

# ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ РОБОТОЮ МЕДИЧНОГО ЗАКЛАДУ

Вінницький національний технічний університет

## **Анотація**

*Роботу присвячено створенню інформаційної системи управління роботою медичного закладу з використанням сучасних методологій розробки програмного забезпечення. Проведено аналіз існуючих підходів до автоматизації процесів у медичних закладах, зокрема управління пацієнтами, персоналом та медичною документацією. Наведено схему архітектури MVP системи управління лікарнею. Побудовано UML-діаграми, які забезпечують наочне відображення функціональних процесів, архітектури та взаємодії між користувачами й підсистемами. Розглянуто застосування гнучких методологій розробки програмного забезпечення для підвищення якості, масштабованості та зручності використання інформаційної системи. Отримані результати підтверджують доцільність використання сучасних підходів розробки для створення ефективних медичних інформаційних систем.*

**Ключові слова:** інформаційна система, система управління лікарнею, медичний заклад, автоматизація, Agile.

## **Abstract**

*The paper is devoted to the creation of an information system for managing the work of a medical institution using modern software development methodologies. An analysis of existing approaches to automating processes in medical institutions, in particular, patient, personnel, and medical documentation management, is conducted. A diagram of the MVP architecture of the hospital management system is presented. UML diagrams were constructed to provide a visual representation of functional processes, architecture, and interaction between users and subsystems. The application of Agile software development methodologies to improve the quality, scalability, and usability of the information system was considered. The results obtained confirm the feasibility of using modern development approaches to create effective medical information systems.*

**Keywords:** information system, hospital management system, medical institution, automation, Agile.

## **Вступ**

Сучасні медичні заклади характеризуються значним обсягом інформаційних потоків, пов'язаних з обліком пацієнтів, управлінням персоналом, веденням медичної документації та організацією лікувального процесу. Використання паперових носіїв або застарілих інформаційних рішень ускладнює оперативний доступ до даних, знижує ефективність управління та може призводити до помилок у процесі надання медичних послуг.

Актуальність роботи зумовлена необхідністю цифровізації закладів охорони здоров'я та підвищення ефективності управління медичними процесами, що потребує використання сучасної гнучкої методології розробки Agile.

Мета дослідження полягає у розробці інформаційної системи управління роботою медичного закладу для автоматизації процесів обліку пацієнтів, управління персоналом, медичною документацією та ресурсами закладу охорони здоров'я.

## **Результати дослідження**

Проведений огляд сучасних реалізацій програмних продуктів свідчить про те, що існуючі рішення або є надмірно складними та дорогими, або не забезпечують комплексного управління всіма процесами лікарні. Це обґрунтовує доцільність розробки системи управління лікарнею з оптимальним набором функцій, гнучкою архітектурою та можливістю подальшого масштабування.

Більшість інформаційних систем мають базовий набір функцій, таких як автентифікація користувачів, керування профілями та робота з даними. Для системи управління лікарнею ключовою цінністю є підвищення ефективності медичних процесів, зменшення навантаження на персонал та

забезпечення зручної взаємодії між усіма учасниками процесу лікування. На основі аналізу існуючих реалізацій було сформовано наступні функціональні вимоги до мінімально життєздатного продукту (MVP). Функціональні вимоги до системи управління лікарнею:

- система повинна надавати можливість реєстрації та автентифікації користувачів із розподілом ролей (адміністратор, лікар, медичний персонал, пацієнт);
- користувач повинен мати можливість переглядати та редагувати власний профіль відповідно до наданих прав доступу;
- система повинна забезпечувати реєстрацію та облік пацієнтів із збереженням їх персональних і медичних даних;
- система має надавати лікарям можливість створювати, переглядати та редагувати електронні медичні картки пацієнтів;
- система повинна забезпечувати запис пацієнтів на прийом до лікаря та перегляд розкладу прийомів;
- система має надавати можливість формування та управління розкладом роботи медичного персоналу;
- система повинна забезпечувати збереження та облік результатів лабораторних досліджень і обстежень;
- система має надавати можливість формування медичних призначень та рекомендацій;
- система повинна забезпечувати ведення обліку використання медичних ресурсів і матеріалів;
- система має надавати адміністраторам можливість формування звітів про діяльність лікарні;
- система повинна забезпечувати розмежування доступу до інформації відповідно до ролей користувачів;
- система має підтримувати ведення історії змін медичних даних.

Серед наявних архітектур програмного забезпечення було обрано мікросервісну архітектуру як оптимальну для створення системи управління лікарнею, адже вона забезпечує масштабовану систему управління роботою медичного закладу, яка може бути легко доповнена новим функціоналом та горизонтально розширена при зростанні кількості користувачів (пацієнтів і персоналу). Також мікросервісна архітектура має ряд додаткових переваг:

- запуск мікросервісів відбувається швидше, що підвищує продуктивність розробників і прискорює процес розгортання системи;
- кожен мікросервіс може бути розгорнутий незалежно від інших, що дозволяє вносити зміни в окремий компонент без необхідності зупиняти роботу інших сервісів, які продовжують функціонувати без перебоїв;
- мікросервіси надають можливість окремого масштабування, що дозволяє оптимально використовувати ресурси для кожного сервісу відповідно до потреб;
- збої в одному мікросервісі не призводять до критичних збоїв усієї системи. проблеми в окремому компоненті зазвичай не впливають значною мірою на загальну працездатність, особливо в масштабних додатках;
- використання мікросервісів усуває довгострокові зобов'язання щодо конкретних технологій. кожен новий сервіс може бути реалізований з використанням нового технологічного стека, а будь-який сервіс може бути замінений або переписаний у прийнятні строки й без необхідності змінювати всю систему;
- завдяки чітко визначеним завданням кожного мікросервісу забезпечується краща організація структури додатка, що полегшує розуміння, підтримку і тестування компонентів;
- відокремлені мікросервіси забезпечують гнучкість у переналаштуванні та перекомпонуванні для виконання завдань у різних додатках, таких як обслуговування вебклієнт та публічних API.

Для розробки системи управління лікарнею було визначено нефункціональні вимоги з урахуванням специфіки медичної сфери, де ключовими аспектами є захист персональних даних, безперервність роботи та швидкий доступ до медичної інформації, а саме:

- безпека збереження медичних даних: персональні та медичні дані пацієнтів повинні зберігатися у зашифрованому вигляді з використанням сучасних криптографічних алгоритмів;
- контроль доступу: система повинна забезпечувати багаторівневу систему авторизації та розмежування прав доступу відповідно до ролей користувачів;
- приватність інформації: доступ до медичних даних повинен бути обмежений та фіксуватися у журналі дій користувачів;

- надійність: система повинна забезпечувати безперебійну роботу та автоматичне відновлення після збоїв без втрати даних;
- доступність: система має бути доступною не менше ніж 99,9% часу протягом року;
- продуктивність: середній час відгуку системи на дії користувача не повинен перевищувати 2 секунд;
- зручність користування: інтерфейс системи повинен бути інтуїтивно зрозумілим та адаптованим для різних категорій користувачів;
- сумісність: система повинна коректно працювати у сучасних веб-браузерах та на різних пристроях;
- підтримка доступності: інтерфейс повинен бути адаптований для користувачів з обмеженими можливостями;
- резервне копіювання: система повинна забезпечувати регулярне автоматичне резервне копіювання даних;
- супровід та оновлення: програмний продукт повинен підтримувати можливість оновлення без зупинки роботи системи.

На рисунку 1 наведено схему архітектури MVP системи управління лікарнею, яка демонструє розподіл функцій між сервісами, взаємодію через API Gateway та використання подій для інтеграції між компонентами.

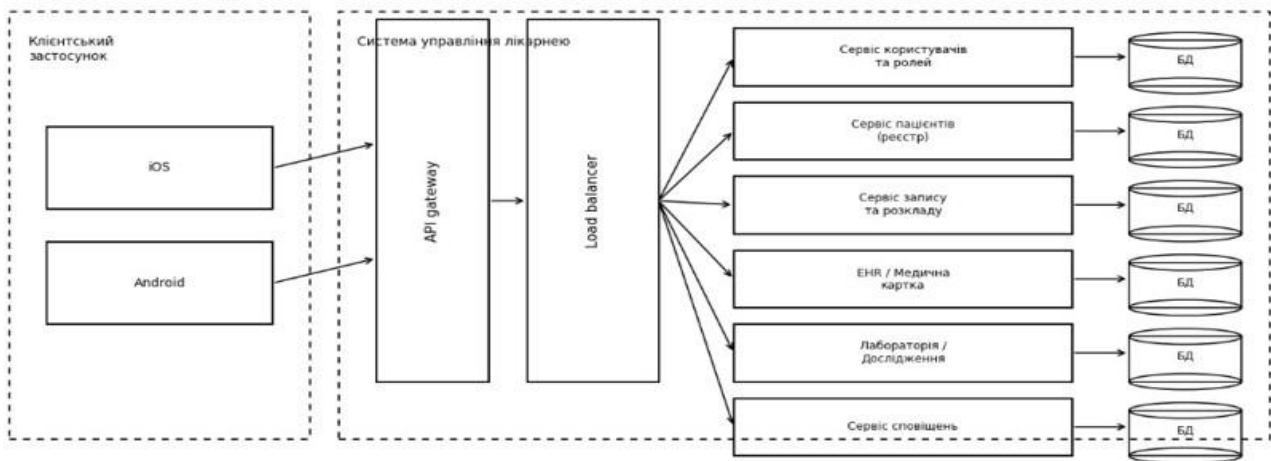


Рис.1. Архітектура (MVP) системи управління лікарнею

Ключові сервіси системи управління лікарнею включають: 1) сервіс користувачів та ролей — забезпечує реєстрацію, вхід, аутентифікацію та авторизацію користувачів, а також розмежування прав доступу (адміністратор, лікар, медсестра, реєстратура, пацієнт); 2) сервіс пацієнтів (реєстр) — відповідає за створення та ведення профілю пацієнта, зберігання персональних даних, пошук і перегляд історії звернень; 3) сервіс запису та розкладу — реалізує запис на прийом, керування чергою/талонами, перегляд та формування розкладу лікарів; 4) ЕНР/медична картка — зберігає медичні записи (візити, діагнози, призначення, процедури), забезпечує доступ лікаря до історії лікування; 5) сервіс лабораторії/досліджень — фіксує направлення на аналізи та обстеження, зберігає результати та надає їх лікарю/пацієнту відповідно до прав доступу; 6) сервіс ресурсів (склад/ліжка) — облік медикаментів/матеріалів, контроль залишків, облік зайнятості ліжок/ліжок (за потреби); 7) сервіс фінансів/білінгу — облік оплат (для платних послуг), рахунки, акти, базова фінансова звітність; 8) сервіс сповіщень — надсилає повідомлення про запис, зміни розкладу, готовність результатів (email/SMS/push).

Для забезпечення обміну інформацією між мікросервісами використано метод SAGA, який полягає у тому, що локальна транзакція оновлює свою базу даних і запускає наступну транзакцію, публікуючи повідомлення або подію. Якщо локальна транзакція не вдається, SAGA виконує серію компенсаційних транзакцій, щоб відкотити зміни, внесені попередніми транзакціями. Це гарантує, що система залишається узгодженою навіть у разі збою транзакцій.

На рисунку 2 представлена UML-діаграма прецедентів системи управління лікарнею.

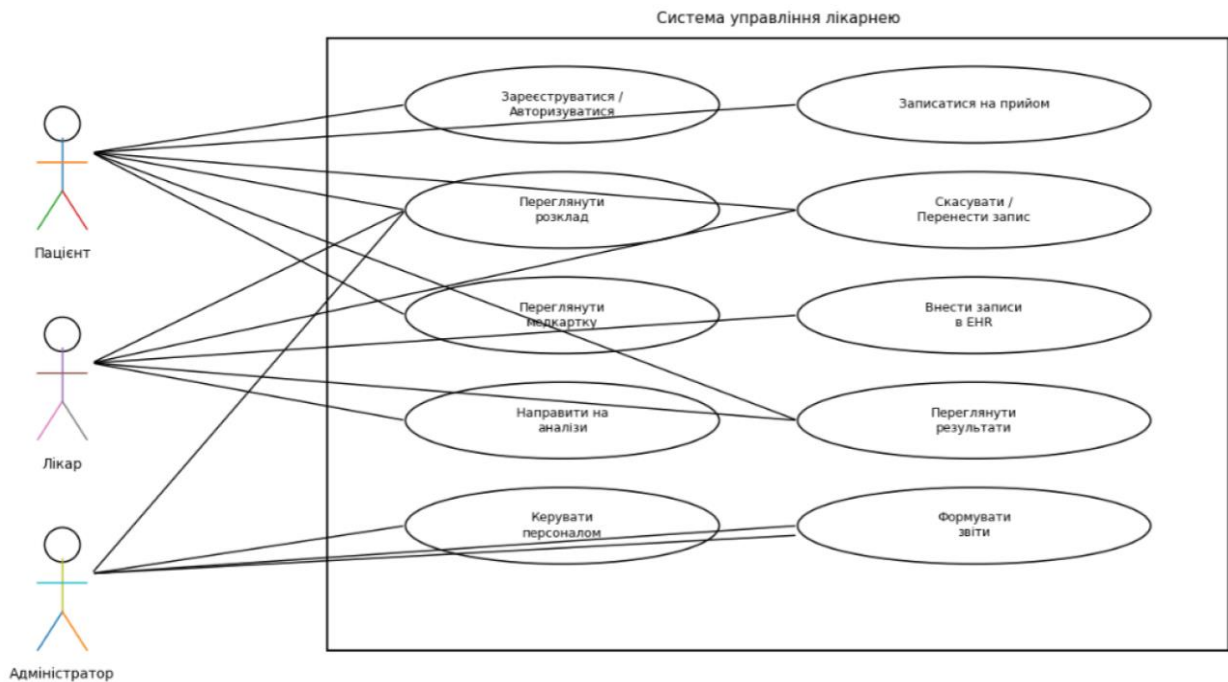


Рис.2. Діаграма прецедентів системи управління лікарнею

Діаграма послідовності (Sequence Diagram) — показує часові особливості передачі та приймання повідомлень об'єктами. Для ключового сценарію «Запис пацієнта на прийом» було визначено взаємодію між: Пацієнт → UI → API Gateway → Сервіс розкладу → БД → Сервіс сповіщень.

Для першої взаємодії було визначено головного актора — користувача, набір сутностей — початковий екран, форма входу, форма реєстрації, сервіс верифікації та база даних. Після встановлення взаємозв'язків за допомогою повідомлень та розробки додаткових сценаріїв взаємодії було отримано діаграму послідовності для сценарію «перша взаємодія користувача», що зображено на рисунку 3.

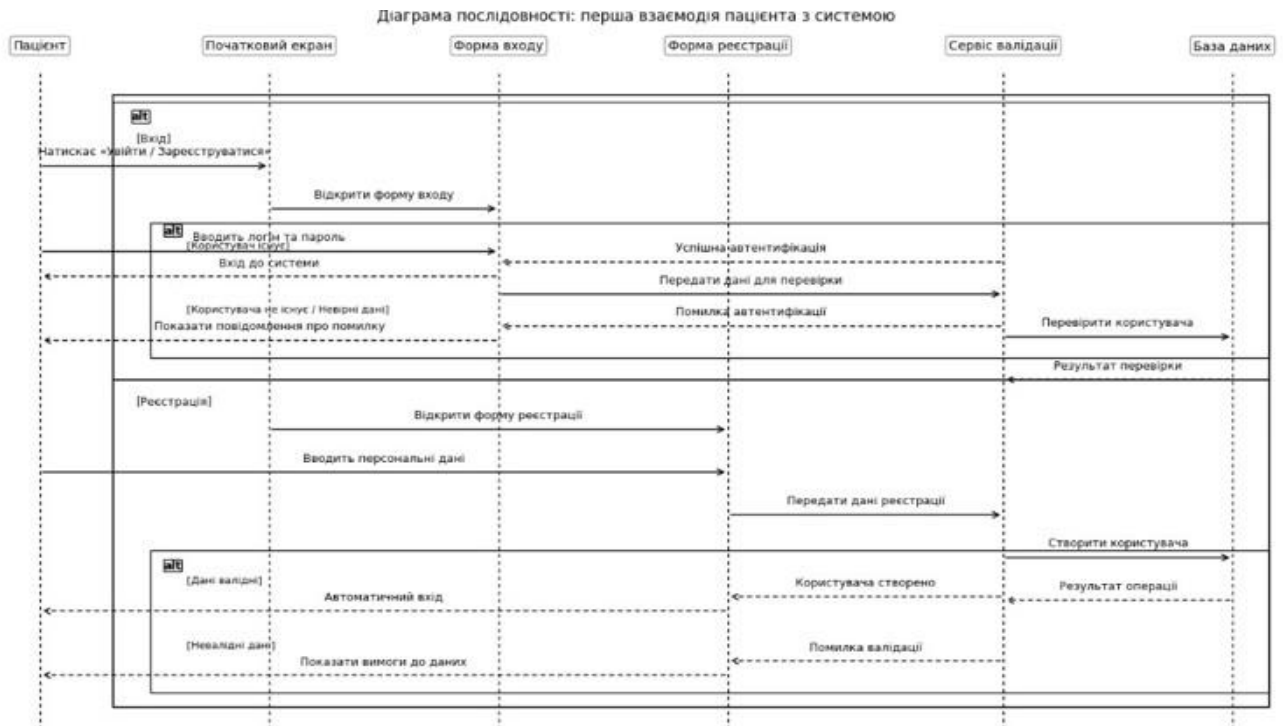


Рис.3. Діаграма послідовності сценарію «Запис пацієнта на прийом»

Діаграма класів — статичне представлення структури моделі в UML. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів.

Для системи управління лікарнею було обрано такі основні класи: особа (абстрактний клас), пацієнт (успадкований від класу особа), персонал (абстрактний клас), лікар (успадкований від класу персонал), медсестра (успадкована від класу персонал), запис на прийом (Appointment), електронна медична картка (MedicalRecord / EHR), призначення (Prescription), направлення на аналізи (LabOrder), результат аналізів (LabResult), рахунок/оплата (Invoice / Payment) та верифікація/сесія (AuthSession / Verification).

Дана структура класів формує основу системи, де абстрактний клас «особа» містить спільні атрибути та базові методи для всіх учасників (пацієнтів і працівників). Абстрактний клас «персонал» забезпечує спільні властивості для медичних працівників, а класи «лікар» та «медсестра» уточнюють рольові характеристики. Класи «запис на прийом», «EHR», «призначення», «направлення на аналізи» та «результат аналізів» реалізують ключові медичні процеси: планування візитів, ведення історії лікування, призначення та контроль обстежень. Клас «рахунок/оплата» відповідає за фінансову частину (за потреби — для платних послуг), а «верифікація/сесія» забезпечує підтвердження доступу та безпеку роботи користувачів. Після встановлення взаємозв'язків між класами було отримано UML діаграму класів, зображену на рисунку 4.

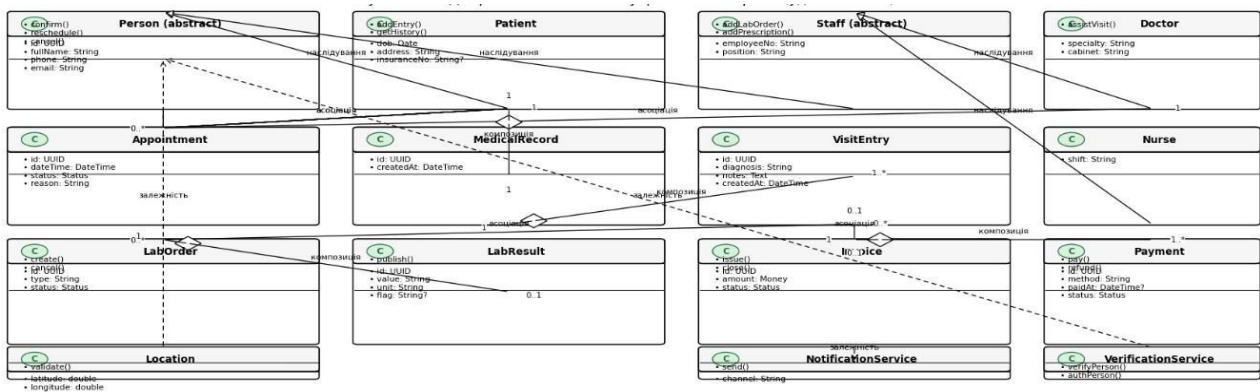


Рис.4. Діаграма класів системи онлайн-замовлень послуг

## Висновки

Запропоновано інформаційну систему управління лікарнею з використанням методології Agile та мікросервісної архітектури, що дозволило чітко структурувати процес розробки та забезпечити узгоджену взаємодію між основними компонентами системи.

Проведений аналіз існуючих програмних рішень засвідчив їх обмежену функціональність або складність впровадження, що підтвердило доцільність створення спеціалізованої системи з гнучкою архітектурою та оптимальним набором функцій.

Побудовано UML-діаграми, які забезпечують наочне відображення функціональних процесів, архітектури та взаємодії між користувачами й підсистемами, що значно спрощує подальшу реалізацію програмного продукту.

Застосування гнучкої методології Agile дозволило адаптувати процес створення системи до можливих змін вимог та поетапно реалізовувати функціональність. Використання мікросервісної архітектури забезпечило масштабованість, надійність і розподіл відповідальності між окремими сервісами, що є особливо важливим для медичних інформаційних систем із високими вимогами до стабільності та безпеки даних.

Таким чином, отримані результати підтверджують, що використання сучасних методів проектування та гнучких методологій розробки програмного забезпечення є ефективним підходом до створення інформаційних систем для медичних закладів, що сприяє підвищенню якості управління, зменшенню навантаження на персонал та покращенню доступу до медичної інформації.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Класифікація медичних інформаційних систем. [Електронний ресурс]: – Режим доступу: <https://www.softwaresuggest.com/blog/types-of-health-information-systemexamples/>
2. Електронна система охорони здоров'я eHealth в Україні. [Електронний ресурс]: – Режим доступу: <https://ehealth.gov.ua/>
3. Agile software development in healthcare: in-depth guide. [Електронний ресурс]: – Режим доступу: <https://decode.agency/article/agile-healthcare-software-development/>
4. Kokol, P. Agile SoftwareDevelopment in Healthcare: ASynthetic Scoping Review. Applied Sciences .2022, 12, 9462. doi: [10.3390/app12199462](https://doi.org/10.3390/app12199462)
5. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2020. – 192 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/c136860d-44cb-4f05-adaf-dcdd20830483/content>
6. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник. Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с. – Режим доступу: <https://files.znu.edu.ua/files/Bibliobooks/Inshi72/0053479.pdf>
7. Яковенко В. О., Ульяновська Ю. В., Костенко В. В., Костенко Д. Є., Лавренюк І. В., Молотков О. Н. Основи автоматизованого проектування складних об'єктів та систем : навчальний посібник. Дніпро: Університет митної справи та фінансів, 2018. – 114 с. – Режим доступу: [http://biblio.umsf.dp.ua/jspui/bitstream/123456789/4330/1/Навч\\_пос\\_ОАП\\_10\\_2018.pdf](http://biblio.umsf.dp.ua/jspui/bitstream/123456789/4330/1/Навч_пос_ОАП_10_2018.pdf)
8. Простий посібник зі схем UML і моделювання баз даних [Електронний ресурс]: – Режим доступу: <https://www.microsoft.com/uk-ua/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>

**Сташенко Софія Олександрівна** – студентка групи СА-23б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [stasenkosofia178@gmail.com](mailto:stasenkosofia178@gmail.com)

**Войцеховська Ольга Олександрівна** – PhD, доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, e-mail: [olgav1085@gmail.com](mailto:olgav1085@gmail.com)

**Stashenko Sophia O.** – student of Faculty of Intelligent Information Technology and Automation, SA-23b, Vinnytsia National Technical University, Vinnytsia, e-mail: [stasenkosofia178@gmail.com](mailto:stasenkosofia178@gmail.com)

**Voitsekhovska Olha O.** – PhD, Associate Professor of the Department of System Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: [olgav1085@gmail.com](mailto:olgav1085@gmail.com)