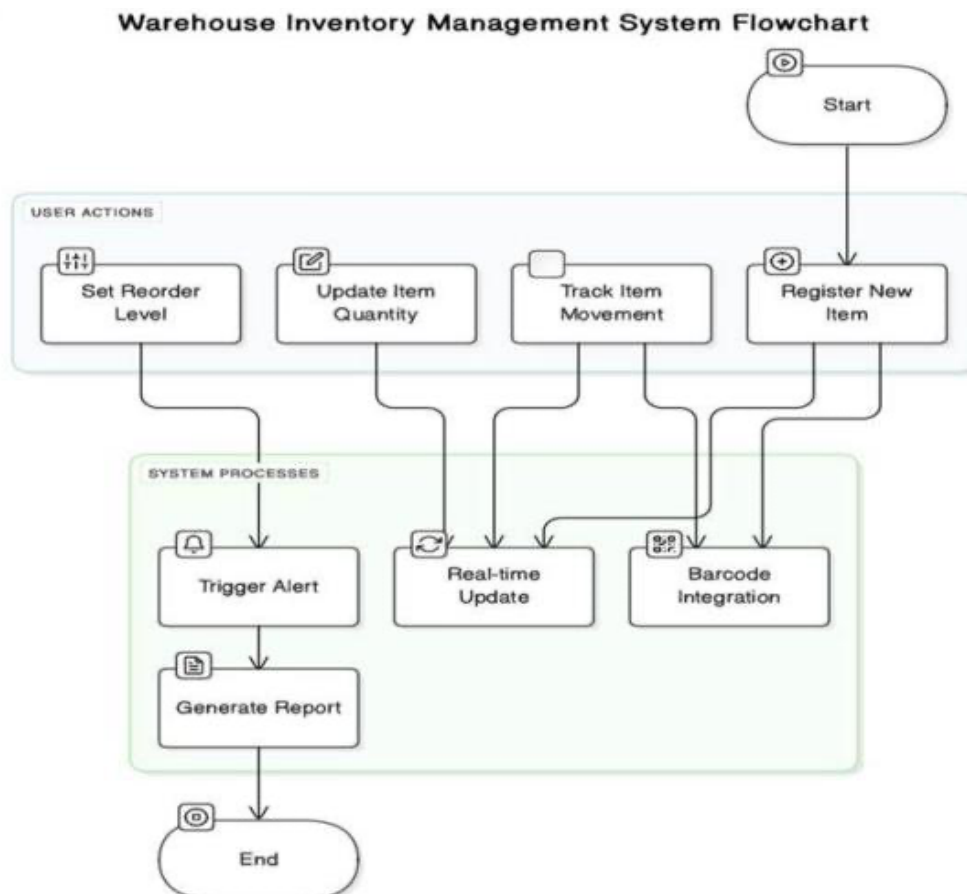


Рис. 2 – Приклад генерації діаграми прецедентів за допомогою засобу DiagramGPT
Обмеження у застосуванні засобу DiagramGPT для побудови діаграм наступні:

- передбачає генерацію лише базових типів діаграм, таких як блок-схеми алгоритмів, діаграми класів, і не призначений для побудови діаграм варіантів використання;
- редагування діаграми можливе лише через текстові уточнення;
- у випадку побудови конструктивно більш складних діаграм, не забезпечує необхідної

точності та коректності отриманого результату;

- не генерує акторів, коректних зв'язків між прецедентами, не розпізнає межі системи.

Переваги засобу DiagramGPT:

- дає змогу здійснити швидку генерацію простих діаграм на основі їх текстового опису;

- містить інтуїтивний інтерфейс;

- наявна можливість інтерактивного уточнення отриманих результатів за допомогою текстового опису.

Найбільш вдалим підходом до побудови діаграм варіантів використання є застосування програмних засобів ChatGPT у поєднанні з PlantUML [9]. Це гібридний підхід до автоматизації створення діаграм, при якому за допомогою великої мовної моделі проводиться аналіз текстового опису, а PlantUML використовується безпосередньо для графічної побудови діаграм.

Діаграму варіантів використання, згенеровану за допомогою ChatGPT та PlantUML, приведено на рис. 3. Дана діаграма відповідає вимогам стандарту UML до побудови діаграм варіантів використання, однак, якість її інформаційного наповнення все ще залишається низькою.

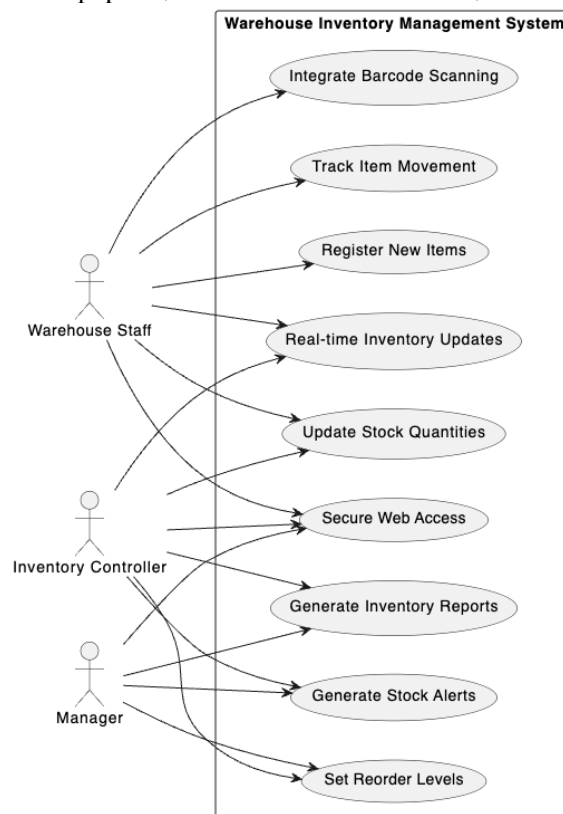


Рис. 3 – Приклад генерації діаграми прецедентів за допомогою ChatGPT та PlantUML

Основні обмеження підходу застосування ChatGPT та PlantUML:

- одночасне використання ChatGPT та PlantUML вимагає відповідних технічних знань і підготовки;

- при внесенні змін необхідно здійснювати написання коду, що вимагає знання синтаксису PlantUML;

- важко здійснювати реалізацію на діаграмах складних взаємозв'язків між прецедентами.

Перевага такого підходу:

- можливість побудови та уточнення діаграм на основі текстового опису.

Метою дослідження є розробка автоматизованої системи побудови діаграми варіантів використання на основі технічного опису системи з використанням елементів штучного інтелекту.

Основна частина. З метою визначення основних зовнішніх сутностей системи і, відповідно, функціональних можливостей засобу, розроблено концептуальну модель системи (рис. 4). До основних зовнішніх сутностей, з якими взаємодіє система, належать: зовнішній сервер з LLM-моделлю; база даних для навчання LLM-моделі та користувач.

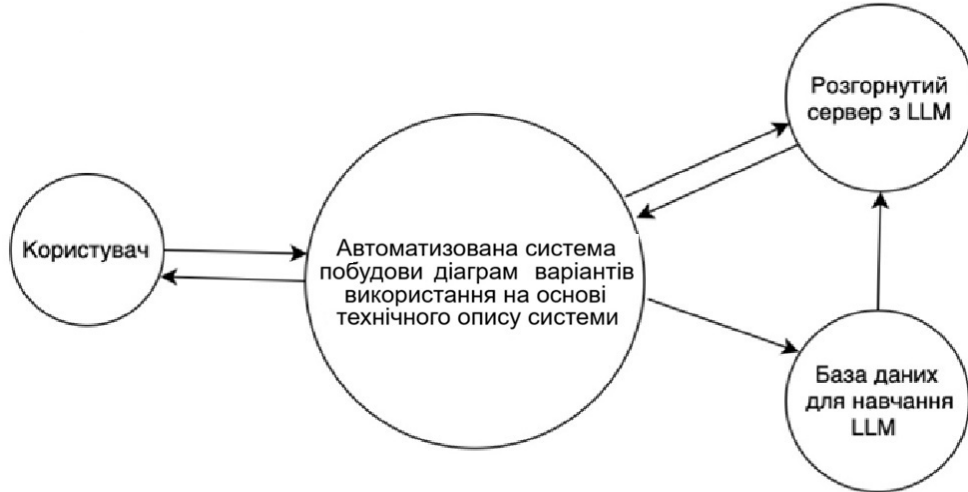


Рис. 4 – Концептуальна модель системи

Сервер з LLM-моделлю здійснює інтелектуальне опрацювання текстових вхідних даних, а саме проводить аналіз отриманого текстового опису, ідентифікує ключові сутності та їх взаємозв'язки. База даних для навчання LLM-моделей є централізованим сховищем даних для навчання великої мовної моделі. Це дає змогу підтримувати цілісність даних протягом всього життєвого циклу запиту та забезпечує можливість відстежувати історію змін.

На рис. 5 приведено архітектуру автоматизованої системи побудови діаграм варіантів використання.



Рис. 5 – Архітектура системи

Вибрана архітектура дає змогу оптимально розподілити обчислювальні ресурси та зменшити навантаження на компоненти клієнтської частини, а це приведе до підвищення продуктивності роботи системи загалом.

Також, було розроблено алгоритм роботи додатку. Блок-схему алгоритму роботи системи побудови діаграм варіантів використання на основі технічного опису приведено на рис. 6. На початку, користувач вводить технічний опис системи. Цей опис є вхідною інформацією, яка передається на серверну частину, де і проводиться її аналіз за допомогою великої мовної моделі. Модель здійснює визначення усіх можливих сутностей і зв'язків, які необхідні для побудови діаграми.

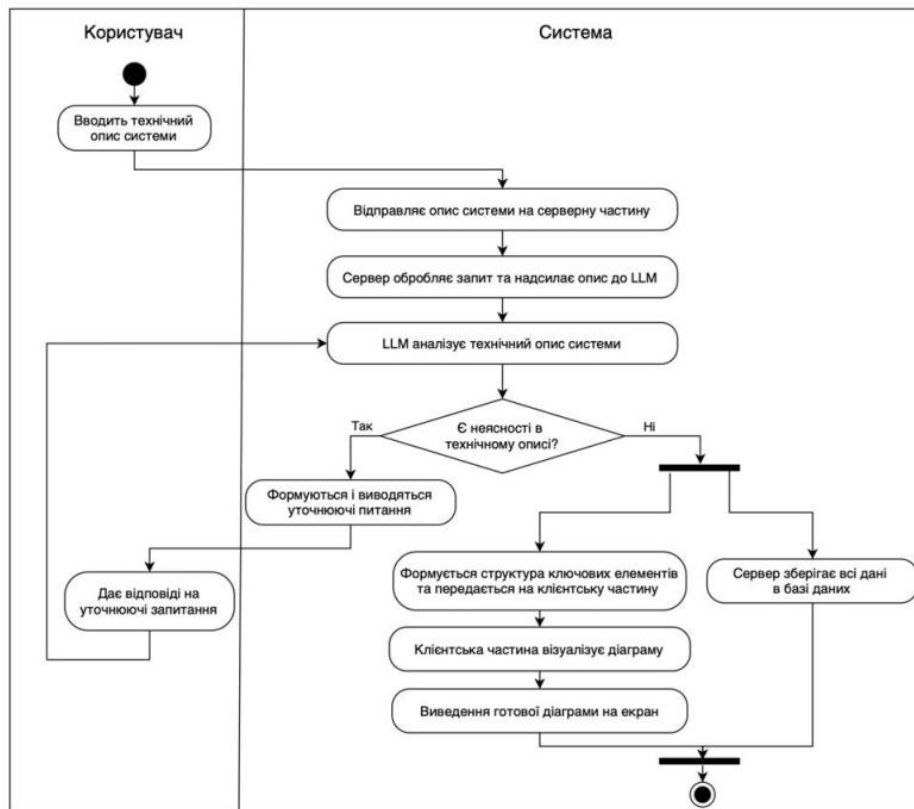


Рис. 6 – Блок-схема алгоритму створення діаграми засобами автоматизованої системи

Під час аналізу вхідної інформації система може виявити певні невизначеності, або недостатність даних у її технічному описі. Тоді формуються уточнюючі питання до користувачів. У такому ітеративному процесі покращується повнота та якість вхідних даних, що приводить до більш точного і адекватного результату роботи моделі. У підсумку, система формує структуру ключових елементів майбутньої діаграми (типи акторів, їх дії і взаємозв'язки). Отримана структура передається на клієнтську частину, де і відбувається її візуалізація.

Розроблено також діаграму варіантів використання для основного користувача системи (рис. 7).

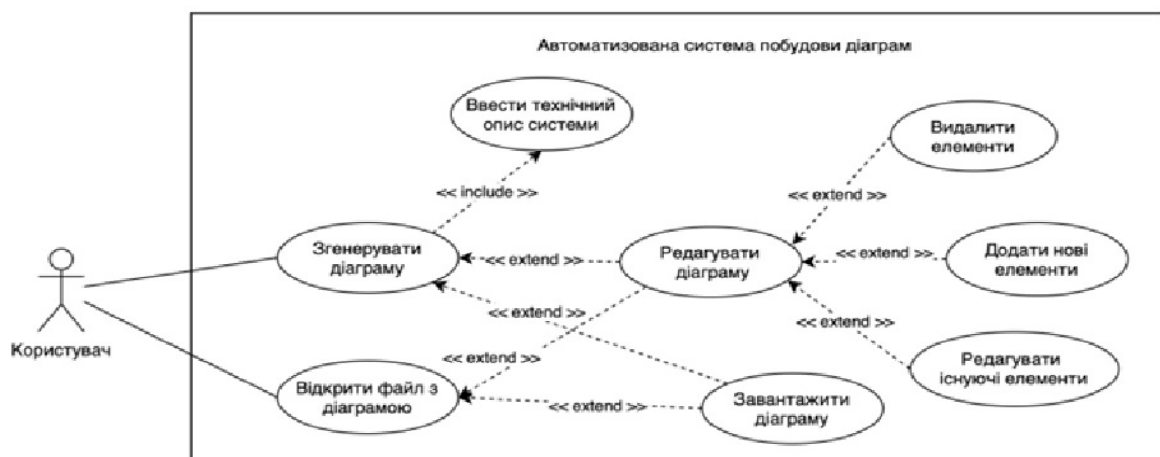


Рис. 7 – Діаграма варіантів використання для користувача системи

Крім можливості генерувати діаграми варіантів використання, користувач може редагувати отриману діаграму, а саме - змінювати графічний вигляд діаграми, додавати нових акторів, редагувати і додавати прецеденти та зв'язки. Також передбачена можливість зберігання отриманих та відредагованих діаграм.

Побудовано функціональну модель розроблюваної системи з необхідними рівнями її деталізації. Контекстну діаграму системи приведено на рис. 8. На цій діаграмі визначено вхідні та вихідні інформаційні та управляючі потоки системи.

Рис. 8 – Контекстна діаграма системи у нотатції IDEF0



З метою оптимального проектування взаємозв'язків між блоками (функціями) системи проведено декомпозицію контекстної діаграми. Діаграма декомпозиції першого рівня контекстної діаграми приведена на рис. 9.

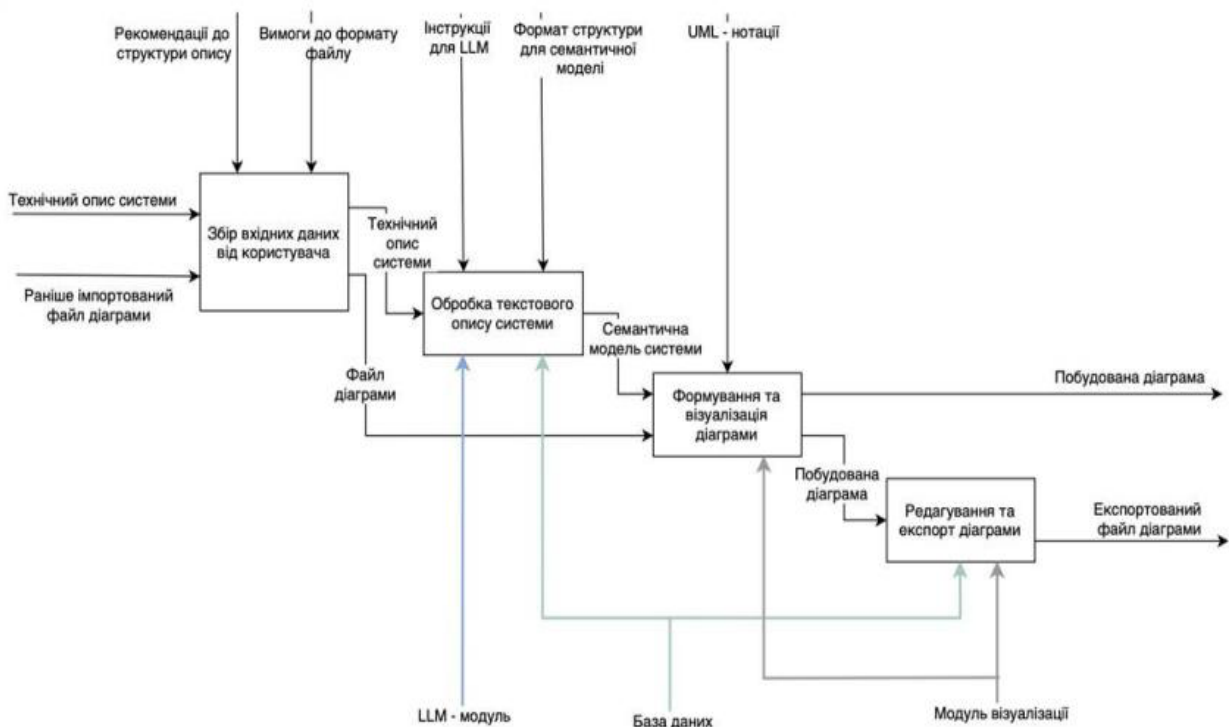


Рис. 9 – Діаграма декомпозиції системи першого рівня

З метою визначення потоків даних між користувачем, підсистемами і базою даних системи під час реалізації ключових функцій, розроблено діаграму потоків даних (рис. 10).

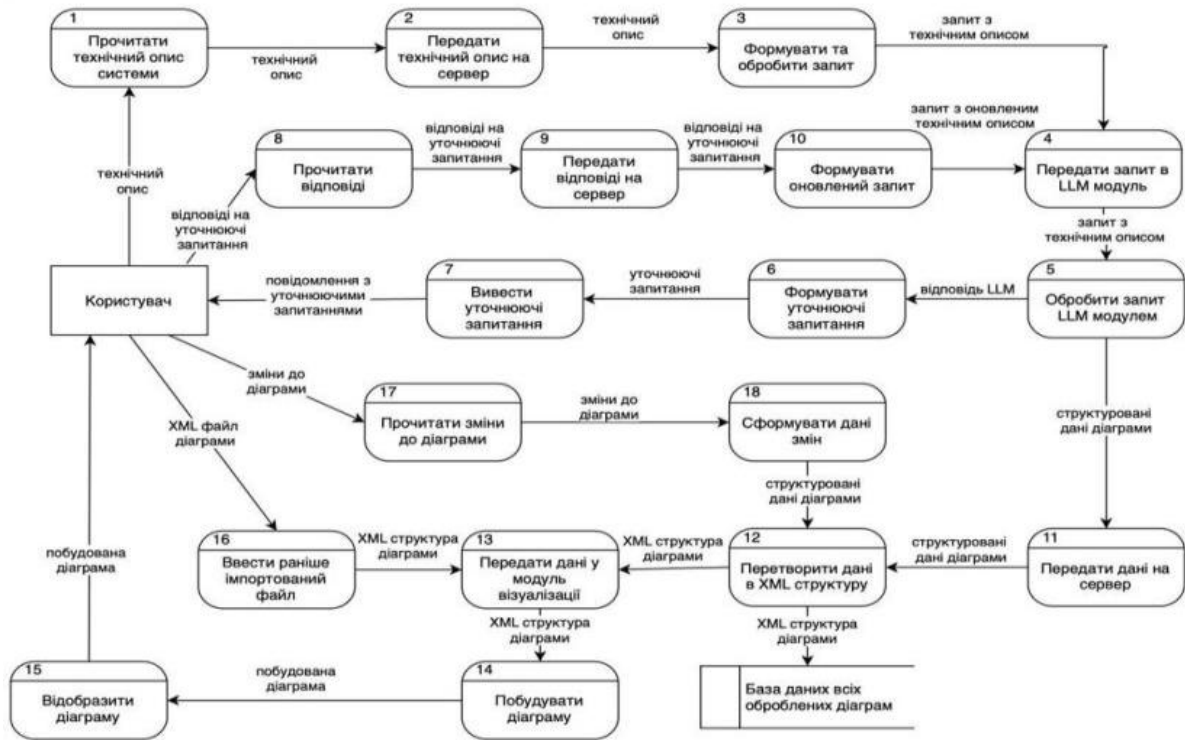


Рис. 10 – Діаграма потоків даних системи

Діаграма компонентів автоматизованої системи побудови діаграм варіантів використання приведена на рис. 11.

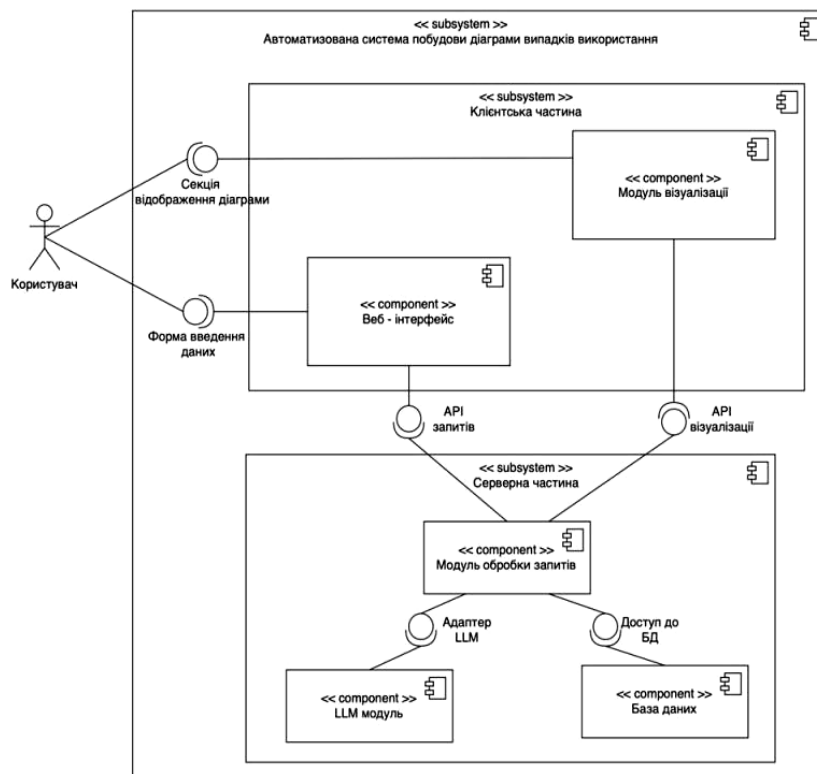


Рис. 11 – Діаграма компонентів системи

Структура розробленого програмного забезпечення автоматизованої системи побудови діаграм варіантів використання відображена на діаграмі класів (рис. 12).

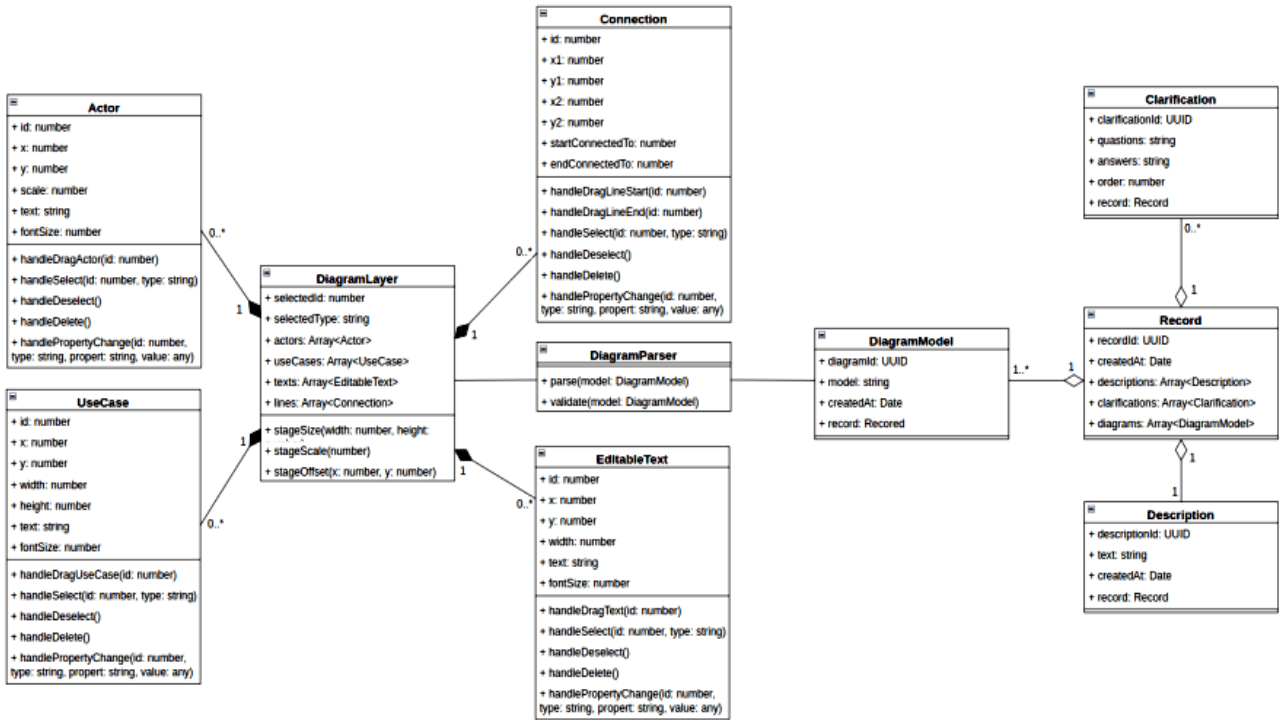


Рис. 12 – Діаграма класів системи

Інтерфейс користувача розроблено за допомогою відкритої JavaScript бібліотеки React [10]. Для роботи з графічними елементами використано бібліотеку React-Konva [11], яка є адаптацією популярної бібліотеки Konva.js [12] для React. Вона дає змогу створювати складні графічні сцени, які містять акторів, прецеденти, текстові елементи, зв'язки, а також підтримує інтерактивні функції (перетягування об'єктів, масштабування сцени, редагування елементів безпосередньо) [13].

Інтерфейс користувача розробленої автоматизованої системи з прикладом побудови діаграми варіантів використання на основі технічного опису, приведено на рис.13.

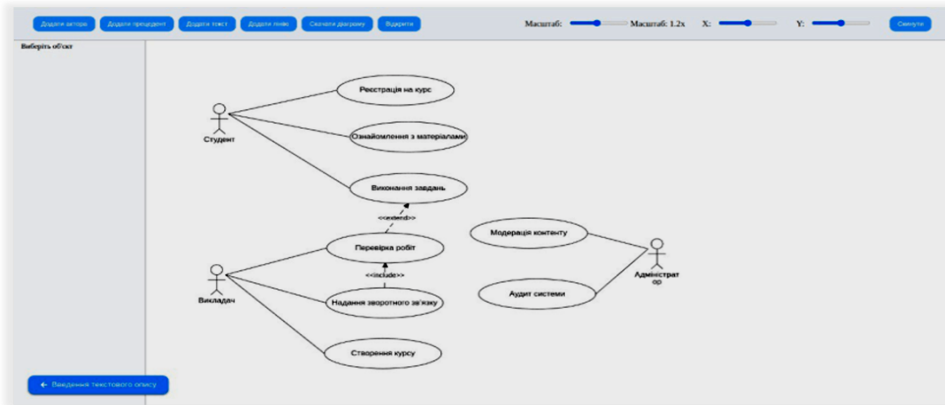


Рис. 13 – Інтерфейс користувача автоматизованої системи побудови діаграм варіантів використання

Також здійснено реалізацію з використанням інструментів мови HTML [14] та JavaScript [15] і тестування усіх компонентів автоматизованої системи побудови діаграм варіантів використання.

Висновки. У результаті виконання роботи розроблено автоматизовану систему для побудови діаграм варіантів використання на основі технічного опису функціонування системи з елементами штучного інтелекту. Розроблена автоматизована система може знайти своє застосування при розв'язанні задач бізнес-аналітики, а також бути використаною розробниками і проектними менеджерами на етапах аналізу вимог і проектування різних комплексних програмних засобів. Вона дає змогу спростити узгодження вимог із замовниками, прискорити створення технічної документації, а також полегшує процеси інформаційної комунікації у команді розробників систем.

Список бібліографічного опису

1. Introduction to Unified Modeling Language (UML). Object Management Group. 2026. URL: <https://www.omg.org/uml/> (date of access: 31.03.2026)
2. By Sruthy. UML Diagram Tutorial: A Complete Guide to UML Diagrams. SoftwareTestingHelp. 2025. URL: <https://www.softwaretestinghelp.com/uml-diagram-tutorial/> (date of access: 31.03.2026).
3. Staff C. How a UML use case diagram can benefit any process. Nulab. 2026. URL: <https://nulab.com/learn/software-development/how-a-uml-use-case-diagram-can-benefit-any-process/> (date of access: 31.03.2026)
4. Lucidchart. Diagramming powered by intelligence. Lucid Software Inc. 2026. URL: <https://www.lucidchart.com/pages> (date of access: 31.03.2026)
5. Free AI Diagram Maker: DiagramGPT Text to Diagram in Seconds. Leiga. 2026. URL: <https://www.leiga.com/free-tools/diagramgpt-free-ai-diagram-generator> (date of access: 31.03.2026)
6. Roques A. PlantUML at a Glance. 2026. URL: <https://plantuml.com/> (date of access: 01.04.2026)
7. Quick Guide: Practice the Basics of Lucidchart. Lucidchart. Lucid Software Inc. 2026. URL: <https://lucid.co/templates/quick-guide-practice-the-basics-of-lucidchart> (date of access: 01.04.2026)
8. Panicker U. DiagramGPT: The AI Diagramming Tool That Will Save You Time. AI Code Journal. Medium. 2026. URL: <https://medium.com/how-we-turn-ideas-into-software-the-creative/diagramgpt-the-ai-diagramming-tool-that-will-save-you-time-3405725eeb8> (date of access: 01.04.2026)
9. De Nicola A., Formica A., Mele I., Missikoff M., Taglino F. A comparative study of LLMs and NLP approaches for supporting business process analysis. Enterprise Information Systems, 2024. 18(10). URL: <https://doi.org/10.1080/17517575.2024.2415578> (date of access: 01.04.2026)
10. React. The library for web and native user interfaces. Meta Open Source. Meta Platforms Inc. 2026. URL: <https://react.dev/> (date of access: 01.04.2026)
11. Getting started with React and Canvas via Konva. Konva. 2026. URL: <https://konvajs.org/docs/react/index.html> (date of access: 01.04.2026)
12. Konva. 2d canvas framework. GitHub Inc. 2026. URL: <https://github.com/konvajs> (date of access: 01.04.2026)
13. Find React Konva Examples and Templates. CodeSandbox.io. CodeSandbox BV. 2026. URL: <https://codesandbox.io/examples/package/react-konva> (date of access: 01.04.2026)
14. HTML: HyperText Markup Language. MDN Web Docs. Mozilla. The Mozilla Foundation. Mozilla Corp. 2025. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (date of access: 01.04.2026).
15. JavaScript. MDN Web Docs. Mozilla. The Mozilla Foundation. Mozilla Corp. 2025. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (date of access: 01.04.2026).

References

1. Object Management Group. (2026). Introduction to Unified Modeling Language (UML). <https://www.omg.org/uml/>
2. By Sruthy. (2025). UML Diagram Tutorial: A Complete Guide to UML Diagrams. SoftwareTestingHelp. <https://www.software-testinghelp.com/uml-diagram-tutorial/>
3. Staff, C. (2026). How a UML use case diagram can benefit any process. Nulab. <https://nulab.com/learn/software-development/how-a-uml-use-case-diagram-can-benefit-any-process/>
4. Lucid Software Inc. (2026). Lucidchart. Diagramming powered by intelligence. <https://www.lucidchart.com/pages>
5. Leiga. (2026). Free AI Diagram Maker: DiagramGPT Text to Diagram in Seconds. <https://www.leiga.com/free-tools/diagramgpt-free-ai-diagram-generator>
6. Roques, A. (2026). PlantUML at a Glance. <https://plantuml.com/>
7. Lucid Software Inc. (2026). Lucidchart. Quick Guide: Practice the Basics of Lucidchart. <https://lucid.co/templates/quick-guide-practice-the-basics-of-lucidchart>
8. Panicker, U. (2026). AI Code Journal. Medium. DiagramGPT: The AI Diagramming Tool That Will Save You Time. <https://medium.com/how-we-turn-ideas-into-software-the-creative/diagramgpt-the-ai-diagramming-tool-that-will-save-you-time-340-5725eeb8>
9. De Nicola, A., Formica, A., Mele, I., Missikoff, M. & Taglino, F. (2024). A comparative study of LLMs and NLP approaches for supporting business process analysis. Enterprise Information Systems. 18(10). <https://doi.org/10.1080/17517575.2024.2415578>
10. Meta Platforms Inc. (2026). Meta Open Source. React. The library for web and native user interfaces. <https://react.dev/>
11. Konva. (2026). Getting started with React and Canvas via Konva. <https://konvajs.org/docs/react/index.html>
12. GitHub Inc. (2026). Konva. 2d canvas framework. <https://github.com/konvajs>
13. CodeSandbox BV. (2026). CodeSandbox.io. Find React Konva Examples and Templates. <https://codesandbox.io/examples/package/react-konva>
14. Mozilla Corp. The Mozilla Foundation. (2025). HTML: HyperText Markup Language. MDN Web Docs. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/HTML>
15. Mozilla Corp. The Mozilla Foundation. (2025). JavaScript. MDN Web Docs. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Історія статті:

Отримано: 03.04.2026 Доопрацьовано: 06.04.2026 Прийнято до друку: 23.05.2026 Опубліковано: 29.05.2026