

## ЛАБОРАТОРНА РОБОТА № 1

*"Розробка системи моніторингу стану серверного обладнання та сповіщення про інциденти"*

**Мета:** Визначення варіантів використання та структури системи.

**Хід роботи:**

**Бізнес-вимоги:**

- 1. Основні цілі:** Створити програмний комплекс для автоматизованого збору телеметричних даних, виявлення аномалій у роботі серверного обладнання та мінімізації часу простою через оперативне сповіщення відповідальних осіб.
- 2. Представлення проєкту:** Проєкт буде реалізовано у вигляді веб-орієнтованої системи (Dashboard), що візуалізує метрики в реальному часі та надає інструменти для управління інцидентами.

**Вимоги користувачів:**

- 1. Можливість перегляду стану системи:** Користувач повинен бачити зведену панель (Dashboard) з ключовими показниками (CPU, RAM, температура, диск) по всіх підключених серверах.
- 2. Можливість налаштування порогів (Thresholds):** Користувач має встановлювати критичні значення (наприклад, температура  $> 85^{\circ}\text{C}$ ), при досягненні яких система реагує.
- 3. Можливість отримання сповіщень:** Система повинна надсилати повідомлення (Email/Telegram) відповідальному інженеру при виникненні інциденту.
- 4. Можливість управління інцидентами:** Користувач повинен мати змогу переглянути деталі збою, взяти його в роботу та змінити статус на «Вирішено».

					ДУ «Житомирська політехніка».22.121.31.000 – Лр1						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Ярошинський С.В.			Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Перевір.		Левківський В.Л.								1	3
Керівник								ФІКТ Гр. ІПЗ-22-2[1]			
Н. контр.											
Зав. каф.											

**5. Можливість генерації звітів:** Отримання PDF-звіту про стабільність роботи обладнання за обраний період (день, тиждень, місяць).

**6. Можливість управління доступом:** Розмежування прав доступу для перегляду даних та зміни налаштувань конфігурації.

**Функціональні вимоги:**

**1. Авторизація та аутентифікація:** В системі повинна бути реалізована безпечна авторизація користувачів із розподілом ролей (Адміністратор, DevOps-інженер, Гість/Менеджер).

**2. Збір та обробка даних:** Система повинна приймати дані у форматі JSON від агентів моніторингу (скриптів-імітаторів) через REST API.

**3. Детекція аномалій:** Система повинна автоматично порівнювати вхідні дані з налаштованими правилами та створювати запис про інцидент у базі даних.

**4. Історія метрик:** Збереження історичних даних для побудови графіків динаміки змін показників за минулі періоди.

**Нефункціональні вимоги:**

**1. Сприйняття та інтерфейс:**

- a. Інтерфейс має бути інтуїтивно зрозумілим, з використанням графічної візуалізації (діаграми, графіки).
- b. Час відгуку інтерфейсу при оновленні графіків не повинен перевищувати 2 секунди.

**2. Надійність:**

- a. Система повинна працювати в режимі 24/7.
- b. У разі збою бази даних система має відновити роботу без втрати конфігураційних налаштувань.

**3. Продуктивність:**

- a. Система повинна підтримувати одночасне оновлення метрик від мінімум 50 серверів без затримок.

**4. Експлуатація та масштабування:**

		Ярошинський С.В.			ДУ «Житомирська політехніка».22.121.31.000 – Лр1	Арк.
		Левківський В.Л.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

- a. Можливість легкого додавання нових серверів для моніторингу без перезавантаження всієї системи.
- b. Архітектура клієнт-сервер (Python/Flask + JS).

**Ролі в системі:**

1. **DevOps-інженер (Оператор):** Основний користувач. Переглядає графіки, реагує на сповіщення, закриває інциденти.
2. **Адміністратор системи:** Має повний доступ. Додає нові сервери в систему, створює користувачів, налаштовує глобальні конфігурації.
3. **Менеджер (Спостерігач):** Має доступ лише на читання. Переглядає загальні звіти про Uptime (час безвідмовної роботи) та ефективність.
4. **Сенсор/Агент (System Actor):** Це не людина, а скрипт, який автоматично надсилає дані на сервер.

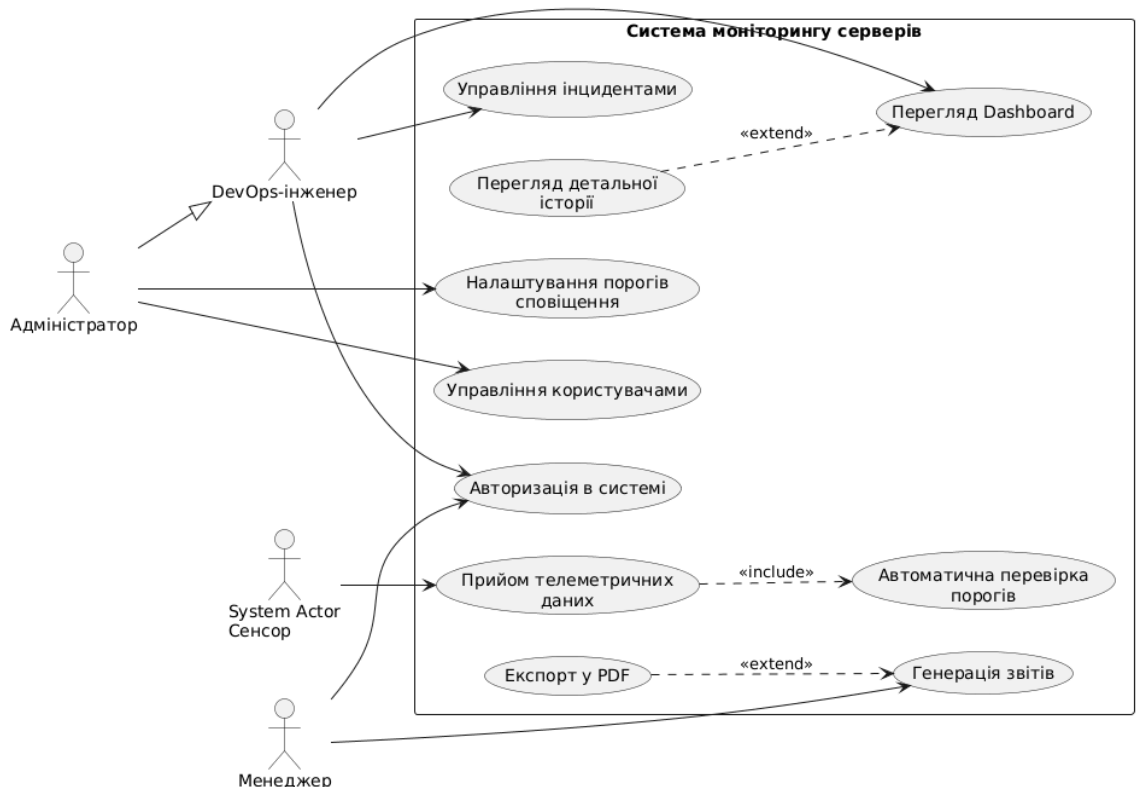


Рис. 1. Створена діаграма

**Висновок:** у ході виконання лабораторної роботи ми дослідили математичну модель нейрона.