

ДОДАТОК А

Рисунки та схеми

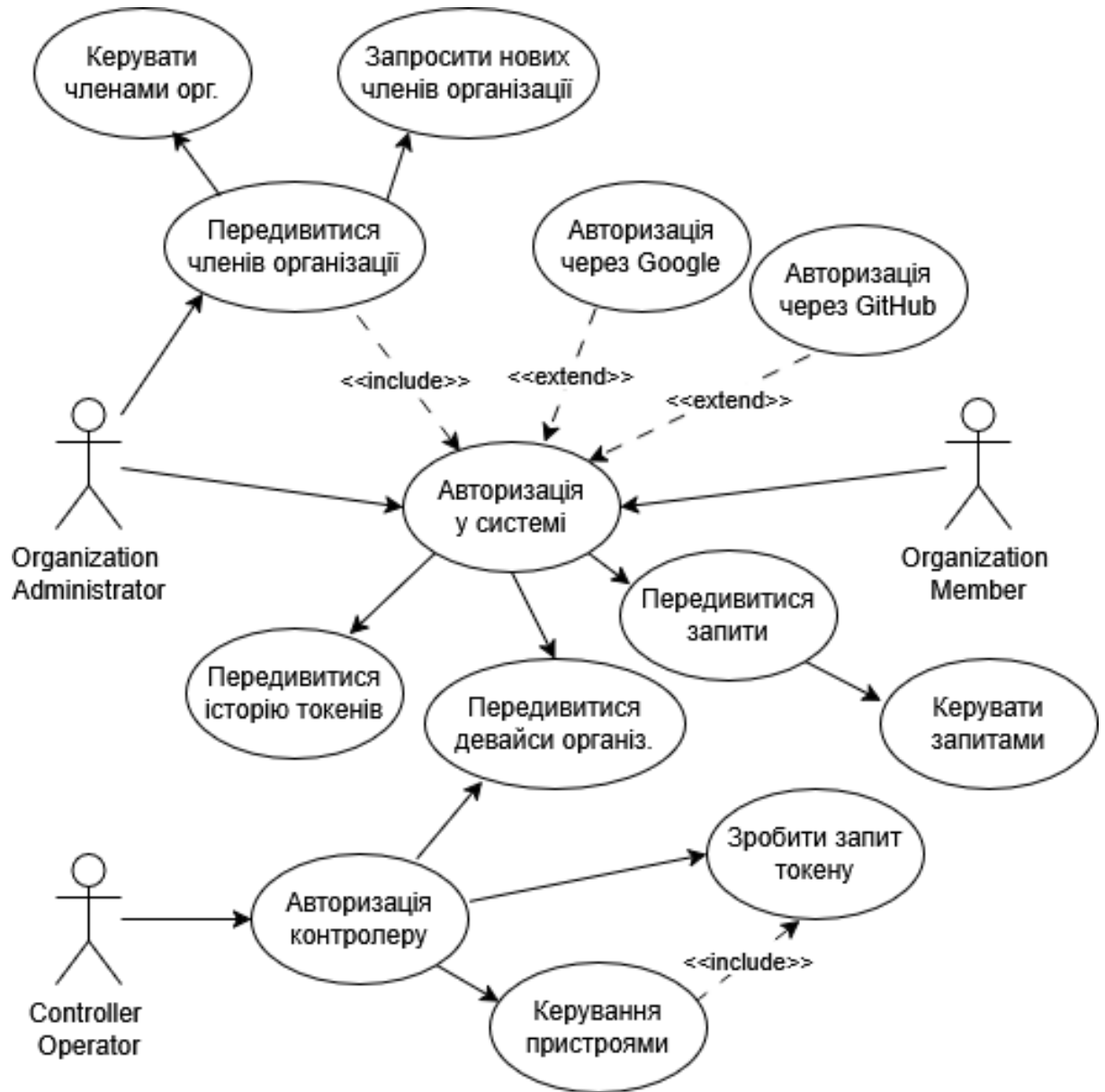


Рисунок А.1 – Use Case діаграма

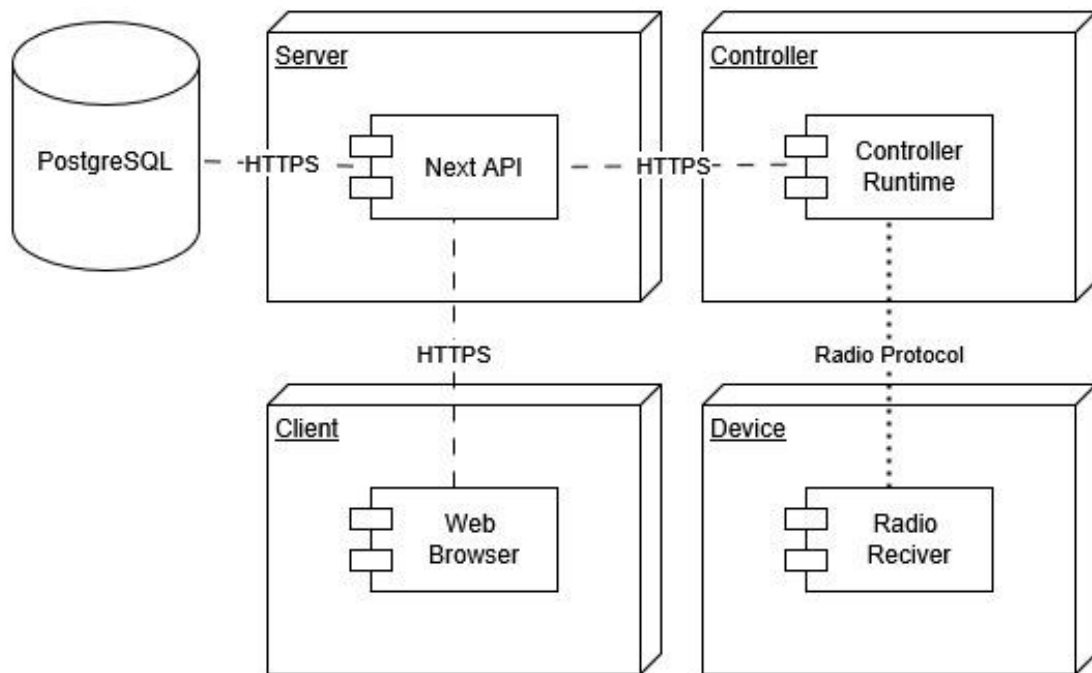


Рисунок А.2 – Діаграма розгортання

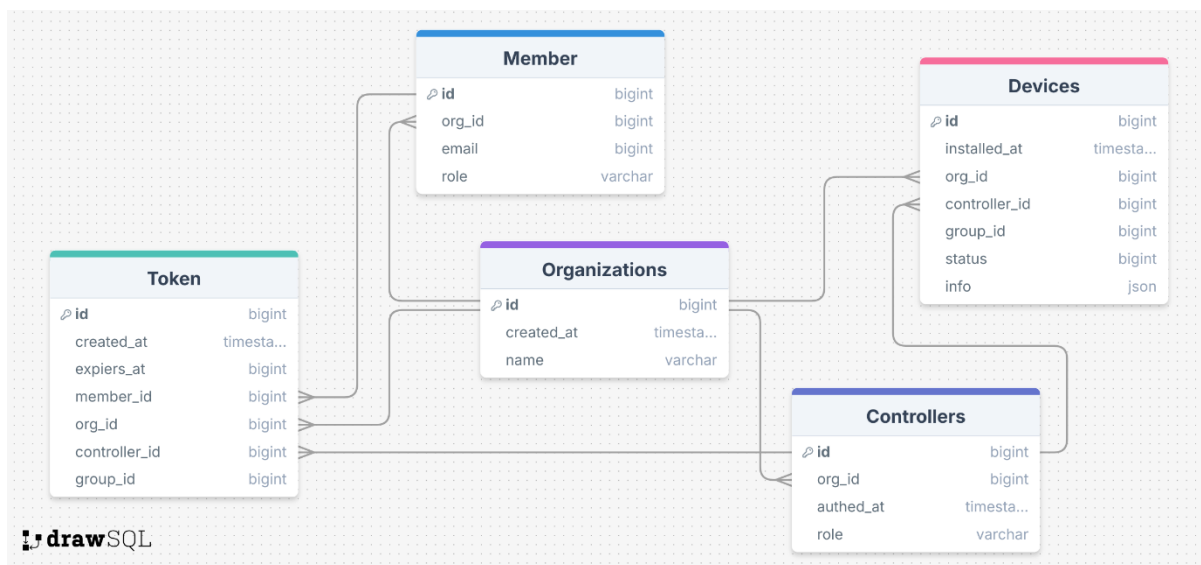


Рисунок А.3 – Схема бази даних

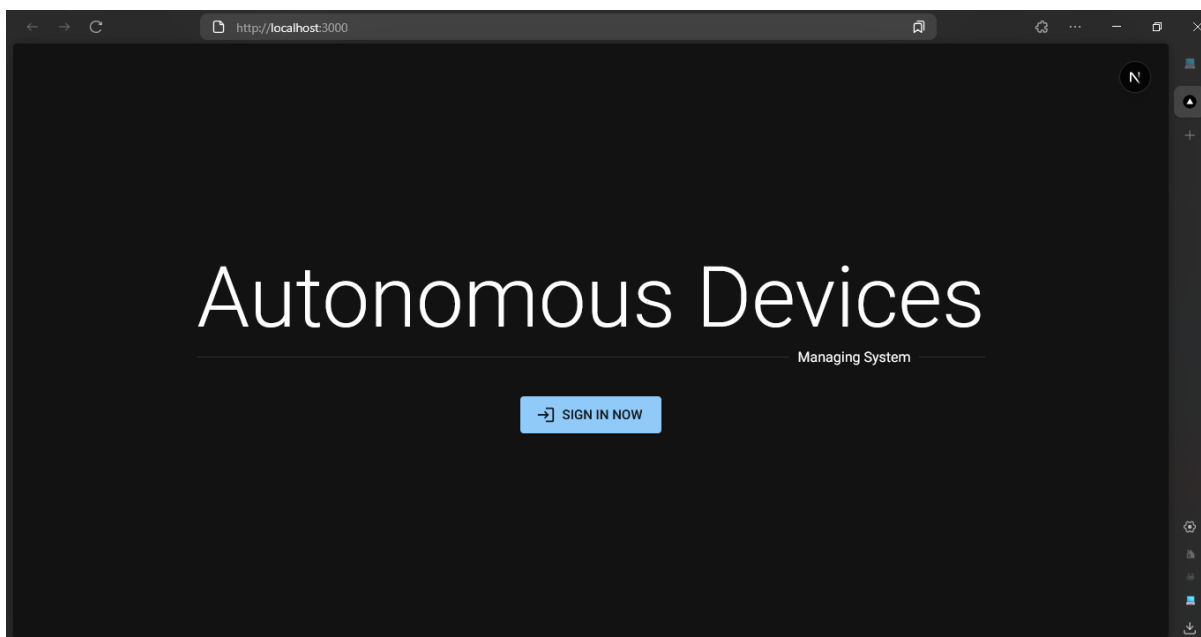


Рисунок А.4 – Вітальна сторінка веб-застосунку

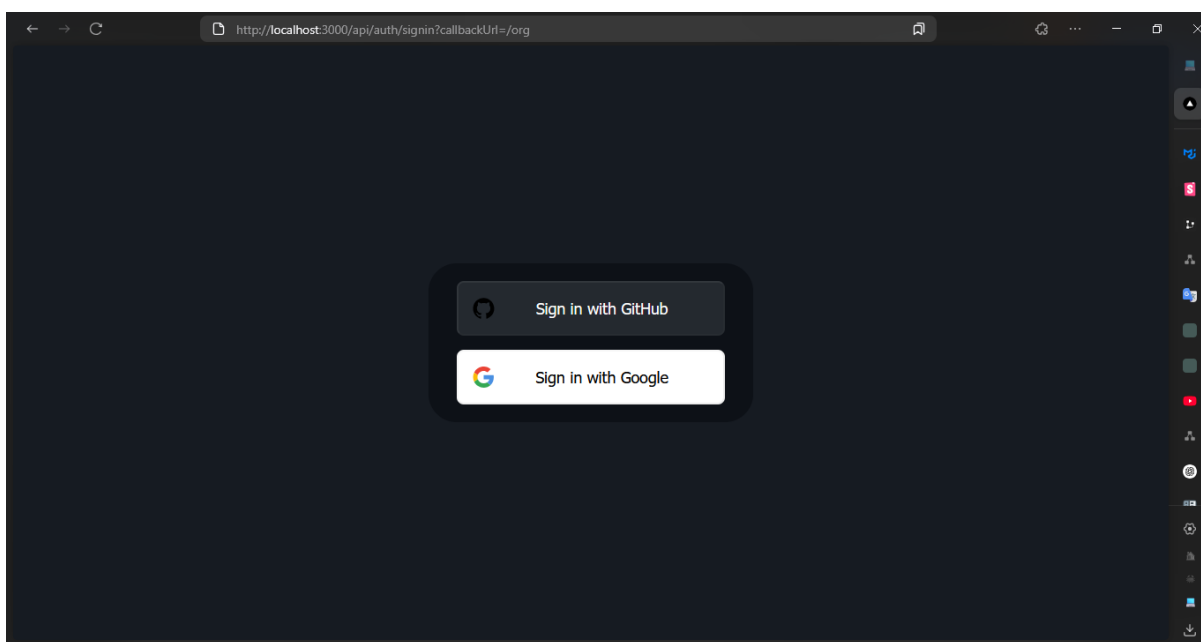


Рисунок А.5 – Сторінка авторизації

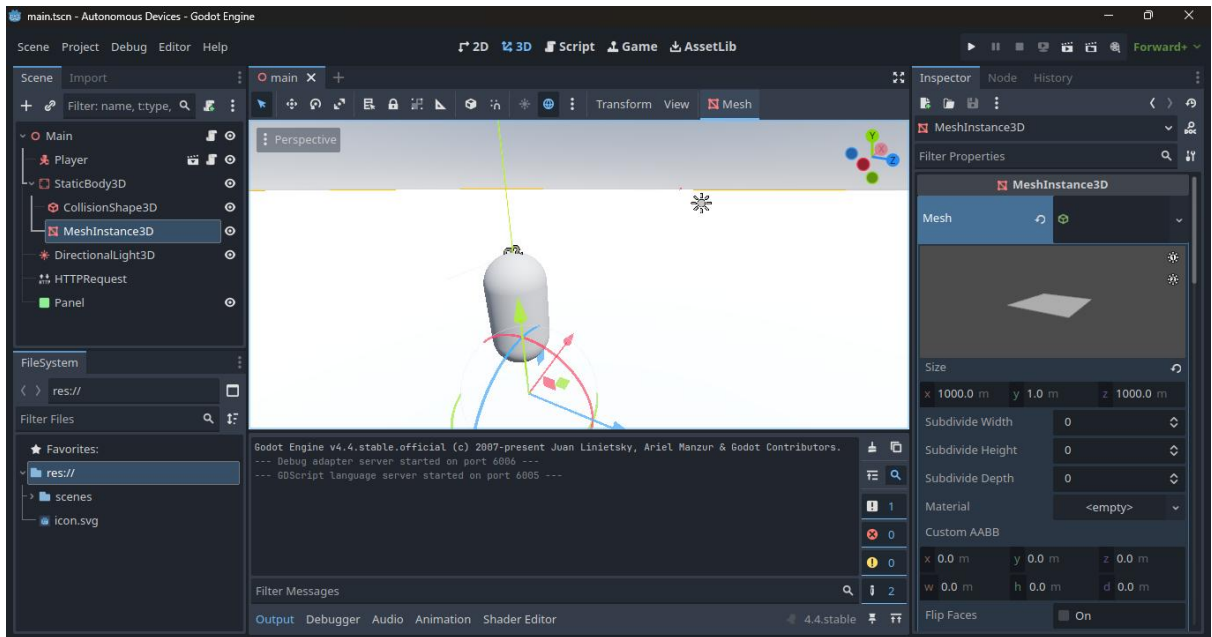


Рисунок А.6 – Середовище розробки Godot

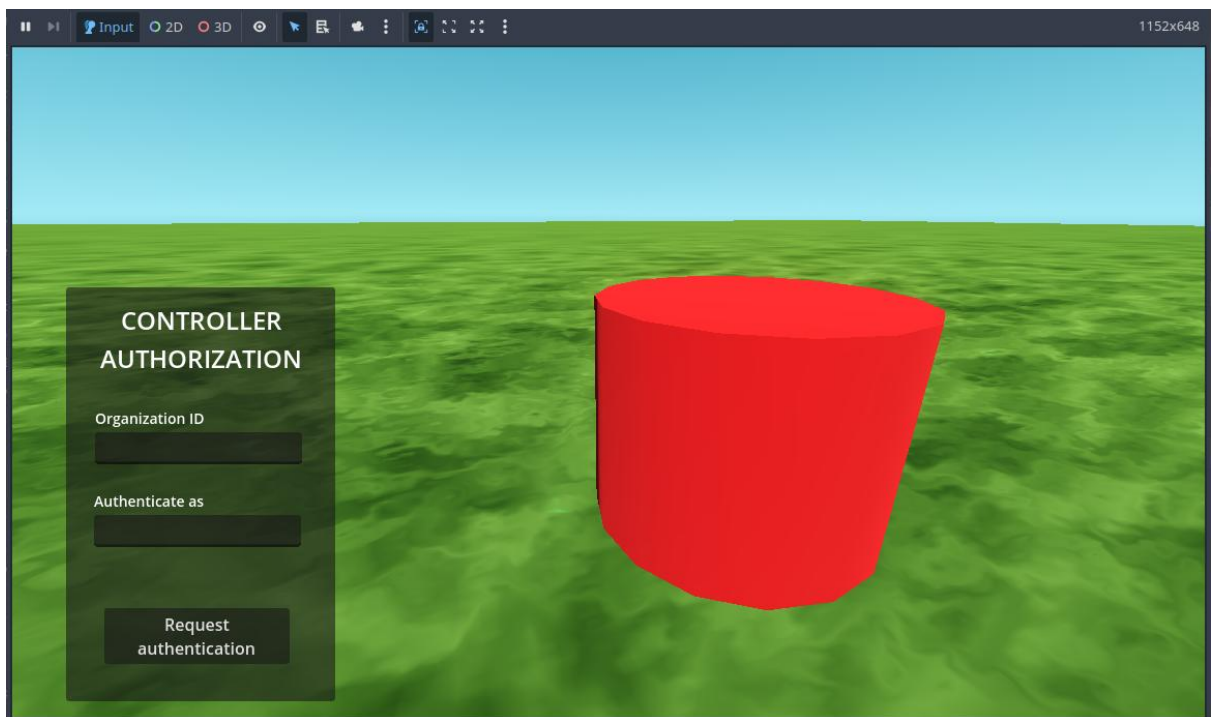


Рисунок А.7 – Панель авторизації токену та репрезентація девайсу

ДОДАТОК Б

Таблиці

Таблиця Б.1 – Колекція REST API кінцівок

URL	HTTP Метод	Опис функціоналу
/api/devices/[orgId]	GET	Для отримання даних про девайси організації
/api/devices	POST	Для надсилання інформації про новий девайс на сервер
/api/request	POST	Для надсилання запиту на сервер
/api/request/[id]	GET	Для отримання інформації про запит та перевірки його статусу
/api/auth/[...]	GET, POST	Резервовано для обробки системи авторизації клієнтів на вебсайті

ДОДАТОК В

Приклади коду

В.1 Генерація ключів:

```
import * as crypto from "crypto";

export function GenKeys() {
  const keys = crypto.generateKeyPairSync("rsa", {
    modulusLength: 2048,
  });

  return {
    public: keys.publicKey
      .export({ format: "pem", type: "pkcs1" })
      .toString(),
    private: keys.privateKey
      .export({ format: "pem", type: "pkcs1" })
      .toString(),
  };
}
```

В.2 Шифрування токєну:

```
export function encrypt(privateKey: string, data: string) {
  const encryptedData = crypto.privateEncrypt(
    {
      key: privateKey,
      padding: crypto.constants.RSA_PKCS1_OAEP_PADDING,
      oaepHash: "sha256",
    },
    Buffer.from(data),
  );

  return encryptedData;
}
```

В.3 Дешифрування токєну:

```
export function decrypt(publicKey: string, data: any) {
  const decryptedData = crypto.publicDecrypt(
    {
      key: publicKey,
      padding: crypto.constants.RSA_PKCS1_OAEP_PADDING,
      oaepHash: "sha256",
    },
    data,
  );

  return decryptedData;
}
```

В.4 Приклад створення таблиці організації:

```
export const orgTable = sqlTable("org", {
  id: int().primaryKey({ autoIncrement: true }),
  name: text().notNull(),
  epoch: int("epoch", { mode: "timestamp" })
    .notNull()
    .default(sql`(strftime('%s', 'now'))`),
});
```

В.5 Код будування вітальної сторінки:

```
import SignIn from "@components/SignIn";
import { Typography, Divider } from "@mui/material/";



export default function Home() {

  return (
    <main className="flex h-[100vh] justify-center">
      <div className="m-auto w-fit text-center">
        <Typography variant="h1">Autonomous Devices</Typography>
        <Divider textAlign="right">Managing System</Divider>


        <div className="h-8" />
        <SignIn />
      </div>
    </main>
  );
}
```

ДОДАТОК Г

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

Дата звіту 6/10/2025
Дата редагування ---


Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kharkiv National University of Radio Electronics

Заголовок
2025_Б_ПІ_ПЗПІ-21-8_Синельников_О_О_скорочений

Автор Науковий керівник / Експерт
Синельников Олександр Олександрович Євген Кардаш

підрозділ
каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

0.00%
0.00%

КП 1

0.98%
0.98%

КЦ

25

Довжина фрази для коефіцієнта подібності 2

3758






Кількість слів

28760

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		0



Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.


10 найдовших фраз			Копію тексту
порядковий номер	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
з бази даних RefBooks (0.00 %)			■
порядковий номер	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
з домашньої бази даних (0.00 %)			■
порядковий номер	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
з програми обміну базами даних (0.00 %)			■

ДОДАТОК Д

Слайди презентації



МІНІСТЕРСТВО
ОСВІТИ І НАУКИ
УКРАЇНИ




ХАРКІВСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНИКИ

Програмна система управління поведінкою пристроїв на заданих ділянках

Керівник:
Доцент кафедри ПП
Лещинський В.О.

Виконав:
Студент групи ПЗПІ-21-8
Синельніков О.О.



19 червня 2025

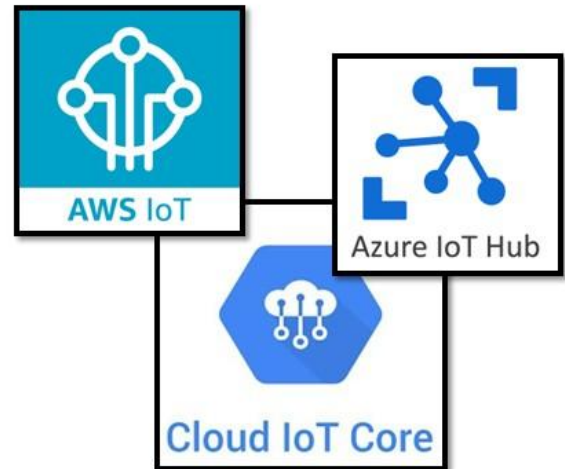
Мета роботи

- IoT девайси в кожній сучасній промисловій галузі
- Тренд на створення “смарт” девайсів

Проектування та реалізація програмної системи для керування пристроями без постійного підключення до Інтернету з можливістю виконувати свої основні функції автономно (вмикання ліхтарів, полив, військове застосування, тощо)

Аналіз готових рішень

- Готове комплексне рішення
- Частина більшої екосистеми
- Масштабованість
- Безпека
- Постійне підключення до Інтернету
- Обмежена конфігурація
- Не розраховано на роботу з обмеженими ресурсами



Альтернативні рішення

- Підтримує альтернативні способи зв'язку
- Порівняно невелика вага
- Слабкий рівень безпеки
- Погана масштабованість
- Розрахована на любителів, а не використання в індустріальному середовищі

MySensors

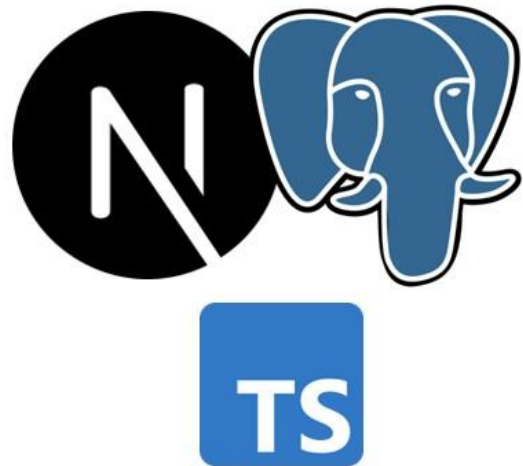
Постановка задачі

Спроектувати та реалізувати програмну систему керування автономними пристроями з можливостями:

- Додавання девайсів
- Моніторинг стану пристроїв та перегляд їх на мапі
- Зміна стану девайсів
- Делегація повноважень через запити

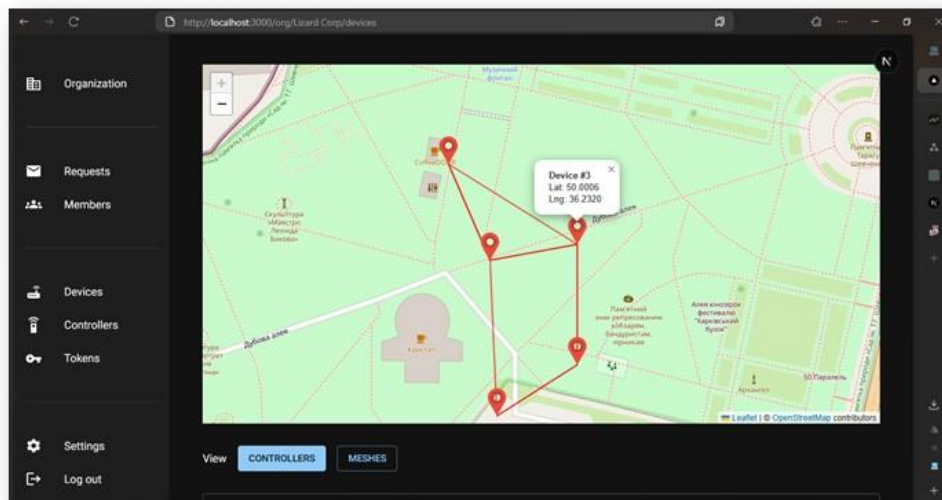
Технології розробки

- Node.js, Next.js, React.js,
- PostgreSQL, Drizzle ORM
- TypeScript



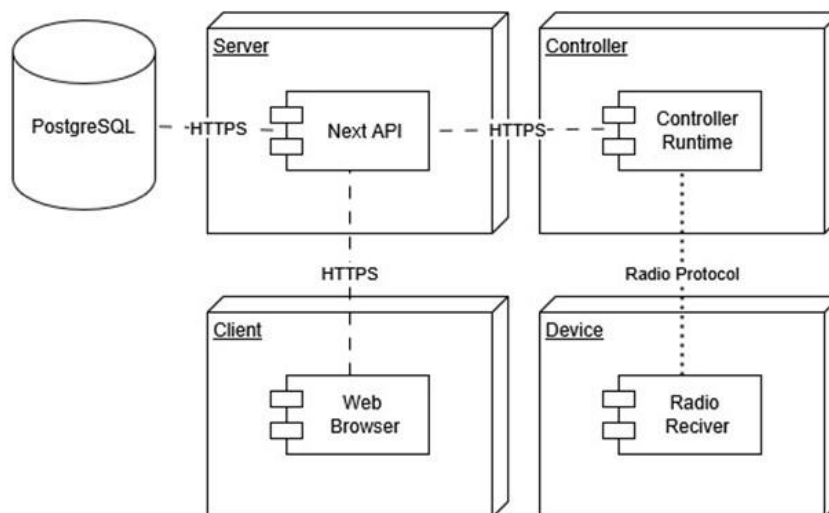
Інтерфейс користувача

- TSX
- Material UI
- Tailwind



7

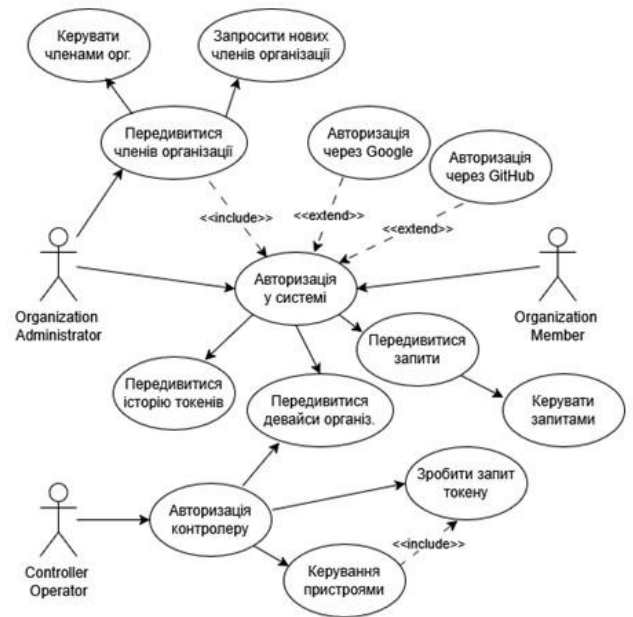
Архітектура системи



8

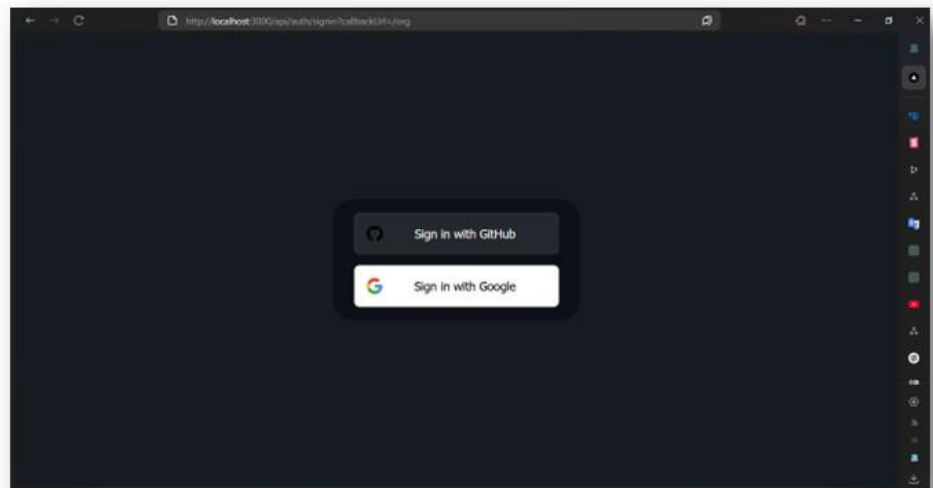
Ролі користувачів

- Оператор контролеру
- Член організації
- Адміністратор організації



Авторизація (Auth.js)

- GitHub
- Google
- Пошту



Генерація токену

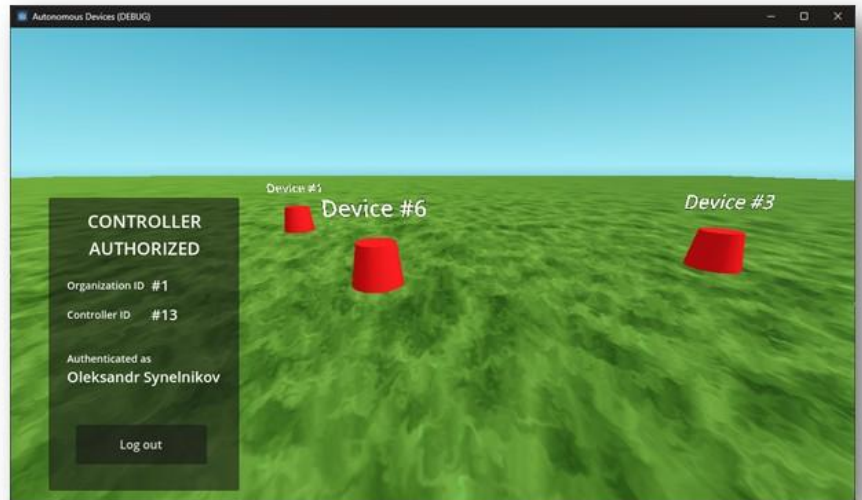
Event	ID	Timestamp
Controller Authentication	#11	17.06.2025 19:13
Controller Authentication	#10	17.06.2025 18:39
Controller Authentication	#9	17.06.2025 12:00
Token Generation	#8	16.06.2025 22:58
Controller ID: #13 Oleksandr Synelnikov Kharkiv Oblast		
Mesh ID: #1 Autonomous Lighting System Kharkiv Oblast		
Permissions: Install, remove, modify behavior, turn on, turn off		
Controller Authentication	#7	16.06.2025 22:57
Token Generation	#6	16.06.2025 22:52
Controller Authentication	#5	16.06.2025 22:18



```
1 import * as crypto from "crypto";
2
3 export function GenKeys() {
4   const keys = crypto.generateKeyPairSync("rsa", {
5     modulusLength: 2048,
6   });
7
8   return {
9     public: keys.publicKey
10    .export({ format: "pem", type: "pkcs1" })
11    .toString(),
12    private: keys.privateKey
13    .export({ format: "pem", type: "pkcs1" })
14    .toString(),
15  };
16 }
17
18 export function encrypt(privateKey: string, data: string) {
19   const encryptedData = crypto.privateEncrypt(
20     {
21       key: privateKey,
22       padding: crypto.constants.RSA_PKCS1_OAEP_PADDING,
23       oaepHash: "sha256",
24     },
25     Buffer.from(data),
26   );
27
28   return encryptedData;
29 }
30
```

Симуляція

Симуляція програмної системи в реальному часі для аналізу процесів та демонстрації роботи системи в реальному світі

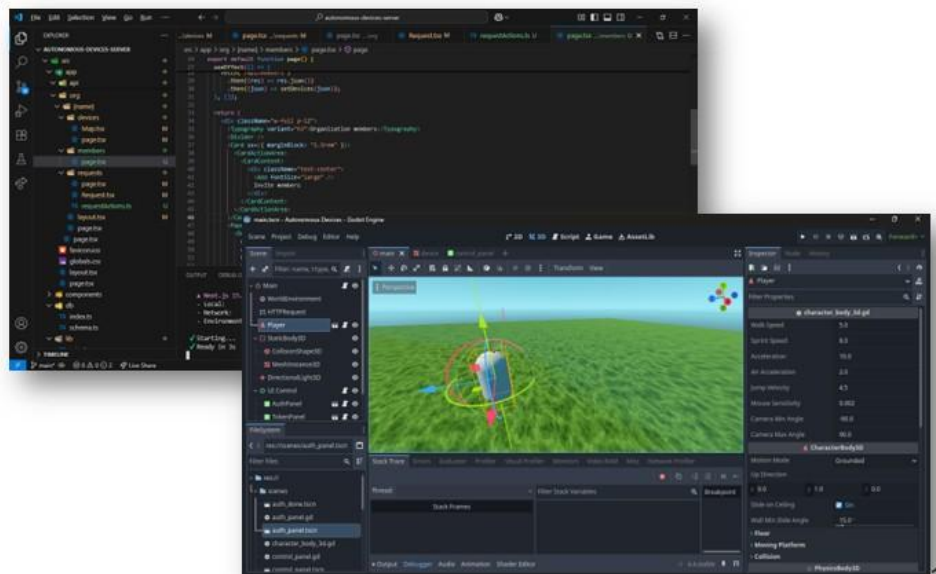


Тестування

- Мануальне тестування: всі функції протестовані
- Тестування API через Insomnia
- Базове модульне тестування з використанням бібліотеки Jest

Середовище розробки

- Editor VSCode
- Godot Engine
- Git, npm



Підсумки

- Проаналізовано предметну область
- Сформульовано постановку задачі
- Проведено аналіз вимог програмної системи
- Реалізовано за допомогою технологій:
 - Node.js – сервер
 - Next.js – веб-застосунок
 - PostgreSQL – база даних
 - Godot – симуляція

ДОДАТОК Е
Специфікація програмного продукту

СПЕЦИФІКАЦІЯ ПЗ

Програмна система управління поведінкою пристроїв на
заданих ділянках

Виконав:

Студент групи ПЗПІ-21-8

Синельніков Олександр

Харків 2025

1 ВСТУП

1.1 Огляд продукту

Програмна система призначена для керування автономними пристроями на ізольованих територіях без постійного інтернет-з'єднання. Основу складають сервер з REST API, веб-застосунок для адміністраторів та контролери з можливістю локальної авторизації і управління. Система використовує LoRa для радіозв'язку та реалізує захищені токени доступу з обмеженим терміном дії.

1.2 Мета

Метою розробки є створення надійної та безпечної системи для керування пристроями без постійного підключення до інтернету. Керування ними буде виконано локальними операторами окремих мереж за допомогою певного девайсу контролеру, якому буде надавати тимчасовий доступ сервер у вигляді токenu з автономним підтвердженням його дійсності девайсом.

1.3 Межі

Програмна система не включає в себе виробництво або фізичне налаштування контролерів та автономних пристроїв. Вона розрахована на взаємодію із зовнішніми девайсами через стандартизовані протоколи. У разі відсутності живлення або повного виходу з ладу пристрою – відповідальність за дії переходить на рівень апаратного забезпечення.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Система є частиною інфраструктури управління автономними системами. Вона передбачає можливість масштабування, гнучкого розгортання та використання у різних галузях: сільське господарство, безпека, енергетика, тощо.

2.2 Функції продукту

- Авторизація користувачів та організацій.
- Управління пристроями через веб-інтерфейс.
- Надання тимчасових прав доступу через токени.
- Локальне керування автономними пристроями.
- Моніторинг статусу пристроїв.
- Збереження та перегляд історії дій.

2.3 Характеристики користувачів

– Член організації керує системою через веб застосунок. Після авторизації в системі, йому буде надано доступ до панелі керування організацією та її пристроями. Головною задачею члена організації є керування запитами, куди входять запити на авторизацію та на генерацію токенів. Член організації також може передивитися інших членів організації, переглянути історію виданих токенів, а також переглянути автономні пристрої організації в форматі списку або на мапі.

– Адміністратор організації – людина, яка створила організацію в системі і відповідає за керування її членами – також має всі повноваження звичайного члену організації, але також може керувати усіма іншими членами організації (запросити, вигнати, змінити роль або посаду), а також налаштуваннями організації в системі (її назву, опис, правила, тощо).

– Оператор контролеру напряду відповідає за контролер – девайс, через який він може керувати автономними пристроями. Для того щоб це робити контролер потрібно авторизувати в системі, а потім отримати токен на керування певними пристроями. Оператор може продивитися усі девайси, що належать організації в виді списку або мапи, а також їх статус та належність до локальних мереж.

2.4 Загальні обмеження

- Використання лише зареєстрованих пристроїв.
- Синхронізації часу між сервером і контролерами.
- Токени мають обмежений термін дії та об'єм доступу.
- Відсутність постійного інтернету може призвести до затримки синхронізації.

2.5 Припущення й залежності

- Контролери та пристрої оснащені LoRa модулями та можуть зчитувати токени та інтерпретувати команди.
- Пристрої працюють в межах локальної радіомережі.
- Всі оператори мають автентифіковані пристрої.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Користувацький інтерфейс

- У системи повинен бути доступний та зрозумілий Web-інтерфейс з підтримкою Material UI
- У системі повинна бути можливість обрати спосіб авторизації через сторонній провайдер
- У членів організації має бути доступний інтерфейс перегляду списку пристроїв та їх розташування на мапі

3.1.2 Інтерфейс з апаратним забезпеченням

- Контролери взаємодіють з автономними пристроями через локальну радіомережу
- Можливість підключення по дроту для аварійного конфігурування

3.1.3 Інтерфейс між модулями системи

- API для взаємодії між контролерами, сервером і базою даних
- Формат повідомлень: JSON

3.2 Властивості програмного продукту

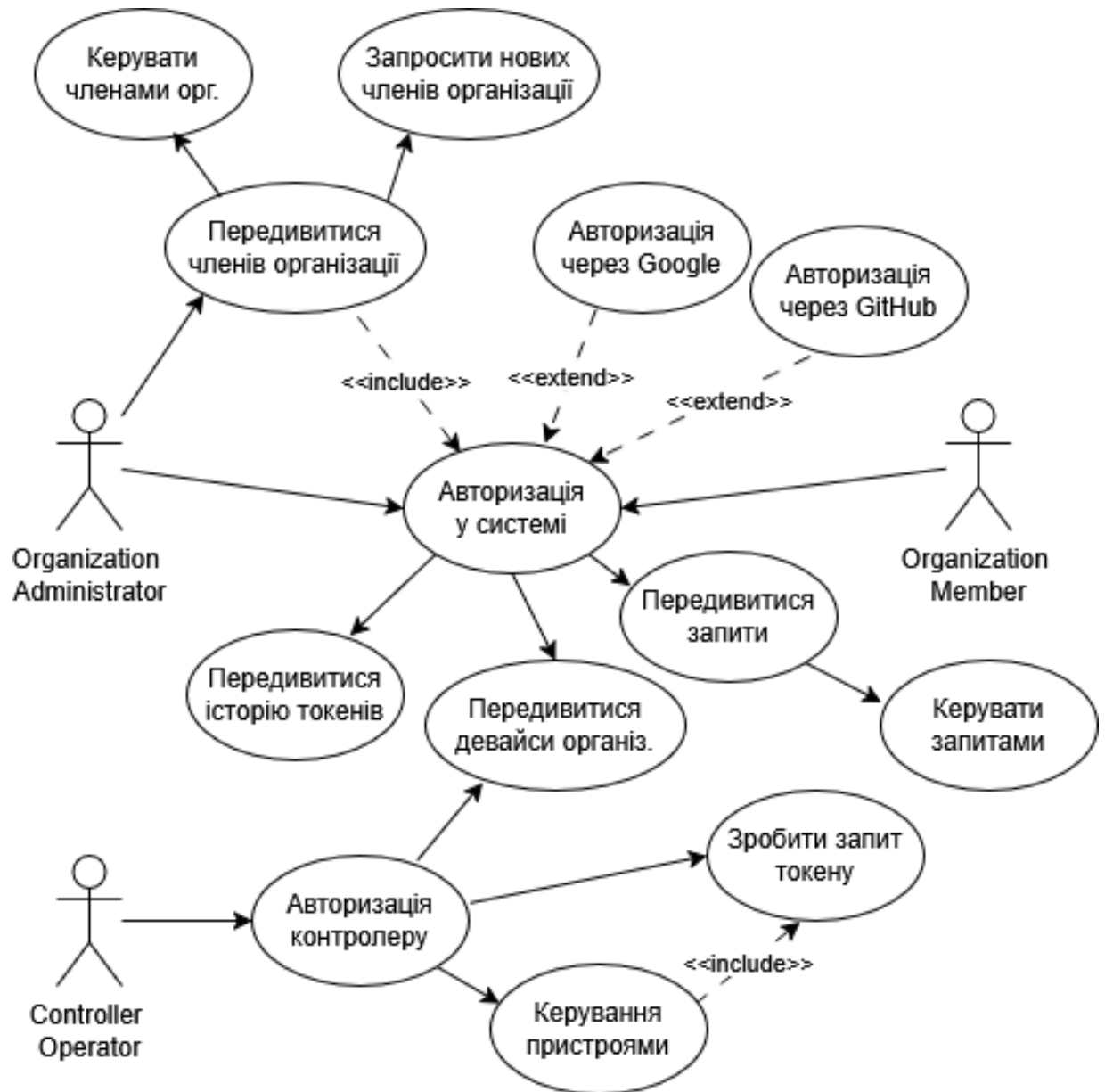
- Керування локальними мережами пристроїв
- Надання асиметрично зашифрованих одноразових локальних токенів доступу
- Запис подій на рівні пристроїв та організацій
- Зберігання даних у PostgreSQL
- Можливість децентралізації через розгортання окремих інстанцій системи

3.3 Атрибути програмного продукту

- Безпека: 2FA, шифрування RSA, блокування при зловживанні
- Надійність: автономна робота без інтернету
- Масштабованість: підтримка багатьох організацій та серверів
- Енергоефективність: ефективне використання заряду автономних девайсів

4 ДОДАТКОВІ МАТЕРІАЛИ

4.1 Діаграма прецедентів



4.2 Таблиця REST API кінцівок

URL	HTTP Метод	Опис функціоналу
/api/devices/[orgId]	GET	Для отримання даних про девайси організації
/api/devices	POST	Для надсилання інформації про новий девайс на сервер
/api/request	POST	Для надсилання запиту на сервер
/api/request/[id]	GET	Для отримання інформації про запит та перевірки його статусу
/api/auth/[...]	GET, POST	Резервовано для обробки системи авторизації клієнтів на вебсайті

4.3 Діаграма розгортання

