

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ


До захисту допущено
Завідувач кафедри ІСТ
_____ Олександр КУЧАНСЬКИЙ
(підпис) (ім'я, ПРІЗВИЩЕ)
“ ____ ” _____ 2022р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

спеціальності 126 «Інформаційні системи та технології»
освітньої програми «Програмні технології інтернет речей»
на тему: «Система управління та обслуговування комунікаціями в парку
імені Пушкіна м. Києва»

Виконав: студент 4 курсу, групи ІР-41

(шифр групи)


_____ Ярослав НІЗОВ _____ 
(Ім'я, ПРІЗВИЩЕ) (підпис)

Керівник к. т. н., асистент Мирослава ГЛАДКА _____
(посада, науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ) (підпис)

Консультант нормоконтроль к. т. н., доц. Ростислав ЛІСНЕВСЬКИЙ _____
(назва розділу) (посада, вчене звання, науковий ступінь, Ім'я, ПРІЗВИЩЕ) (підпис)

Рецензент директор ТОВ «Тест» Андрій ВОВК _____
(посада, науковий ступінь, вчене звання, науковий ступінь, Ім'я, ПРІЗВИЩЕ) (підпис)

Засвідчую, що у пояснювальна записка не має
запозичень з праць інших авторів без відповідних
посилань.

Здобувач освіти _____ 
(підпис)

Київ – 2022 року

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра Інформаційні системи та технології

Освітній рівень Бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітня програма Програмні технології інтернет речей

ЗАТВЕРДЖУЮ

Завідувач кафедри,

д.т.н., доцент

Олександр КУЧАНСЬКИЙ

«__» _____ 2022 року

ЗАВДАННЯ

НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Здобувач освіти: Ярослав НІЗОВ

Група: IP-41

1. **Тема кваліфікаційна робота бакалавра:** «Система управління та обслуговування комунікаціями в парку імені Пушкіна м.Києва ».

Затверджена протоколом засідання кафедри ІСТ №05/21_22 від 03.12.2021 року

2. **Строк подання студентом готової роботи** «23» червня 2022 р.

3. **Вихідні дані до роботи:** дослідження елементів системи автоматизації та моніторингу зелених зон парку імені Пушкіна. Створення таблиць баз даних для подальшого впровадження у розроблений веб-портал

4. **Зміст роботи:** робота складається з чотирьох розділів поетапно виконуючи поставлені задачі у дослідженні аналогів, створенні план схеми, бази даних та відтворенні інтерфейсу.

5. **Перелік графічного матеріалу:** план схема проекту, структурні схеми системи, приклади аналогів елементів системи, фотографії місця дослідження.

6. **Календарний план виконання роботи:**

Етапи виконання кваліфікаційної роботи бакалавра	Термін виконання	Результат виконання
1. Вибір тематики кваліфікаційної роботи бакалавра	01.12.2021	виконано
2. Наказ про затвердження тем кваліфікаційної роботи бакалавра та призначення керівників	03.12.2021	виконано
3. Розробка плану кваліфікаційної роботи бакалавра і його погодження з керівником	25.12.2021	виконано
4. Написання I розділу кваліфікаційної роботи	20.01.2022	виконано
5. Написання II розділу кваліфікаційної роботи	19.02.2022	виконано
6. Написання III розділу кваліфікаційної роботи	05.03.2022	виконано
7. Написання IV розділу кваліфікаційної роботи	05.04.2022	виконано
8. Підготовка висновків і пропозицій	05.05.2022	виконано
9. Попередній захист кваліфікаційної роботи	07.06.2022	виконано
10. Перевірка на плагіат	15.06.2022	виконано
11. Нормоконтроль	17.06.2022	виконано
12. Рецензування кваліфікаційної роботи бакалавра і представлення роботи на кафедрі в друкованому вигляді	21.06.2022	виконано
13. Захист кваліфікаційної роботи бакалавра	23.06.2022	

Дата видачі завдання «03» грудня 2021 р.

Керівник роботи: к. т. н., асистент Мирослава ГЛАДКА _____

(підпис)

Завдання прийняв до виконання:

Здобувач освіти на освітньому рівні «бакалавр» 4-го курсу групи ІР-41

Ярослав НІЗОВ _____



(Власне Ім'я, ПРІЗВИЩЕ)

(підпис)

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра Інформаційних систем та технологій

Освітня програма «Програмні технології інтернет речей»

Кваліфікаційна робота бакалавра Ярослава НІЗОВА

Тема роботи: «Система управління та обслуговування комунікаціями в парку імені Пушкіна м.Києва».

Мета кваліфікаційної роботи бакалавра – розробка автоматизованої системи управління дорожнім рухом з симуляцією потоку трафіку.

Об’єкт дослідження – парк імені Пушкіна за відтвореною план схемою, аналоги елементів системи та платформи реалізації зв’язку елементів.

Предмет дослідження – створення бази даних за результатами аналізу елементів системи та схеми парку для відтворення програмного додатку.

Апробація результатів - за темою дослідження “Сучасні тенденції в розвитку впроваджень «розумних» рішень у парках та зонах відпочинку” була написана наукова публікація для міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами». Також прийнята участь в всеукраїнському конкурсі наукових робіт за темою автоматизації та моніторингу паркових зон.

Кваліфікаційна робота бакалавра складається зі змісту, вступу, основної частини, яка включає чотири розділи, висновків та списку використаних джерел. Всього 90 сторінок.

КЛЮЧОВІ СЛОВА: IoT, Wi-Fi, ПРОЕКТ, ПАРК, ПЛАТФОРМА, ЗЕЛЕНА ТЕРИТОРІЯ, РОЗУМНІ ТЕХНОЛОГІЇ.

ANNOTATION

TARAS SHEVCHENKO KYIV NATIONAL UNIVERSITY

Faculty of Information Technology

Department of Information Systems and Technologies

Educational program "Software technologies of the Internet of Things"

Qualification work of bachelor Yaroslav NIZOV

R&D: " Communications management and service system in Pushkin Park in Kyiv ".

The purpose of the bachelor's qualification work is to develop an automated traffic control system with traffic flow simulation.

The object of research is Pushkin Park.

The subject of research is the creation of a database based on the results of the analysis of system elements and the scheme of the park for the reproduction of the software application.

Approbation of results - on the topic of the study "Modern trends in the development of" smart "solutions in parks and recreation areas" was written a scientific publication for the international scientific and technical Internet-conference "Modern methods, information, software and technical management systems of organizational, technical and technological complexes ». She also took part in the all-Ukrainian competition of scientific works on the topic of automation and monitoring of park areas.

The bachelor's thesis consists of a table of contents, introduction, main part, which includes four sections, conclusions and a list of sources used. Total 90 pages.

KEY WORDS: IoT, Wi-Fi, PROJECT, PARK, PLATFORM, GREEN AREA, REASONABLE TECHNOLOGIES.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ СФЕРИ ЗАСТОСУВАННЯ СИСТЕМИ ОПТИМІЗАЦІЇ ПАРКУ	9
1.1 Проблематика проекту.....	9
1.2 Постановка задачі.....	10
1.3 Місце проектування системи	11
1.4 Минулі оновлення парку	13
1.5 Парки та зони відпочинку	14
1.6 Аналіз існуючого аналогу проекту.....	15
1.7 Українські проекти.....	17
1.8 Створення бізнес моделі.....	19
Висновок до розділу 1.....	20
РОЗДІЛ 2. РОЗГЛЯД АНАЛОГІВ ІоТ ПЛАТФОРМ З ЕЛЕМЕНТАМИ СИСТЕМИ. ВІДТВОРЕННЯ ПЛАН-СХЕМИ ТА БАЗИ ДАНИХ.....	21
2.1 ІоТ платформи та їх використання.....	21
2.2 Розгляд кращих варіантів серед ІоТ платформи	21
2.3 Мета дослідження відносно розглянутих пристроїв	25
2.4 Автоматичний полив зелених територій	26
2.5 Розумні паркові лавочки.....	30
2.6 Технології розумних сміттєвих баків.....	33
2.7 Освітлення парку.....	35
2.8 Визначення вигляду території парку.....	39
2.9 Розгляд програми для побудови схеми	39
2.10 Пояснення побудови схеми парку	41
2.11 Функціональна модель та опис класів для майбутньої бази даних.....	42
2.12 Створення та опис контекстної діаграми.....	44
2.13 Повне представлення діяльності згідно нотації UML	48
Висновок до розділу 2.....	51
РОЗДІЛ 3. СТВОРЕННЯ БАЗИ ДАНИХ ТА РЕАЛІЗАЦІЯ ПРОЕКТУ	53
3.1 Створення бази даних.....	53
3.2 Імпортування баз даних.....	55
3.3 Реалізація інтерфейсу	59

3.4 Реалізація авторизації у системі	64
3.5 Таблиця запитів	67
3.6 Таблиці діючих елементів системи	69
Висновок до розділу 3.....	71
ВИСНОВКИ.....	72
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	74
ДОДАТКИ.....	76

ВСТУП

На сьогоднішній день у Києві ми маємо 127 парків, 500 скверів, 78 бульварів. За котрими доглядають живі люди, яких може не вистачати на усі зелені зони. Для вирішення цієї проблеми краще усього можуть підійти сучасні технології, які автоматизують догляд за територією. Одним із рішень може бути веб-портал, який надає доступ користувачу переглядати девайси розташовані у парку.

Робота над даною темою, спираючись на дані наведені вище, дає розуміння тому, що реалізація цього додатку та системи в цілому є дуже важливою для міста та його населення. Одним з прикладів схожого проекту є відомий парк «Сакер» в Єрусалимі, котрий став учасником зі створення «розумних» парків.

Цей випадок слугує найкращим випадком, можливого інноваційного підходу задля підтримки екологічного стану міста та подальшого розвитку подібних проектів по всій країні. Також, треба зауважити, що саме у цьому проекту головним є застосування інтернет речей.

Об'єктом дослідження є елементи та складові для формування IoT рішення та розробки програмного інтерфейсу “Система управління та обслуговування комунікаціями в парку імені Пушкіна м. Києва”.

Предмет дослідження є елементи для формування розумних рішень щодо створення комфортних умов перебування в місцях громадського відпочинку, функціонування та взаємодія, реакція на події та зміни, керування елементами та програмним інтерфейсом.

Мета роботи полягає в проектуванні інформаційної технології, що поєднує розумні елементи комфортного перебування в громадських місцях в єдиний керований комплекс для забезпечення оперативного управління усіма елементами адміністратором або користувачем; моніторингу стану парку та розташованих в ньому елементів.

Завдання, які постають для виконання полягають в створенні програмного продукту з можливістю моніторингу певного парку та сповіщення робітників прив'язаного до цього парку; проектуванні та моделюванні бази даних, заповненні

таблиць БД значеннями для подальшого маніпулювання з ними та фінальна реалізація з інтерфейсом, який може буде в подальшому розвивати.

Методи дослідження: розгляд аналогів, створення концептуальних діаграм, розрахунок прибутку системи, програмна реалізація створення інтерфейсу для користувача та засвоєння набутих знань та навичок.

У результаті ми отримуємо готовий програмний продукт, що зберігає дані через сховище – базу даних, має користувацький програмний інтерфейс, що дозволяє організувати доступ та проводити маніпуляції з даними даних та прив'язаний до неї програмний продукт, включає в себе керування базою даних як напряду так з можливістю отримувати дані відразу з датчиків, додавання, редагування або видалення даних, моніторинг.

РОЗДІЛ 1. АНАЛІЗ СФЕРИ ЗАСТОСУВАННЯ СИСТЕМИ ОПТИМІЗАЦІЇ ПАРКУ

1.1 Проблематика проекту

Оскільки екологічне становище у місті Києві не є ідеальним, можливо використати новітні технології для покращення ситуації та створення сприятливих умов для населення.

Одним з варіантів може бути проекту, який надає змогу контролювати діяльність зелених зон міста, сповіщати про зміни та проводити моніторинг дистанційно лише однією людиною. Тобто, для покращення перебування населення на території парків та міських зелених зон, доречно використовувати інноваційні пристрої з урахуванням інтернет речей. Такими можуть бути розумні лавочки та смітники, тому що маючи влаштовані сонячні батареї надається можливість заряджати власний пристрій такий як телефон, годинник, чи, навіть, планшет. Датчики наповненості у смітниках можуть збільшити продуктивність робітників парку та зменшити час на перевірку цього кожного разу, коли це не потрібно. Також, камери, котрі слугують для безпеки парку, бо моніторинг можливий у будь-який час з будь-якого пристрою, та запис, котрий можна зберігати у хмарі і витягувати їх, коли потребується. Лазерні датчики можуть бути використані для стеження за зростанням газону, що також зменшує час роботи працівників та може слугувати як система безпеки, коли парк не можна відвідувати з технічних причин. Ще, до цієї схеми можна додати зрошувачі рослин, які можуть налаштовуватись на певний час та кут поливу, щоб не марнувати воду на тротуар. І цього вже буде достатньо до вдалого контролю та моніторингу.

Усе це можливо, насамперед, після розгляду аналогів, створення концептуальних схем, розуміння повної роботи проекту, розробки бази даних та її відтворення у програмному додатку вже з готовим інтерфейсом, але на початку треба зрозуміти головні **проблеми сучасних парків:**

- стара зовнішня та внутрішня інфраструктура;
- залежність від людської праці;

- відсутність систем сповіщення;
- відсутність можливості спостерігати за парком віддалено;
- відсутність покриття території парку зв'язком.

Вдосконалення також потребує програмні реалізації:

- встановлення зв'язку з фізичними технологіями;
- кросплатформеність;
- підтримка різних технологій зв'язку;
- вибір елементів системи з можливістю оновлюватись.

1.2 Постановка задачі

З кожним роком усе більше ми стикаємося з вимогами, що до використання IoT (Internet of Things) в екологічних сферах. Авжеж, бо це невід'ємна частина нашого технічного та інформаційного розвитку, котра при своєму впровадженні може не тільки зберегти наш час, але й зовнішнє середовище навколо нас. Ми можемо використовувати смарт-девайси не помічаючи цього, але - точно знаємо, що вони полегшують наше існування та перебування в різних місцях світу. В наш час, особливо важливим є застосування IoT у поліпшенні умов у парках чи інших зелених територіях, тому що з розвитком корисних технологій, що розвиваються, поширюється можливість застосування нових пристроїв та інноваційних програм, котрі можуть автоматизувати світ навколо нас.

Це надихає на думки, як можна вдосконалювати звичайні нам місця відпочинку чи повсякденного перебування так, щоб вони не тільки були корисними для людини, а й для навколишнього світу. Також, маючи можливість вдосконалюватися з часом не тільки турнікетами чи клумбами, але й технологіями, які з'являються кожен рік. Тому ми повинні використовувати розумні системи, які можуть взаємодіяти між собою для оптимізації та автоматизації процесів, які залежать від людини, що не завжди є правильним рішенням.

Тому, **задачею** дипломної роботи можна вважати створення системи оптимізації та автоматизації роботи парку імені Пушкіна, що дозволить відвідувачам перебувати в комфортних умовах на території парку, а робітникам

виконувати свою роботу не витрачаючи зайвого часу чи тільки слідкувати за справною роботи системи.

Об'єкт дослідження, як вже було зазначено, є парк імені Пушкіна, який не має вже влаштованих інноваційних систем та технологій, та, загалом, має вже застарілу інфраструктуру, окрім споруд, які належать іншим юридичним особам.

Завданням дипломної роботи є розробка концепту проекту з оптимізації системи паркової діяльності, який може слугувати першим етапом відтворення даної системи в реальному світі, якщо дана система з її елементами буде відповідати поставленій задачі наприкінці розробки та буде розцінена, як діюча для впровадження в існуючий парк. Це буде першим кроком до розвитку міста Києва в напрямку IoT технологій поєднаних з екологічними та розважальними напрямками.

Головне питання, які переваги може нести за собою такий проект? Звісно, існують декілька важливих аргументів на користь цього:

- *Економія часу.* Зменшення часу на догляд за великими територіями, як парки, алеї, студентські містечка надає змогу розподілити час працівників на більш важливі речі та надати можливість вирішувати питання, які потребують постійного догляду, не відволікаючись на перевірку приладів чи систем, які працюють самі.
- *Автоматизація та оптимізація процесів.* Коли усі процеси можуть робитися без зайвих доглядачів, це надає змоги економити не тільки витрати людської праці, а й матеріальні витрати, які б повинно було робити при відмові пристроїв та не вчасному догляду за територіями.
- *Можливість вдосконалення та впровадження схожих проектів по всій країні та світу в цілому.* Використання нових технологій спричиняє вдосконалення системи в теперішньому часі та надає змогу робити це у майбутньому. Усі технології оновлюються, а разом із ними, це буде робити і цілий проект.

1.3 Місце проектування системи

Об'єктом дослідження та місцем для теоретичного відтворення проекту було обрано Пушкінський парк(рис. 1.1). Парк[1] розташований у

Шевченківському районі міста Києва, біля станції метро Шулявська, між проспектом Перемоги та кіностудією ім. О. Довженка. Площа території становить 19,45 га.



Рис. 1.1. - Центральний вхід парку імені Пушкіна

Навколо парку розташовані новобудови, котрі мають прямий вид на його території, що є однією з причин, чому потрібно оновити парк сучасними технологіями. Поруч, також, є університети, студенти яких можуть відвідувати ці місця після пар чи в перервах. Але саме головне, що це доволі старий парк, який досі виглядає, як за часи свого заснування.

Також, наведена схема(рис 1.2), що демонструє на які саме зони поділяється територія парку, на прикладі різних кольорів:

- 1) Зелений - зони, які повинні зрошуватись.
- 2) Блакитний - зони відпочинку, для пікніків, відкритих зібрань, тощо.
- 3) Сірий - алеї, які повинні бути обладнані лавочками та смітниками.
- 4) Червоний - споруди, які належать іншим власникам(кафе, ресторани)

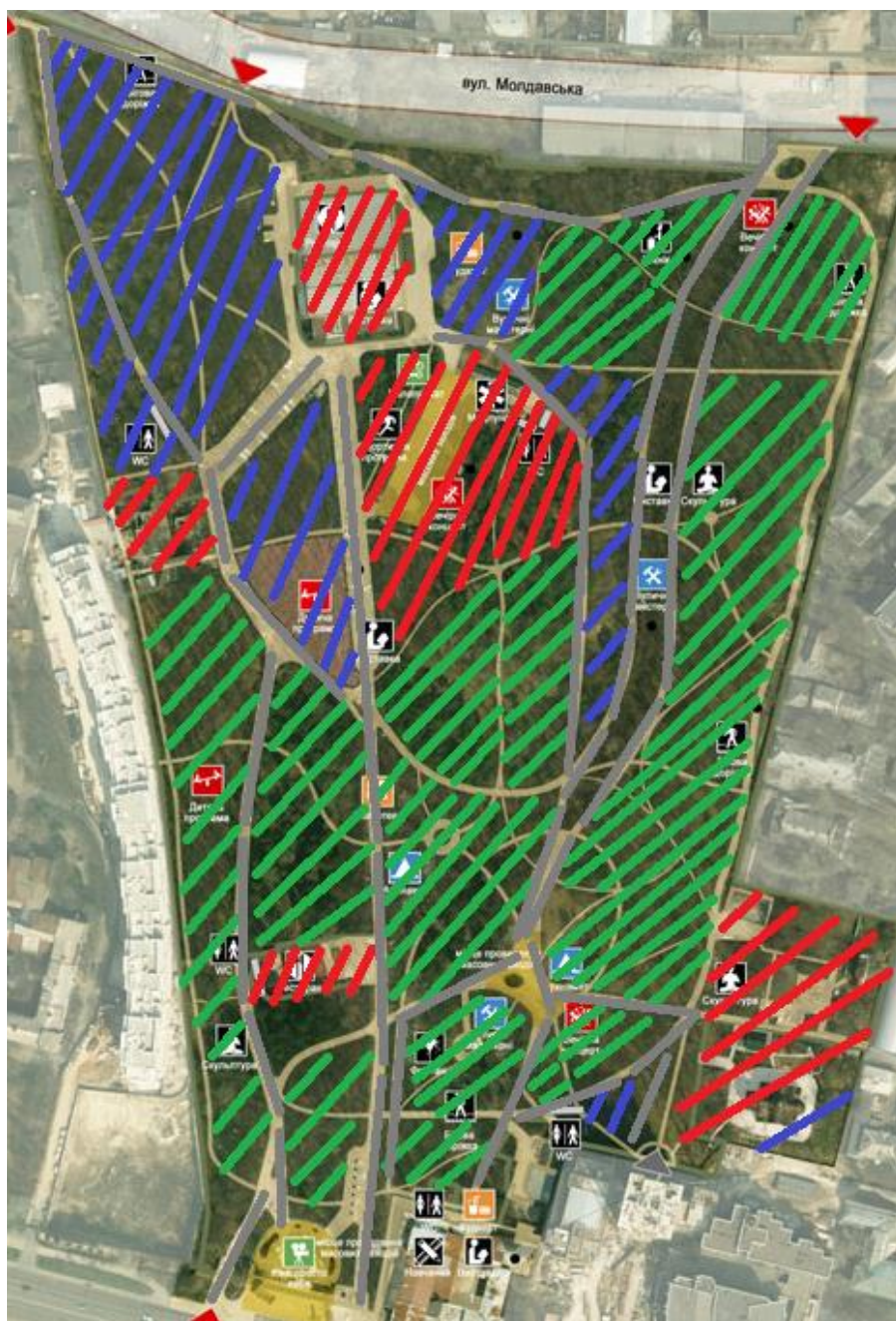


Рис. 1.2. - Схема парку з виділеними зонами

1.4 Минулі оновлення парку

Заснований ще в 1901 році парк зазнавав багато змін тільки з приводу нових насаджень, але цілковитих змін в самій інфраструктурі не було до 21-ого століття, коли там почали відкриватися кафе та був збудований Експо центр, який сьогодні широко використовується для різних виставок.

Парк намагалися оновлювати, додаючи нові об'єкти мистецтва різних художників(рис. 1.3) чи композиції з відсиланням на творчість поета, але проекти

були відкладені та так й не втілилися у життя. Хоч деякі зі споруд митців такі з'явилися, вони втрачаються через мало доглянуті зелені території. У 2007 році у парку було проведено реконструкцію - роботи з благоустрою території, висаджені молоді дерева: червоний дуб, каштан, ясен, горобина, хвойні породи, влаштовані квітники та ландшафтні композиції. Також реконструювали систему зовнішнього освітлення, вимостили доріжки та встановили лавки. Авжеж, ліс посеред міста є доволі цікавою ідеєю, але тільки, якщо він належно доглядається та не витрачає зайвих ресурсів, якщо взагалі витрачає.



Рис. 1.3. - Одна зі споруд парку

Усе ж таки даному парку, потрібно кардинальне оновлення, яке змусить не тільки місцевих жителів відвідувати його, а й людей з усього міста може вражати технологічний та доглянути парк, який буде й далі розвиватися разом з інноваціями та місто у цілому.

1.5 Парки та зони відпочинку

Загалом, парки потрібні для створення у містах зелених зон, яких не вистачає через постійні новобудови та які не можна замінити лише алеями з каштанами у центрі міста. Також, вони є одними з головних місць перебування людей кожен день. Неважливо вихідні чи будні, люди завжди будуть відвідувати парки та не тільки для відпочинку. На великій території можуть проводитись

корпоративні мітинги, концерти на відкритому повітрі, екскурсії чи іншого роду зібрання людей. Але для таких заходів потрібно забезпечити комфортне перебування, яке б сприяло їх постійному проведенню та взагалі притоку відвідувачів.

Тому було сформовано 5 вимог для створення комфортного парку:

- Баланс між насадженнями та спорудами.
- Постійний догляд територій.
- Оновлення обладнання, яке використовується для роботи парку.
- Автоматизована робота елементів системи.
- Використання інноваційних технологій для покращення системи.

1.6 Аналіз існуючого аналогу проекту

Звісно в світі вже є схожі проекти, які були відтворенні різними компаніями, які є спеціалізованими на створенні системи поєднаних з IoT технологіями. Наприклад, технологія компанії Gemalto привела «інтелект» до «Розумного парку».

Система, котру ще називають системою «Інтелектуального парку», побудована на основі технології Інтернет речей (IoT) від компанії Gemalto, та дозволяє підвищити безпеку з комфортом для відвідувачів. Вона складається з пристрою встановленого на території парку чи будь-якої іншої зеленої території та пов'язана з програмним додатком, який потрібен для зв'язку та поширення інформації про стан територій. А саме, завдяки безкоштовним програмам, що завантажуються в смартфони, планшети або інші підключені пристрої (підтримують iOS, Android та Windows), відвідувачі можуть розсилати оновлену інформацію лісникам та співробітникам парку, коли зіштовхуються з заповіданням шкоди природі або коли їм може знадобитися допомога. Також, можуть отримувати текстові SMS-повідомлення з оновленнями інформації про погодні умови, попередження про небезпеку пожеж та перекриття доріг через повені або снігопади. У випадку, коли хтось заблукає або дитина відійде далеко від свого табору чи батьків, співробітники парку негайно про це сповіщаються, а

інструменти геозони встановлені у парку чи заповіднику можуть надати інформацію про точне місцезнаходження для прискорення зусиль з пошуку. Та, навіть, якщо є потреба, можуть негайно надіслати текстові повідомлення сім'ї, коли їх близьких буде знайдено.

Усе це можливо завдяки пристрою встановленому на територіях, де є потрібним допомога в спостережанні за природою та відвідувачами. Установка наведена нижче складається з датчиків руху, відеокамер, датчиків температури, тиску та інших кліматичних умов. Звісно, одна з головних особливостей це використання сонячної панелі для отримання альтернативного виду енергії, щоб жити даний пристрій.



Рис. 1.4. - Пристрій аналізу даних компанії Gemalto

Підключення, як було вже зазначено відбувається через програмний додаток, який в свою чергу використовує технологію Wi-Fi для підключення до мережі.

Значною перевагою цього є можливість отримувати повідомлення про тривожні сигнали в режимі реального часу(рис. 1.4), що значно покращує здатність невідкладних служб швидко реагувати, коли їхня допомога дійсно потрібна, а також розуміння, що конкретно сталося допомагає тому, що вони вже

прибувають підготовленими до конкретної ситуації, незалежно від того, чи потрібні місцеві пожежники та поліція, швидка медична допомога чи додаткові сили, наприклад, служби МНС федерального рівня.

Можливо це, завдяки, програмній платформі SensorLogic компанії Gemalto, яка була відзначена нагородами технологій, які перетворюють великі дані на дані для оперативної діяльності інтелектуальної обробки даних у хмарі. SensorLogic збирає різнотипні дані практично з будь-яких типів датчиків і швидко перетворює їх на, легку в сприйнятті по відношенню до користувача, стандартизовану інформацію, до якої не є важким отримати доступ з центру керування в режимі 24/7/365. А оскільки ця програма працює на Java, вона може дуже легко взаємодіяти з іншими Java-додатками, включаючи програмний агент SensorLogic, вбудований в комунікаційні модулі.

Це, насамперед, є перша у своєму роді інновація, яка допомогла забезпечити високий рівень поінформованості лісникам, співробітникам та відвідувачам. Окрім цього, цю інноваційну технологію можна також застосувати для підвищення безпеки в інших галузях, таких як транспорт та інженерні мережі, охорона громадського порядку та підприємств у всьому світі, через легку змогу налаштування та адаптування до будь-яких умов відповідно до поставлених задач.

1.7 Українські проекти

Треба зауважити, що в Україні наявні схожі проекти, які мають за мету покращити природний стан міста та поєднати міське життя з зеленою його частиною. Одним з прикладів такого проекту можна зазначити станції екомоніторингу, які розташовані у кожному районі міста з метою слідкування та аналізу еко- та метео- даних. Дані станції контролюються людиною, але усі показники збираються автоматично, що дає змогу обробляти дані не перевіряючи станцію кожного разу. Також, уся структура організації розробника(рис. 1.5) цих станцій може функціонувати не втрачаючи час на заплановані перевірки, бо усі

несправності отримуються одразу за наявності, після чого тестуються та виправляються для подальшої роботи станції. Можливість відсилати дані на смартфон, також є перевагою.

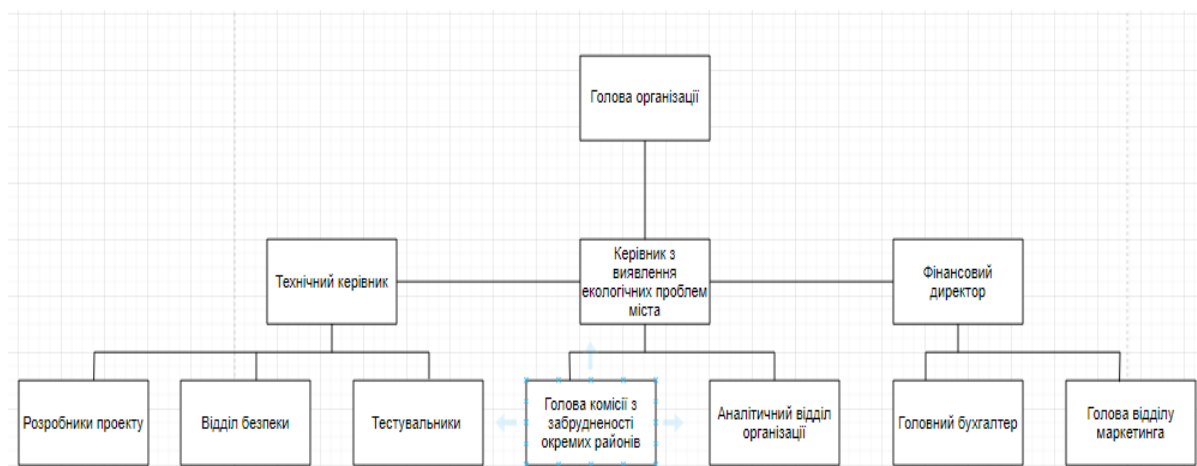


Рис. 1.5. - Структурна схема організації розробника станцій екомониторингу

Ще можна зазначити проекти по налагодженню системи розумних лавочок у студентських парках КПІ та НУБІП. Це можна вважати доволі інноваційною системою направленою на потреби відвідувачів, тобто студентів. Лавочки, які мають сонячні панелі для накопичення енергії та заряду різних девайсів є дуже корисними в такому місці як університет (рис.1.6.).

Схожі проекти є і в інших містах країни, міська адміністрація Полтави впровадила такі ж елементи нових технологій, але додавши до цієї збірки ще й сміттєві баки з влаштованим пресом. Нажаль без датчиків завантаженості чи сонячних панелей, але якщо проект цієї дипломної роботи буде розвинутий далі, то можливе й впровадження більш сучасних елементів.



Рис. 1.6. - Лавка з сонячною батареєю в КПІ

1.8 Створення бізнес моделі

Створення системи моніторингу зелених зон, точно зможе привернути увагу та фінансування у подальшій розробці, через більший масштаб проекту, бо це не тільки окремі розумні лавочки чи сміттєві баки, але й ціла система, яка може модернізуватися та оновлюватися за потребою. Звісно, після створення вже тестової версії, хоча, якщо розглядати далі, то ще потрібно знати як виглядає повна бізнес модель роботи системи (рис.1.6).

Дана модель(рис. 1.7) описує початок створення самого проекту вже з урахуванням причетності державних органів, тому це є повна модель системи, яку можна використовувати після створення баз даних та інтерфейсу, щоб отримати розуміння подальшої взаємодії користувача з додатком.

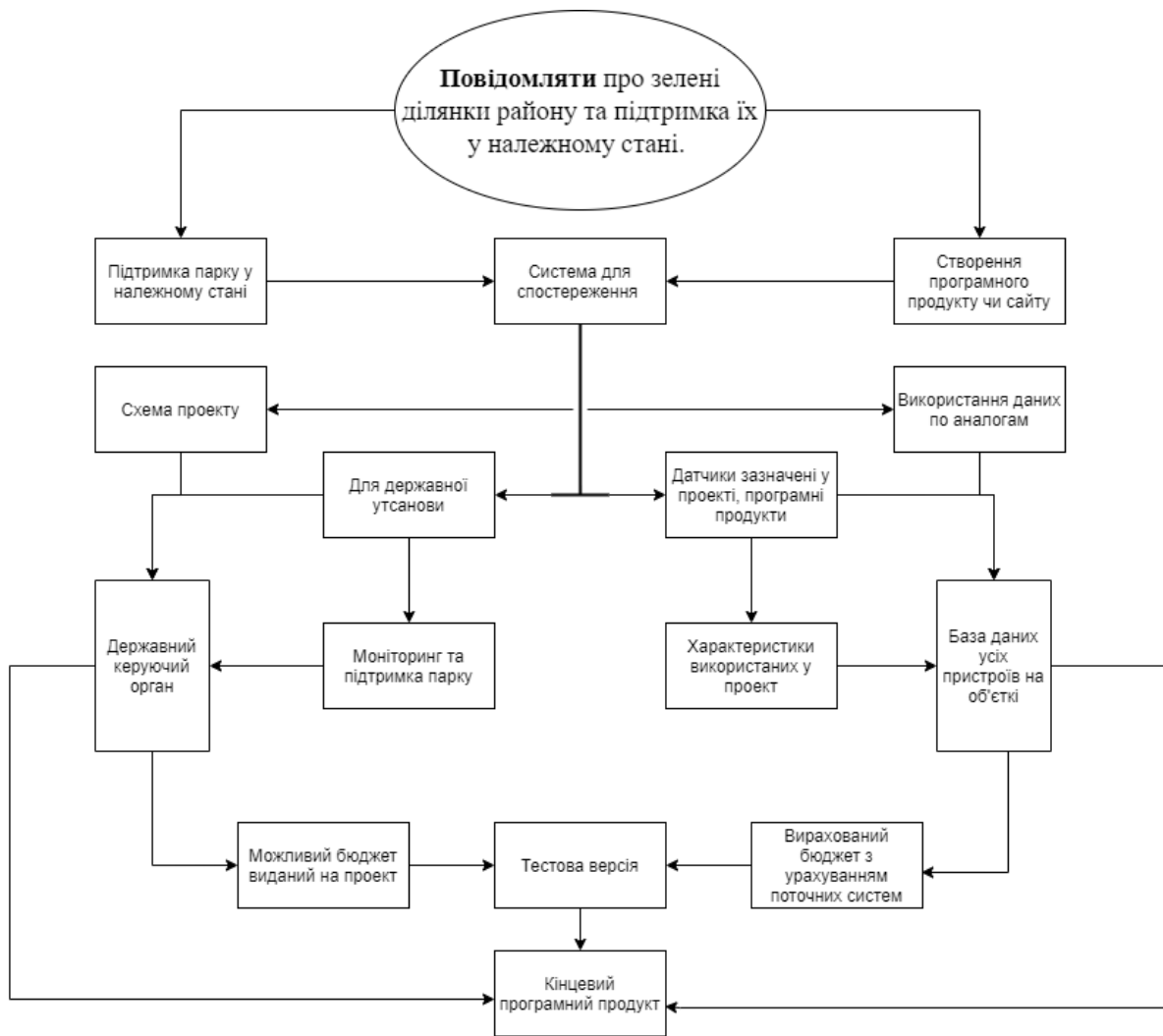


Рис. 1.7. - Повна Бізнес Модель Програмного Продукту

Висновок до розділу 1

Було проаналізовано поставлену задачу дипломного проекту з розглядом схожих за напрямком аналогів, які вже були впровадженні в реальному світі; також, проаналізувавши дані щодо парку імені Пушкіна: історію створення, розташування, проекти, які вже намагалися відтворювати на його території та загальні характеристики парку; було сформовано вимоги для створення комфортних умов для відвідувачів, які повинні відображатися у модернізації території впровадженням системи її оптимізації. Також, розглянуті аналоги проектів системи, яка створюється. Відображення бізнес моделі майбутньої платформи та робота кожного її кроку, враховуючи розподілення обов'язків зовнішніх факторів, які можуть сприяти на вихідні дані від системи.

РОЗДІЛ 2. РОЗГЛЯД АНАЛОГІВ ІОТ ПЛАТФОРМ З ЕЛЕМЕНТАМИ СИСТЕМИ. ВІДТВОРЕННЯ ПЛАН-СХЕМИ ТА БАЗИ ДАНИХ

2.1 ІоТ платформи та їх використання

Для відтворення макету чи цілого ресурсу за стеженням паркової території, збору даних та їх аналізування можна використовувати різні сервіси, ресурси чи платформи. Саме ІоТ-платформи є однією з головних частин проектів з оптимізації та автоматизації будь-чого, які застосовуються різними компаніями усе частіше. Загалом, ІоТ-платформи [2] об'єднують власне речі та інтернет технології. По суті - це ключовий інструмент розробки ІоТ-додатків і сервісів, що поєднує фізичні об'єкти з мережею. При створенні платформи, треба зауважити, що вона завжди має відповідати восьми компонентам.

Перелік усіх восьми компонентів повноцінної ІоТ-платформи, яка може впроваджуватись в систему певного проекту:

- External interfaces.

Дотримавшись відтворення усіх восьми компонентів є можливим досягти розробки ідеальної системи з точки зору інтернет речей.

2.2 Розгляд кращих варіантів серед ІоТ платформи

Оскільки термін ІоТ-платформи не є зовсім новим, можна зрозуміти, що вже є досить багато прикладів даних програмних продуктів чи сервісів, які були розроблені знайомими для багатьох компаніями.

Однією з таких компаній є Linux, які створили проект EdgeX Foundry (рис. 2.1), котрий має за мету надавати доступ до універсальної модульної платформу, щоб забезпечити взаємодію між ІоТ-пристроями, додатками і сервісами, а також створювати екосистеми з компаній-виробників, що випускають сумісні і взаємозамінні компоненти для інтернету речей. Також, представлена платформа дозволяє створювати шлюзи, які б могли поєднувати наявні ІоТ- пристрої і збирати дані від різних датчиків; та попри організації взаємодії з пристроями, в даній платформі шлюз виконує завдання по первинній обробці, агрегування та аналізу інформації, виступаючи проміжною ланкою між мережею з ІоТ-пристроїв

і локальним керуючим центром або хмарної інфраструктурою управління, що дає змогу зберігати інформацію в великих обсягах та на довгий час.

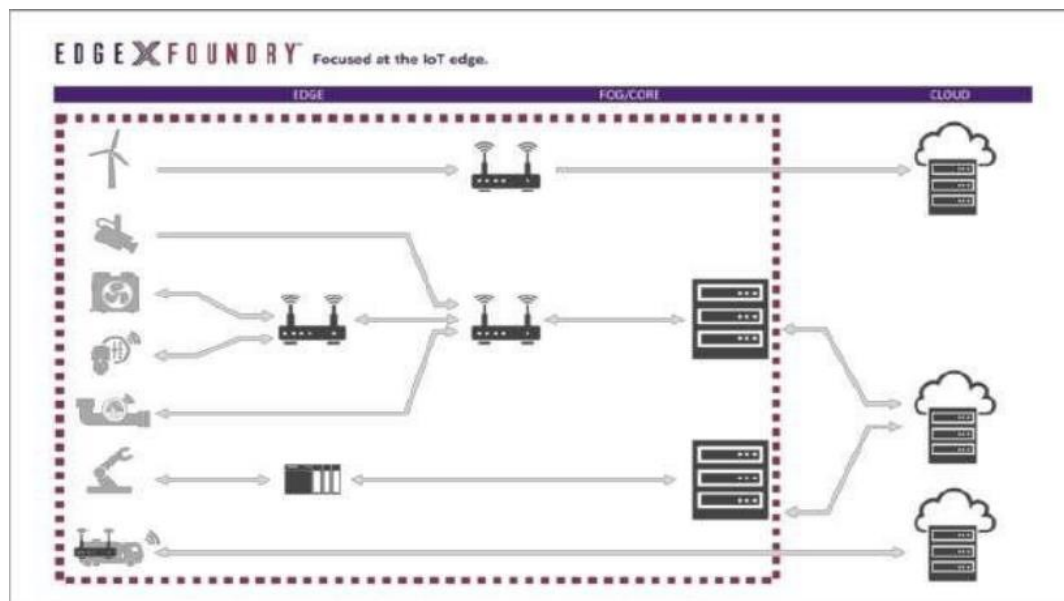


Рис. 2.1. - Структура платформи компанії Linux

Ще одним прикладом вдалої інтеграції інтернет речей до фізичних пристроїв є платформа Everyware Cloud від компанії Eurotech(рис. 2.2). Ця IoT-платформа, спрощує управління пристроями і збір даних шляхом підключення розподілених пристроїв через безпечні і надійні хмарні сервіси. Хоч платформа Everyware також використовує хмарні технології, як і минулий приклад, вона є більш зосереджена на цьому питанні. Але головною з переваг вважається взаємодія через M2M, тобто від машини до машини, що говорить про передачу даних без втрати сигналу між двома автоматичними пристроями.

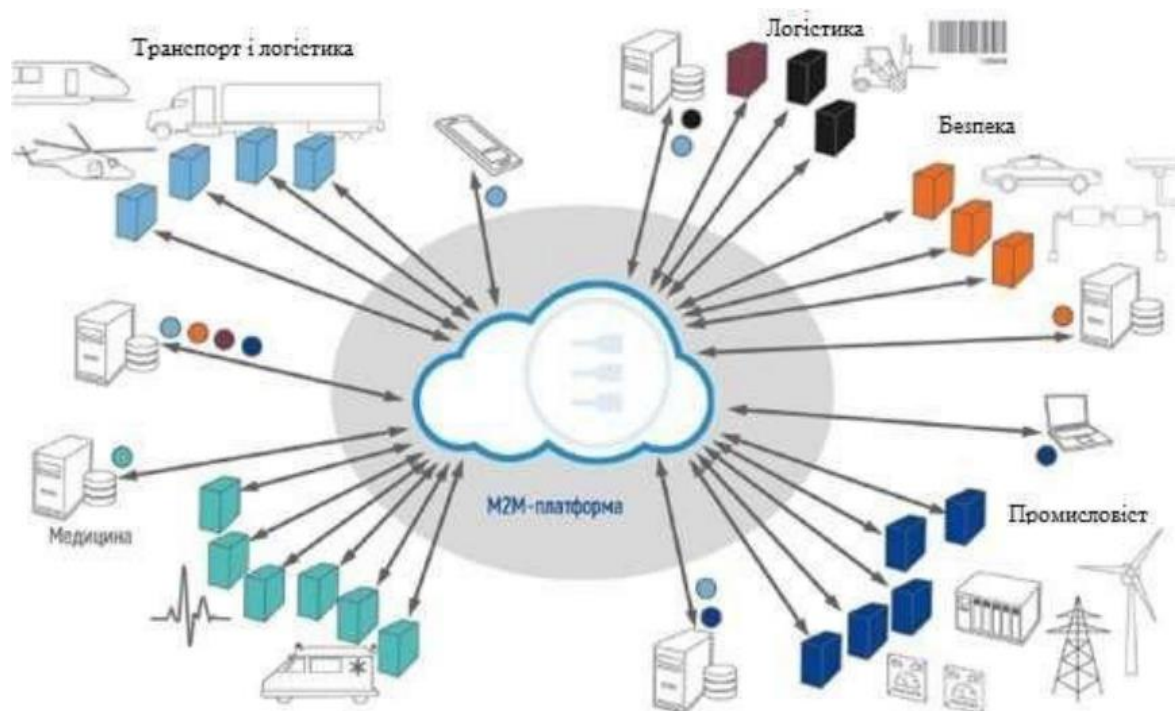


Рис. 2.2. - Хмарна платформа *Everyware Cloud*

Звісно, що це не всі приклади платформ, які використовують хмарні технології для поєднання фізичного та мережевого “світу”. Можна зазначити ще такі сервіси, як:

- IRI Voracity яка використовує механізми IRI CoSort і Hadoop для контролювання даних IoT.
- IBM Watson IoT, завдяки якій можна збирати інформацію про автомобілі, будівлі, обладнання, активи та речі.
- Amazon AWS IoT Core, що дозволяє підключати пристрої до хмарних сервісів AWS без необхідності керувати серверами.

І це ще не всі приклади платформ які можна застосовувати IoT проекті, але також є бажання відзначити Oracle з платформою Arex (рис. 2.3), яка має багато інструментів для відтворення повноцінного сервісу будь-якої розумної системи.

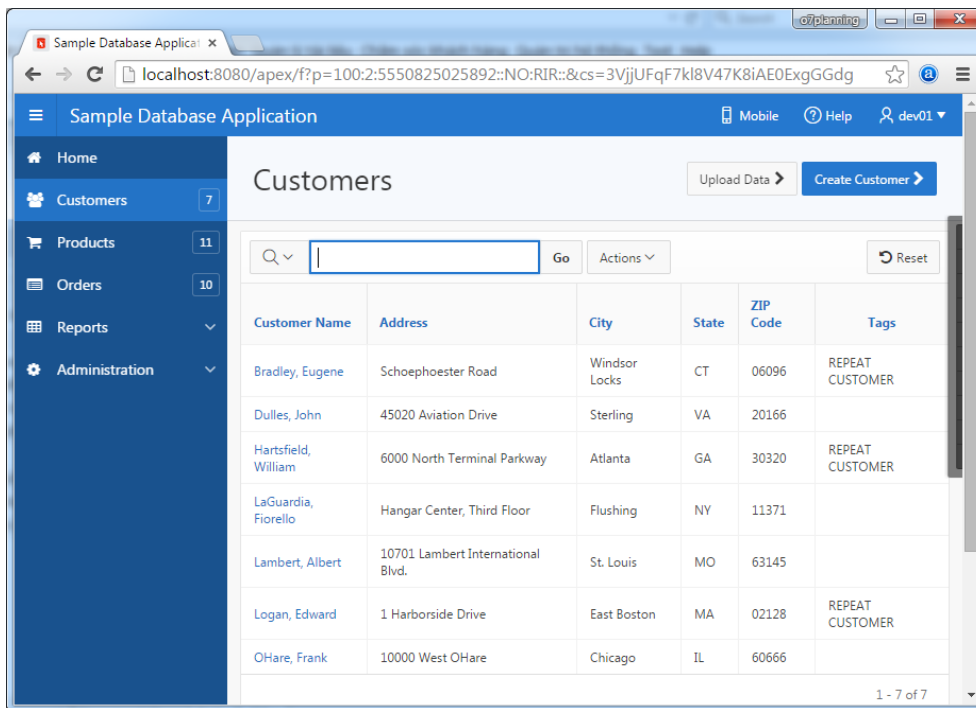


Рис. 2.3. - Приклад проекту в Apex від Oracle

Апекс надає змогу налаштовувати проект, як автоматично так і програмно, але при умові, що ви знаєте програмну мову платформи та її синтаксис, бо вона є доволі специфічною та без змоги знаходження помилок, що може привести до труднощів у реалізації. Попри це платформа є корисним інструментом для роботи з базами даних, легке перенесення даних до сервісу та подальша модернізація економить час, який витрачався б на відновлення даних самостійно. Загалом, це досить зручна IoT платформа для створення власних проектів з використання вже готової власної бази даних.

Проаналізувавши різні сервіси, які надають послуги для створення власних проектів є ще один, не відомий, але дуже зручний та легкий сервіс для створення макетів сайтів чи платформ з базою даних, тому серед розглянутих аналогів була обрана зовсім інша платформа – QuintDB (рис. 2.4). Вона має влаштовану внутрішню базу даних та слугує для проектування прикладу IoT-системи [3], яка може бути відтворена у майбутньому по заданому, свого роду, макету сервісу, який відповідає системі та слугує прикладом того, що треба відтворити для з'єднання існуючих та вже підключених розумних пристроїв.

ID	Client	Balance	Contract start date	Contract expiration date	Contact person
1	Texas Motango LTD.	67888.0	01-10-2019	02-28-2019	A. Jackson
2	Mallister and Brothers co.	35699.0	01-31-2019	03-13-2019	J. Mallister
3	Crossfield	25699.000	02-01-2019	03-29-2019	J. Luwinsky
4	RedTail Co.	58900.000	01-10-2019	02-28-2019	K.J. Smith
5	L-Studio	47990.000	02-15-2018	06-03-2019	S. Alexander
6	Maschinima LTD.	45389.000	02-01-2019	02-01-2020	T. Takashi
7	Sunset LTD	89539.000	06-12-2018	06-23-2019	J. Robinson
8	J&M Production	49299.000	09-17-2018	06-20-2019	A. Milan
Table average:		52590.380			

Рис. 2.4. - Приклад проекту на платформі QuintDB

Сервіс має внутрішню базу даних, котру можна використати для заповнення власних форм чи завантажити таблиці окремо. Усі налаштування відбуваються виключно на сайті сервісу, тому для цього потрібно тільки підключення до мережі інтернет. Звичайно є можливість програмувати окремо, для чого, також, наводиться документація. Надається змога додавати зв'язки між таблицями, робити власний інтерфейс, створювати діаграми, акаунти користувачів, влаштовувати карти місцевості та інші корисні в розробці інструменти, які роблять можливим зробити якісне представлення майбутнього проекту, як, наприклад, системи оптимізації паркову імені Пушкіна.

2.3 Мета дослідження відносно розглянутих пристроїв

Головною метою дослідження, після розгляду платформ, є створення схеми проекту оптимізованого парку та розгляду приладів, які будуть розташовані по площі. Загалом, у 21-ому сторіччі тема зелених ділянок та їх вдосконалення є дуже популярною.

Тому, щодо технічних аналогів чи прикладів пристроїв, котрі можна використовувати, то це питання не є складним, бо технології за доглядом садів, парків чи схожих за цією тематикою об'єктів оновлюються з кожним роком. Розумні екологічні системи використовуються, навіть компаніями для облагороджування власних робочих об'єктів, де наявна велика кількість працівників. Але перед початком розгляду аналогів, потрібно зрозуміти, що саме

повинно бути впроваджено до системи та за що вони будуть відповідати. Тому, треба сформуванати перелік усіх елементів з відповідними для них задачами:

- Автоматичний полив рослин, що складається з контролеру, який буде розподіляти задачі між зонами поливу та зрошувачі, які будуть цей полив здійснювати.
- Елементи комфорту перебування у парку - розумні сміттєві баки та розумні лавочки. Люди повинні мати можливість відпочити, знаходячись чистому середовищі.
- Догляд за парком та здійснення можливим його відвідування у темряві - ліхтарі з влаштованими датчиками руху.

2.4 Автоматичний полив зелених територій

Першим та головним таким пристроєм для кожного парку є автоматичний полив рослин. Точніше це два пристрої, які складають собою автоматичну систему:

- 1) контролер поливу рослин
- 2) зрошувач рослин



Рис. 2.5. - Вуличний контролер Hi-Garden Aquarius 10 Wi-Fi з Wi-Fi і Bluetooth

Hi-Garden Aquarius (рис. 2.5), котрий керує системою поливу через інтернет та технологію Bluetooth. Hi-Garden Aquarius є стаціонарним контролером, що можна доповнити датчиками погоди, які встановлюються на ділянці і передають на пульт дані про вологість ґрунту, температурі повітря і опадах, а, іноді, навіть про заморозки. Змога переглядати та аналізувати дані навіть зі смартфона, надають працівникам парку більше мобільності та самому пристрою також. Його можна спокійно встановити на вулиці та не хвилюватися за цілісність чи спробу втрутитись до даних чи внутрішньої начинки контролера. При будь-якій спробі втручання контролер автоматично відправляє повідомлення про загрозу злому оболонки, та вимкнеться аби запобігти фальсифікуванню даних чи спрацювання під'єднаних об'єктів. Дані про справність зрошувачів, можливі датчики для аналізу стану повітря та землі, тиск води та її температура, корегування праці та встановлення таймеру - усе це може Hi-Garden Aquarius [4], враховуючи, що зону покриття можна збільшувати за потребою, як у випадку цього проекту.

Звісно ж, цей контролер не єдиний в своєму роді. Є простіші варіанти, які можна розглянути для порівняння та розуміння, чому Hi-Garden Aquarius кращий саме для парку. Наприклад, WI-FI контролер HC-601iE(рис. 2.6), котрий керує системою поливу через інтернет. Цей контролер може керувати автополивом тільки у шести зонах та з обмеженням в довготривалій праці до 24-ох годин. Головна особливість даного контролера, це збір інформації не з наявних датчиків, які треба додатково встановити, а з поблизу розташованих метеостанцій, а головним недоліком системи є обмежена кількість покриття території та підключення до контролера тільки через технологію WI-FI. Це обмежує можливість встановити менше контролерів на такі великі ділянки як у парку ім. Пушкіна, тому це призводить до зайвих витрат; та при відсутності доступу до інтернету, але знаходячись коло пристрою, немає змоги встановити з'єднання та провести діагностику, наприклад.



Рис. 2.6. - WI-FI контролер HC-601iE для керування системою поливу через інтернет

Також, треба зазначити, що важливою частиною системи поливу є самі зрошувачі, які це роблять. PGP Ultra [5] найбільш вдалий варіант для того, щоб саме його поєднати з вже обраним контролером. Це сучасний зрошувач рослин, який, навіть, має оновлену конструкцію, яка збільшила радіус поливу від 4,6 м до 10,7 м. Для цього використовується влаштовані роторні дощовики, в яких головка з лопатями обертається і розбризкує воду під заданим кутом. У нерегульованих роторів струмінь води описує повне коло, у регульованих, як PGP Ultra, сектор поливу задається в діапазоні від 40° до 360° , що можливо регулювати дистанційно через контролер. Ще одною з особливостей є покращена стійкість до брудної води на внутрішньому редукторному приводі, що в синергії з контролером може повідомити про це, якщо у цьому є необхідність.

Особливості нової версії PGP Ultra (рис. 2.7) представлено далі. Таку систему поливу вже можна вважати укомплектованою, але, можливо, що буде не зайвим переглянути додаткові варіанти зрошувачів, які можуть бути альтернативою при труднощах з основним підібраним пристроєм чи для запасного варіанту для менше складних проектів.

Взагалі, окрім зрошувача, який розбризкує воду по своєму радіусу, маюючи кола та дуги є інші види подібних пристроїв. Наприклад, дощувачі, котрі виконують полив завдяки симуляції дощу, тобто вода з отворів виходить, як з фонтану. Тому можна перейти до наступного аналогу, яким є імпульсний

дощувач GARTNER (рис.2.8.), котрий закріплено на штанзі, що додає висоти до 75 см.



Рис. 2.7. - Система поливу рослин PGP Ultra



Рис. 2.8. - Дощувач імпульсний GARTNER 80081067

Завдяки цьому площа можливого поливу збільшується до 315 м², та здійснюється по колу. Режими поливу можна змінювати, змінюючи радіус кола та його цілісність, але такі налаштування робляться тільки вручну. Хоч даний пристрій не є автоматизованим, його усе ж таки можна під'єднати до ланцюгу поливу з кількох розпилювачів, що дає змогу комбінувати цей пристрій з першим

розглянутим PGP Ultra. Звісно, необхідності у цьому немає, бо дані приклади зрошувачів не є настільки інноваційними, що позбавляє систему поливу синергії, але усе ж таки дана модель, може підійти як запасний варіант, який може бути потрібний при втраті контролю над системою у разі вторгненні до контролера чи його поламці.

2.5 Розумні паркові лавочки

Обравши пристрої для однієї з головних частин парку, треба перейти до приладів, при впровадженні яких територія буде рахуватися розумно облагородженою.

Першими на черзі ідуть смарт лавочки, які вже використовуються в різних парках та суспільних місцях не тільки в закордоном, а в Україні також. Подібні лавочки можна зустріти й у Києві, наприклад у студентському містечку КПІ та центральних алея міста. Загалом, вони мають тільки дві особливості, це можливість заряджати мобільні пристрої та отримувати енергія для заряду через сонячні батареї. Це рахується вже досить сучасним, але технології оновлюються постійно, тому вже більш інноваційні моделі та їх аналоги.

Модель на огляд буде з тієї ж серії, як і наступна SMART EKO CITY, яку було обрано для проекту, бо лавочки SMART EKO CITY вже є на території України та логічним і оптимальним рішенням було б обирати саме серед них. Тому аналогом існує SMART EKO CITY Model SC22 (рис.2.9), котра також при собі має сонячну батарею для накопичення енергії, щоб заряджати власні пристрої. Також, особливістю даної моделі передбачена можливість комплектуватися зі стойками для паркування велосипедів, максимум 4 штуки, а керування функціями лавочки в змозі контролюватися за допомогою мобільного телефону.



Рис. 2.9. - Лавочка SMART ЕКО CITY Model SC22

Для парку дана модель доволі вдало вписується через розташування місць по обидві сторони та створення уявної тіні, завдяки сонячній батареї. Вона має такі ж переваги як перша модель, хоч і зі своїми особливостями. Лавочка SMART ЕКО CITY Model SC22 [6] могла би бути тим самим рішенням для парку, яка б могла його заповнити та надати відвідувачам купу місць для відпочинку, якщо б не існувала модель ще краща та сучасніша, така як лавочка SMART ЕКО CITY Model SC18 (рис.2.10).



Рис. 2.10. - Лавочка SMART EKO CITY Model SC18

Саме ця розумна модель є ідеальним варіантом для проекту та оптимізованої системи загалом. В неї також, звісно, впровадженні технології з сонячними панелями, зарядкою та підсвіткою, але що саме головне - ця лавочка модифікована LCD екраном для показу мультимедійного контенту та Wi-Fi модулем. Модуль Wi-Fi, спроможний отримувати інтернет від 4G LTE модема, а площа покриття становить від 10 до 20 м від лавочки, що дає змогу людям, які будуть знаходитися поблизу чи відпочивати завжди залишатися на зв'язку. Ще враховуючи, що елементи нашої системи повинні синергувати між собою, технологія Wi-Fi може значно цьому посприяти.

З даними перевагами це вже є лідер зі схожих пристроїв, але детальніше про варіанти застосування LCD екрана треба зазначити:

- Демонстрація карти самого парку.
- Вивід даних про стан погоди у місті в реальному часі.
- Карта міста.

- Розклад автобусів, які прямують до парку та поблизу нього.
- Реклама.

2.6 Технології розумних сміттєвих баків

Визначившись з вже з трьома елементами нашої системи, правильним буде переходити до наступного - це сміттєві баки з датчиком на завантаженість, також, сонячною батареєю та прес-контейнером.

Одним за таких є Clever Bin, який є прес-контейнером для сміття нового покоління: вбудований всередину бака прес дозволяє помістити в ємність кількість сміття у вісім разів більше, ніж аналогічні за розміром сміттєві контейнери, які не обладнані цією системою. Тому спустошувати бак необхідно набагато рідше - потреба у звільненні бака від сміття знижується до 80%. Також, крім вбудованого преса, в контейнері встановлений бездротової датчик наповнення бака - CB Sensor. Він контролює наповненість бака. У режимі реального часу датчик сигналізує про заповнюваності контейнера за допомогою програми Clean City Networks.

«Розумний смітник», також, працює з сонячною батареєю, що значно знижує витрати на енергію. Притому з урахуванням установки в різних локаціях баки Clever Bin (рис.2.11) виробляються з трьома варіантами подачі живлення:

- електромережу;
- сонячна енергія;
- обидва види харчування.

Ураховуючи це, можна обрати оптимальний варіант баку для певних зон, чи, взагалі, встановити різні баки в одній зоні.



Рис. 2.11. - Розумний Сміттєвий Бак Clever Bin

Нажаль, Clever Bin [7] використовуються не усюди, тому впровадження може бути складним, але бути першопроходьцями теж має привілеї.

У свою чергу, польські інженери та дизайнери розробили розумний сміттєвий бак Vin-e (рис.2.12), який за їх розрахунками покращить якість життя в розумному місті та збільшить кількість переробленого сміття. У свою чергу вони також привели статистику, де надавалася інформація, що на планеті переробляється лише 10% відходів, а решта 90% забруднює довкілля. І через це розробники вирішили розробити інноваційний продукт в даній сфері, що спрощує сортування сміття. Розумний сміттєвий бак Vin-e сам ідентифікує, класифікує та сортує відходи, а також надсилає запит про чищення обслуговуючої компанії, що значно пришвидшує роботу, економить ресурси та час.

Vin-e вміє попередньо обробляти сміття. Наприклад, змінює пластик для економії простору. У Vin-e закладено алгоритм машинного навчання. Він щодня навчається розпізнавати нові об'єкти, підключаючись до хмарної бази даних, що є його головною особливістю та метою - навчатися для більшої працездатності.



Рис. 2.12. - Польський розумний сміттєвий бак Vin-e

2.7 Освітлення парку

Пройшовши майже усі елементи даного проекту, фінальним буде освітлення, котре використовується усюди та першочерговою потребою у місцях з великим скупченням людей.

Для освітлення території на великих просторах, є логічним використання пристроїв, які можуть покривати більшу площу, ніж звичайні ліхтарі. Звичайно, треба пам'ятати нашу мету зі створення системи оптимізації паркової зони, тому гарним рішенням буде використовувати ліхтарі, котрі мають реагування на рух, бо наш парк має години праці з 8-ої ранку до 8-ої вечора, тому в інший час, коли нікого не має бути на території парку, ми можна використовувати влаштовані датчики руху.

Переваги влаштованих у ліхтарі датчиків руху:

- економії електроенергії;
- швидке виявлення несанкціонованого вторгнення;
- два прилади в одному;

Зрозумівши, який пристрій нам потрібен ми повинні обрати найкращий варіант, як було зроблено із іншими розумними технологіями [7]. Тому після аналізу ринку пристроїв відповідних даному запиту, було знайдено три пристрої, які підходять до даної системи:

1) Регульований ліхтар світильник - СВК FL-1752(рис. 2.13)

2) Вуличний ліхтар у вигляді камери - Solar Light BL JLP 2178 (рис. 2.15)

Другий варіант - це вуличний ліхтар у вигляді камери - Solar Light BL JLP 2178 (рис.2.13 – 2.14). Особливість цього ліхтаря, окрім того, що в нього вже є вбудований датчик руху та сонячна батарея, це те, що він має пульт керування та виглядає, як камера. Тобто, цей пристрій слугує муляжом камери з можливістю реагувати на рух включенням світла. Також, він має сонячну батарею, яка дає змогу використовувати альтернативну енергію та економити електроенергію загалом. Головне ж, це те, що ліхтар містить у собі датчик руху, тому він відповідає третьому пункту з переваг. Попри це, є налаштування сутінкового датчика. Тип лампи у цьому ліхтарі - LED з вбудованими світлодіодами. Загальна кількість даних ламп становить 77, по 7 у кожному з 11 вікон ліхтаря. Матеріал основи приладу - пластик. Також можна зазначити, що кольори ліхтаря можуть бути тільки чорні або білі, але це більше до дизайну проекту.

Щодо захисту самого пристрою, існують класи герметизацій - IP(ступінь захисту оболонки). IP-рейтинг (Ingress Protection Rating, вхідний захист) - система класифікації ступенів захисту оболонки електрообладнання від проникнення твердих предметів та води відповідно до міжнародного стандарту ІЕС 60529. В даному випадку, цей ліхтар має захист IP-65, де 6 позначає повний захист від проникнення пилу, а 5 - захист від струменів води з усіх напрямків. Треба зазначити, що 6 це максимальне визначення захисту, тому дана система класифікує цей прилад, як один з найкращих. Захист від води також достатньо високого рівня, тому дощ не зможе завадити роботі ліхтаря. Але треба зауважити, що кріплення ліхтаря не є надійними та при зливі чи бурі, що трапляється під кінець літнього сезону та восени в Києві, їх може зірвати, що призведе тільки к більшим збиткам.

З цього виходить, що даний варіант розумного ліхтаря не влаштовує нашу систему та не підходить по усім критеріям наведеним вище.

Вигляд ліхтаря Solar Light:



Рис. 2.13. - зони реагування на рух ліхтаря Solar Light BL JLP 2178



Рис. 2.14. - Вуличний ліхтар у вигляді камери Solar Light BL JLP 2178 з датчиком руху та пультом керування на сонячній батареї Прожектор - муляж камери

Тому, для даного проекту був обраний саме регульований ліхтар світильник Split Solar Wall Lamp GS-4053. Цей зразок, також, як і другий варіант має влаштовану сонячну батарею та датчик руху. Одразу треба пояснити чому саме цей пристрій є тим самим, що буде відповідати нашій системі та вдало підходить для паркових зон.

Вигляд вуличного ліхтаря в розгорнутому стані:



Рис. 2.15. - Регульований ліхтар-світильник Split Solar Wall Lamp GS-4053

На початку підбору варіантів було вказано, що потрібен ліхтар, який буде покривати як можна більші ділянки парку, бо для 19-ти га території потрібно багато світла, навіть, якщо в ночі вона не працює. Даний прилад має 138 світлодіодних ламп які розташовані на трьох панелях, під різним кутом, який можна змінювати, що дає змогу збільшувати зону покриття світла. Центральна панель, яка розташована між двома іншими, має датчик руху, який реагує на будь яке пересування, так само як і другий варіант. Також, слід нагадати про сонячну панель, котра більше ніж у вже зазначеного пристрою раніше Solar Light BL JLP [8], та також може змінювати кут нахилу, залежно від розташування, яке буде влаштовувати, звісно з розрахунком на найбільше споживання сонячної енергії.

Проаналізувавши приклади, є очевидним, що дана модель вуличного ліхтаря є найбільш відповідною для нашої системи. Усі особливості та переваги, які були потрібні знайдені в даному приладі, який задовільнить й відвідувачів оновленого парку.

2.8 Визначення вигляду території парку

Розглянувши аналоги всіх елементів потрібних для системи оптимізації парку імені Пушкіна потрібно переходити до проектування та розташування обраних пристроїв на визначеній території.

На початку, треба ще раз переконатися, що дані про парк дійсні на теперішній час. Наприклад, нещодавно була перебудована історична пам'ятка, на місці якої зараз знаходиться ресторан. Розташування якого у самому початку парку. Також, поруч з територією парку відбувається налагодження будівництва новобудов, які йдуть вздовж периметру зелених насаджень. Тому мною була знайдена оптимальна карта, на якій я базувався при створенні схеми розташування об'єктів. Разом з картою від Google (рис. 2.16) дана схематична зображена територія показує не тільки розташування будівель на території парку, а й канатне містечко, яке працювало тільки влітку, наразі воно не є активним.

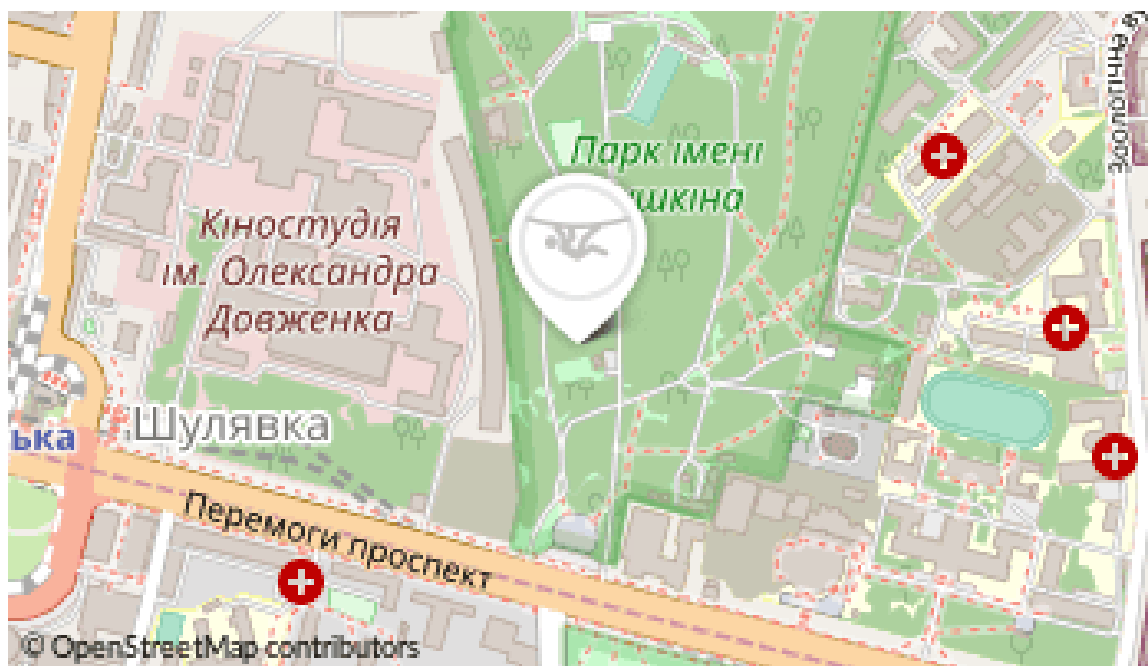


Рис. 2.16. - Карта парку імені Пушкіна

2.9 Розгляд програми для побудови схеми

Після огляду місцевості та отримання розуміння щодо знаходження алей, будівель та санвузлів можна починати відтворення схеми парку власноруч. Для цього була використана веб версія сервісу Draw.io [9].

Доволі легка у використанні програма дозволяє обрати будь які позначки для власних проектів чи створити нові. В даному проекті були використані вже існуючі позначки та фігури, тільки під назначеними згідно елементів назвами (рис. 2.17).

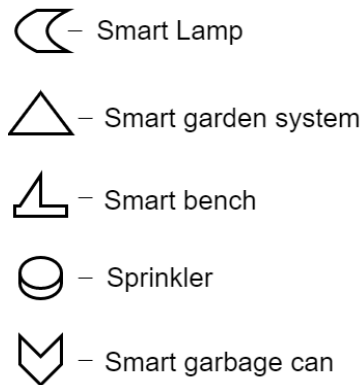


Рис. 2.17. - Позначення елементів на схемі проекту

Представлена на наступному зображенні схема (рис. 2.18) є частиною загальної схеми парку та слугує демонструванням зон з найпоширенішою відвідуваністю людей. Саме у цьому прямокутнику землі є важливим поруч з експоцентром є необхідним використати системи автоматичного поливу, кількість яких буде становити 7 у кожній зоні, де є позначка зрошувача; та пов'язані вони разом з двома автоматичними контролерами. Вони потрібні для поливу газону ввечері, коли там вже не має бути людей та зранку до початку їх відвідування.

Також, необхідним є освітлення, яке розташовано вздовж доріжок кожні 25 метрів, що становить для цієї ділянки 22 ліхтарі. Разом з цим на алейках встановлені лавочки та сміттєві баки, загальна кількість яких 15 та 10. Сміттєві баки не розташовані разом з лавочками через створення меншого забруднення території відпочинку самими відвідувачами. Таким чином, усі елементи системи можуть створювати комфортне місце перебування в цій зоні.

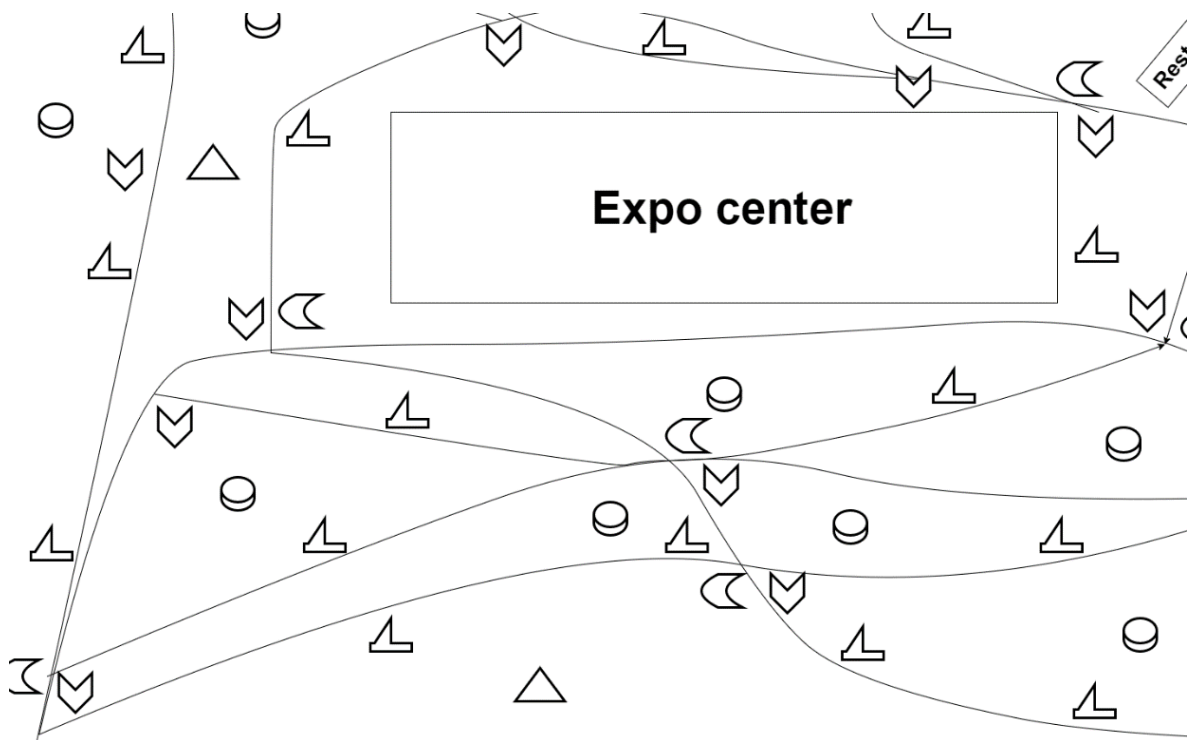


Рис. 2.18. - Схема з місцем найбільшою відвідуваністю людей

2.10 Пояснення побудови схеми парку

Накресливши алеї та схематично зобразивши споруди (Додаток А), було розпочати розташування пристроїв (рис. 2.19) згідно з планом та обраними варіантами.

Першими були розумні ліхтарі, які розташовані на кожному перетині доріжок. Умовне позначення на перехресті позначає, що кожні 25 метрів на цих проміжках між ліхтарями розташовані такі самі розумні лампи за датчиками руху. Так зроблено для того, щоб не займати місце на плані.

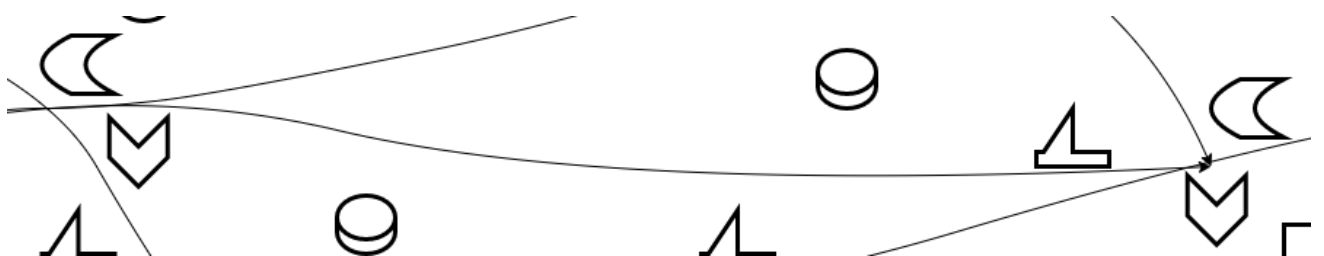


Рис. 2.19. - Фрагмент схеми з розташованими елементами системи.

Проміжок у 25 метрів є середнім значенням для того, щоб не перенасичувати світлом увесь простір, тому дане значення може змінюватися.

Другими в розташуванні йдуть зрошувачі та контролери поливальних систем. Обрані зрошувачі розташовані по ділянкам де є газони та рослини, котрі потребують постійного догляду. Усього таких зон є 26, де кожні 20 метрів розташовані пристрої для поливу зелених зон. Контролери у свою чергу можуть робити охоплення на декілька зон одразу, тому вони розташовані не по всіх зонах, а тільки на межах їх перетину чи поблизу скупчення зрошувачів, тому їх кількість 7, по 4 зони на 6 контролерів та 2 на 7-омий, що дає у подальшому, за потребою, можливість розширення.

Третіми та останніми елементами цього проекту є розумні сміттєві баки та інноваційні лавочки. Їх розташування розглядалося лише відповідно до алей та доріжок для відвідувачів, тому їх кількість становить 33 сміттєвих баки та 57 лавочок, де з кожною третьою також будуть розташовані сміттєві баки вздовж усіх пішохідних зон. Розташування баків поруч з кожною лавкою є не потрібною витратою, тим паче, що люди, котрі будуть відпочивати не будуть мати можливості смітити поруч з такими інноваційними лавочками. Хоча й на початкових етапах впровадження, рекомендується слідкувати та перешкоджати порушникам, щоб у подальшому перебуванні на території парку відвідувачі звикли з цим правилом.

2.11 Функціональна модель та опис класів для майбутньої бази даних

На даному етапі використовувались нотації BPMN для опис роботи проекту та детального розгляду класів для майбутніх таблиць.

Спочатку створюється функціональна модель діяльності проекту (рис. 2.20). Ця модель описує роботу проекту при процесі запиту схеми проекту для подальшого затвердження:

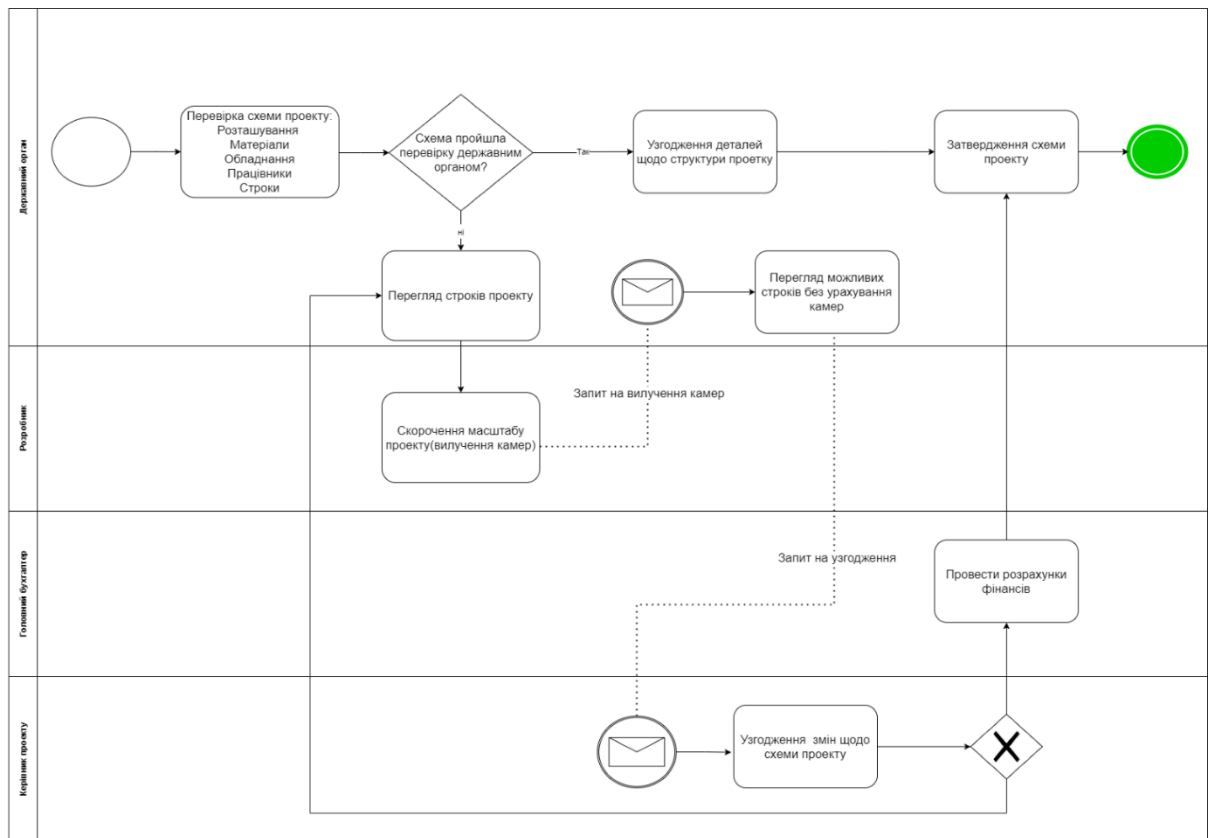


Рис. 2.20. - Процес "Запит на схему проекту для парку"

Після "запиту" створюється ділова модель, яка є описом класів та функцій яким вони відповідають (таблиця 2.1):

Таблиця 2.1 Ділова модель проекту

Функції \ Класи	Обладнання	Працівник Парку	Держ Службовець	Розробник
1	2	3	4	5
Догляд за парком	*			
Внесення оновлень				*
Перевірка роботи системи				*
Контроль системи парку			*	
Сповіднення про зміни	*	*		

Дана таблиця зазначає класи, які використовуються для первинного створення таблиць бази даних. Далі їх буде становитися більше з розширенням функцій та датчиків задіяних у проекті.

Концептуальна схема робочого проекту (рис. 2.21) здебільшого зазначена для представлення Таблиці 2.1 у більш розгорнутому вигляді:



Рис. 2.21. - Концептуальна схема "Робочий проект"

Також, цю схему можна віднести до схеми взаємодії окремих класів бази даних між собою у системи веб-порталу.

2.12 Створення та опис контекстної діаграми

Виконуючи попередню підготовку для створення бази даних у програмному середовищі Microsoft SQL Server, треба зазначити опрацювання проекту також у середовищі AFPM та відтворення функціональної моделі системи в нотації IDEF0 та IDEF3.

AllFusion Process Modeler (AFPM) - CASE-засіб для моделювання бізнес процесів, що дозволяє створювати діаграми в нотації IDEF0, IDEF3, DFD. В процесі моделювання AFPM дозволяє переключитися з нотації IDEF0 будь-якої 36 гілки моделі на нотацію IDEF3 або DFD і створити змішану модель. AFPM підтримує функціонально-вартісний аналіз (ABC).

Першим етапом створення контекстної діаграми (рис. 2.22) є обрання нотації. В нашому випадку це IDEF0.

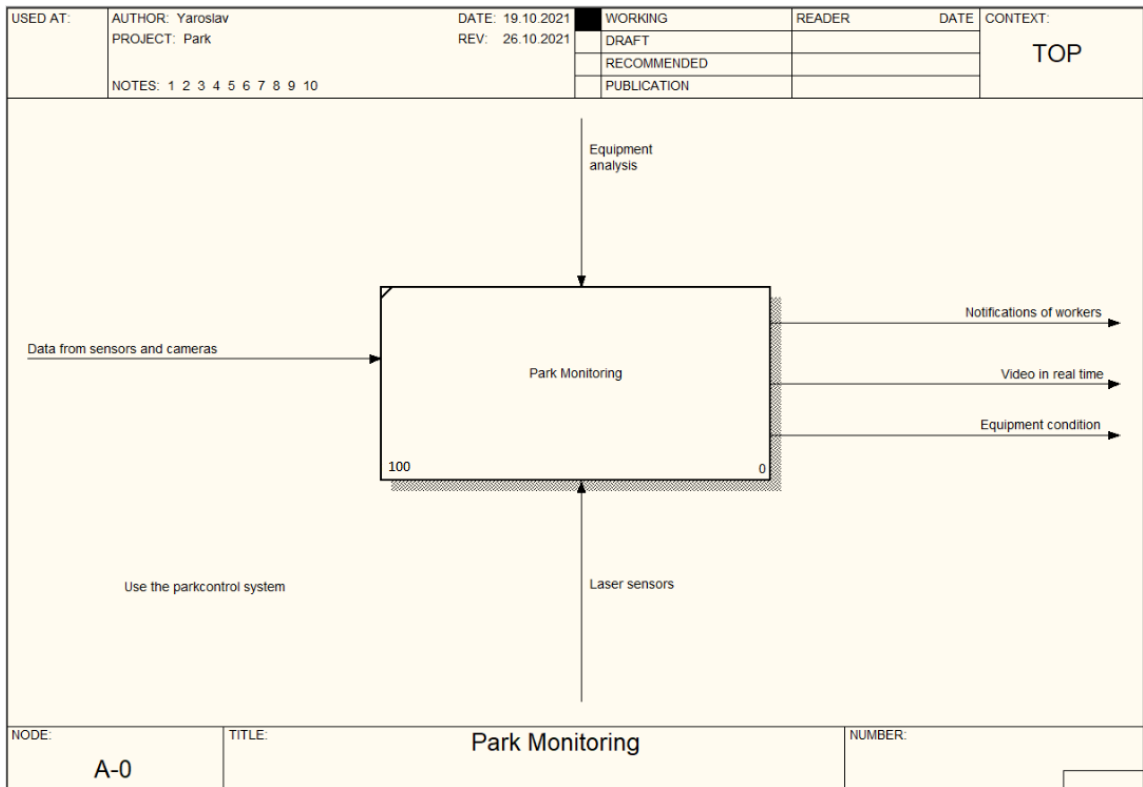


Рис. 2.22. - Контекстна діаграма

Виконавши створення контекстної діаграми можна перейти до діаграми декомпозиції (рис.2.23):

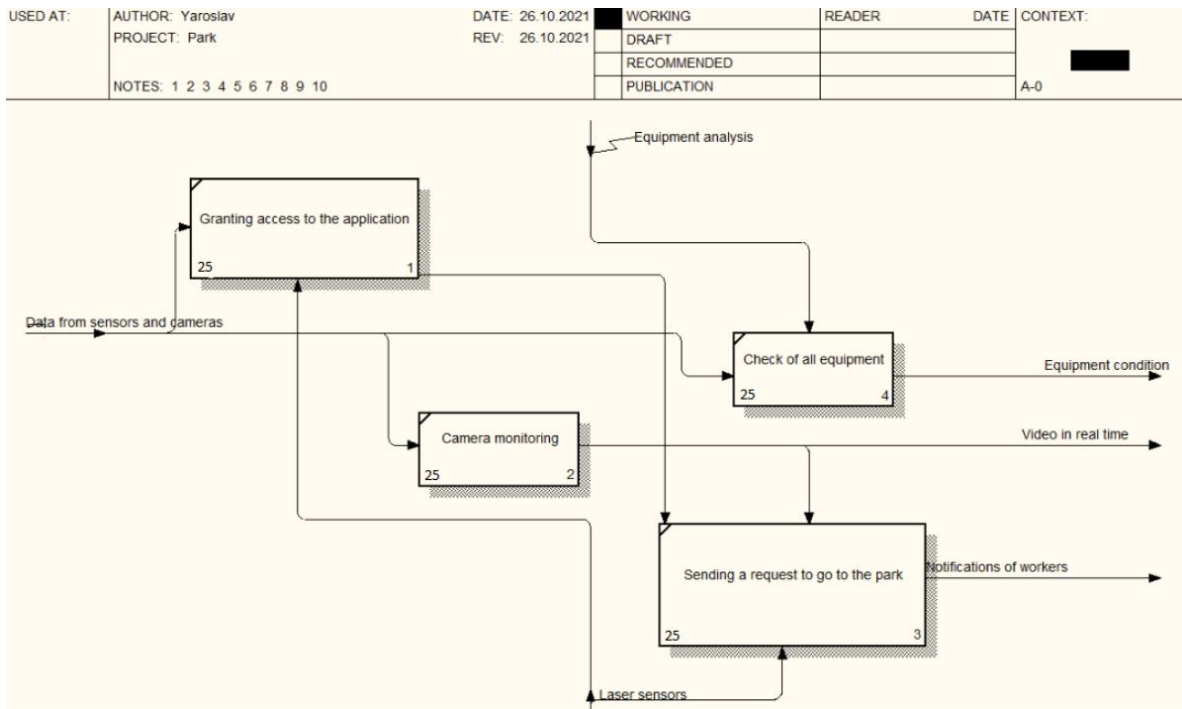


Рис. 2.23. - Готова діаграма декомпозиції з розрахунками

Взагалі, створення цих діаграм сприяє на головне виконання роботи - це відтворення бази даних. Тому у даних діаграмах треба налаштувати зв'язки, які зображують роботу елементів системи. Наприклад, наведені вище декомпозиції є прикладами того, як система працює при поступі даних до пристроїв та вже з пристроїв до самого сервісу. Передача даних вже зрозумілих для людини з датчиків руху та систем поливу надається через програмні додатки вже розроблені для даних приладів, вони можуть просто переадресувати показники до платформи.

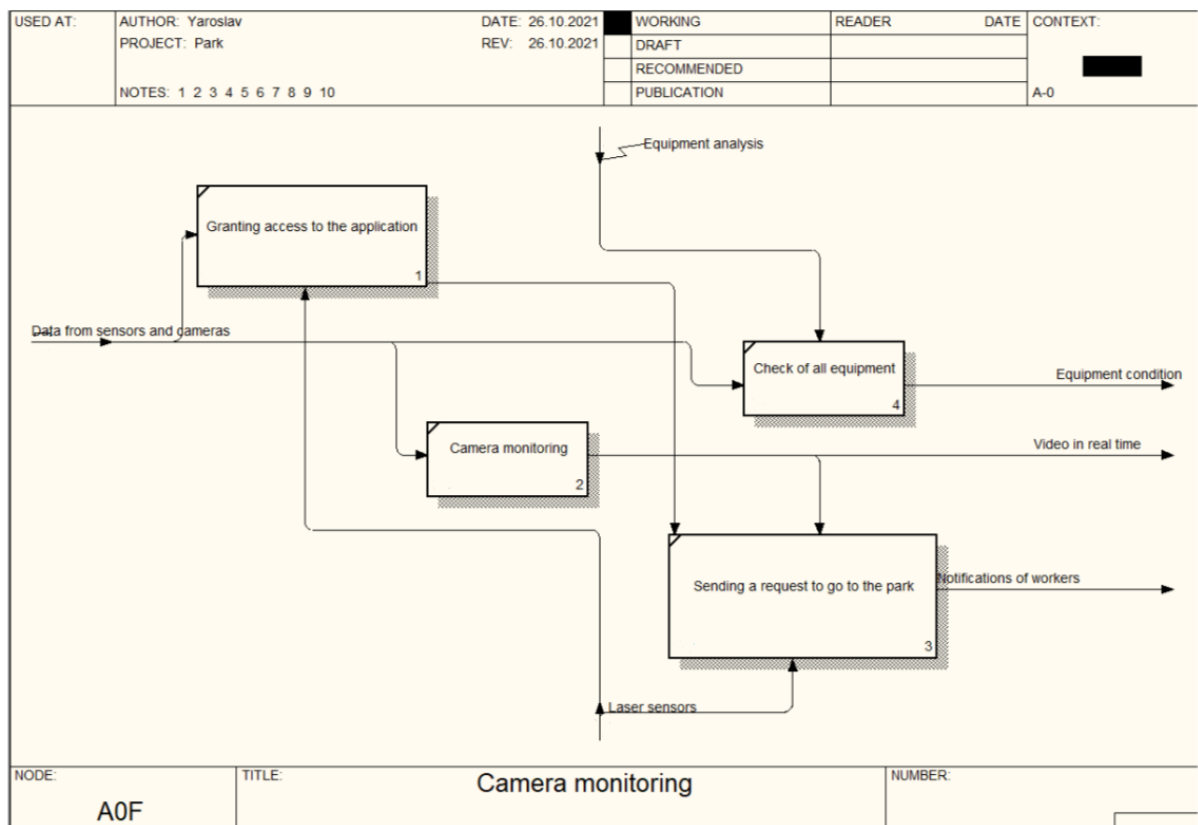


Рис. 2.24. - FEO діаграма

Таким чином, розробивши попередні діаграми ми наблизилися до діаграми в нотації IDEF3 (рис.2.24), яка створюється для окремої функції проекту, в даному випадку “Доступ до програми” для подальшого використання її функцій.

На вході даних діаграм система отримує дані з сенсорів та камер, дані про робітників підв'язаних до окремих парків та дані документацій, тобто інформацію про технології застосовані на зеленій території, їх розташування, кількість та типи.

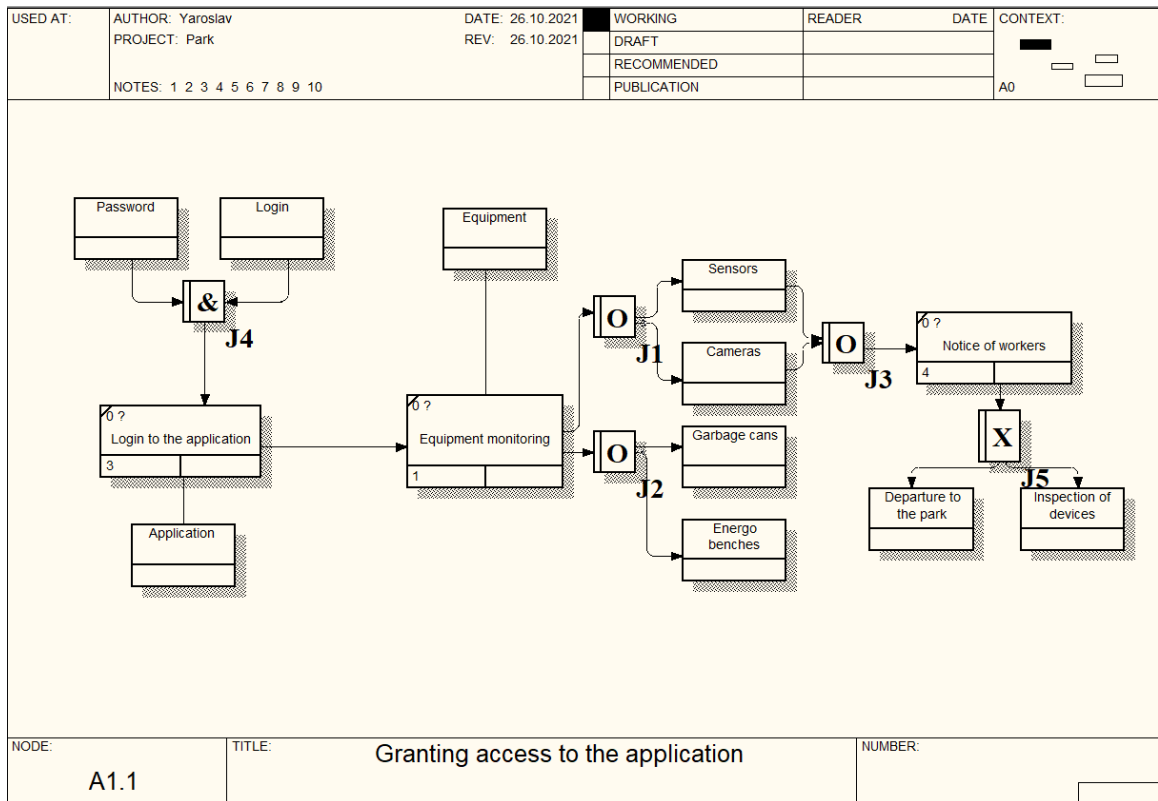


Рис. 2.25. - Діаграма в нотації IDEF3

Окрім, позначення моніторинг та входу до системи, в даній діаграмі (рис. 2.25) були використанні класи, додані в минулі таблиці, які є поділеними на пари для передачі запиту працівникам на виїзд до парку.

Наступним етапом підготовки до моделювання баз даних було створення діаграми в нотації DFD та вивчення процесу моделювання для моніторингу парку за допомогою того ж самого інструментального середовища AllFusion Process Modeler (рис.2.26).

Нотація DFD представляє собою сутності потоку даних з точки зору їх збереження та обробки. Вони, в свою чергу, утворюють один із головних інструментів структурного та проектування інформаційних систем.

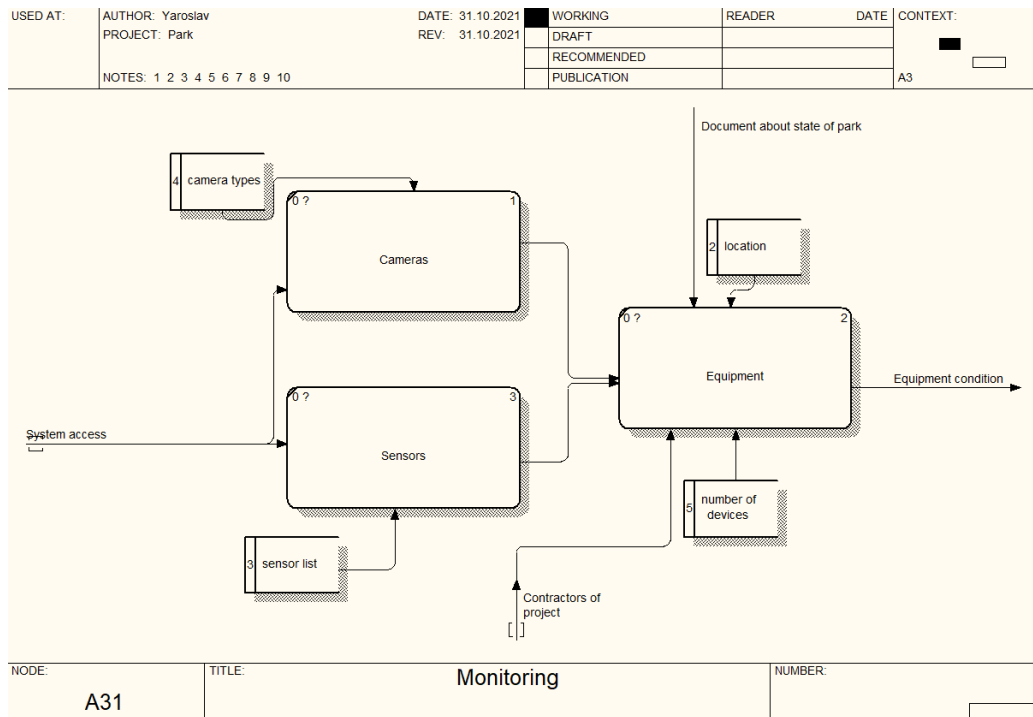


Рис. 2.26. - DFD діаграма з діяльності робітників

2.13 Повне представлення діяльності згідно нотації UML

Визначивши сферу автоматизації догляду паркових зон да моніторинг парків дистанційно, треба завдяки мові об'єктно-орієнтованого моделювання – UML (Unified Modeling Language) створити моделі діяльності проекту.

Перша створена модель була прикладом діаграми прецедентів для інформаційної системи оформлення проекту. Ця діаграма описує взаємодію робітників парку та користувачів веб-порталу з самим додатком та функції, які кожен з них повинен виконувати згідно своєї ролі (рис.3.12).

Діаграма даного типу більш показує взаємодію між користувачами різного класу та додатку, ніж саму роботу програми, але завдяки цій інформації можна позбутися певних етапів тестування, якщо воно буде проводитися.

Другою діаграмою стала діаграма діяльності, яка представляє собою кожний крок користувача при початку використання веб-порталу, тобто, від авторизації до моніторингу та створення запитів до працівників парку (рис.2.27)

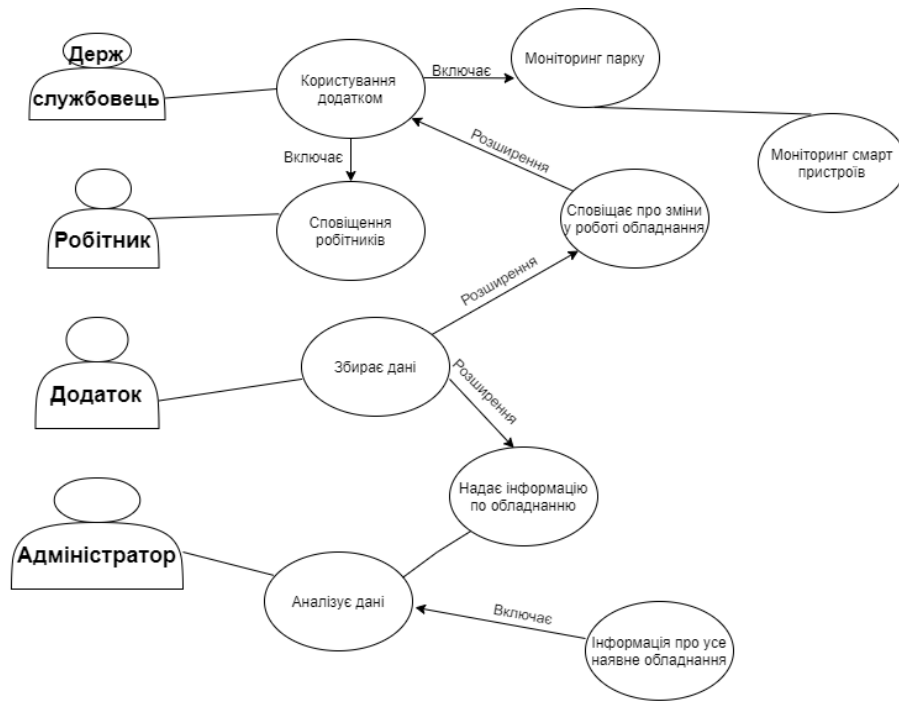


Рис. 2.27. - Приклад діаграми прецедентів для інформаційної системи
Оформлення системи

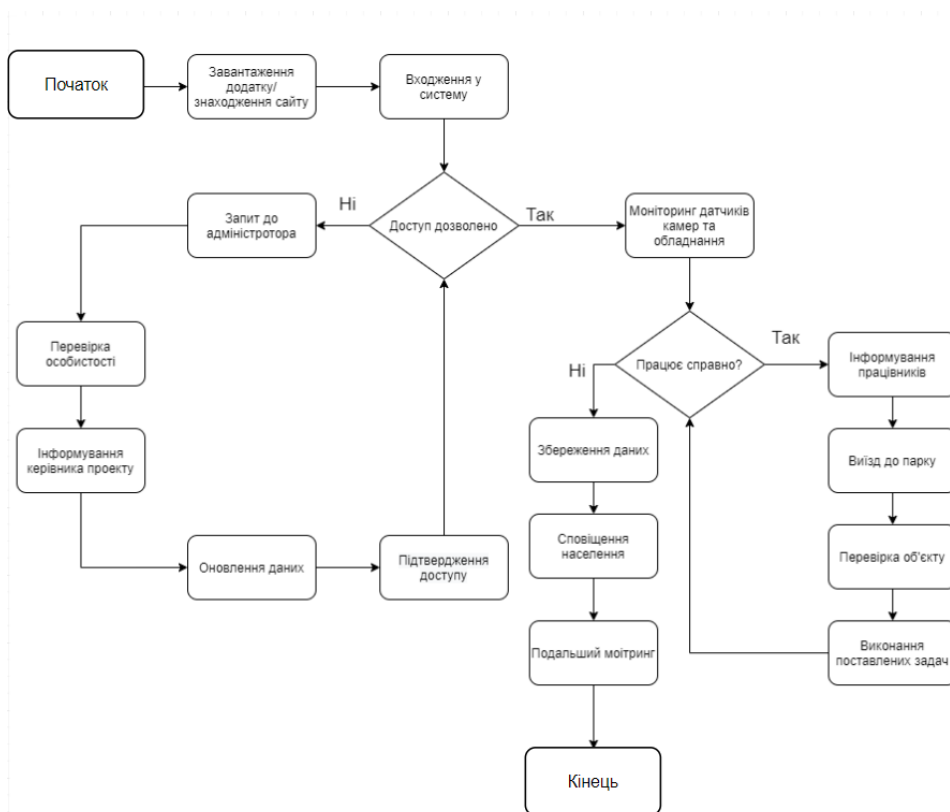


Рис. 2.28. - Приклад діаграми діяльності

Ще, даний приклад діаграми (рис. 2.28) вже може демонструвати повну картину діяльності додатку та проекту в цілому. Поетапний опис виконання

кожної дії зазначені з різними варіантами результатів, але кінцевий завжди однаковий - справна система для моніторингу.

Третьою діаграмою зазначена діаграма послідовності дій від особи адміністратора (рис. 2.29):

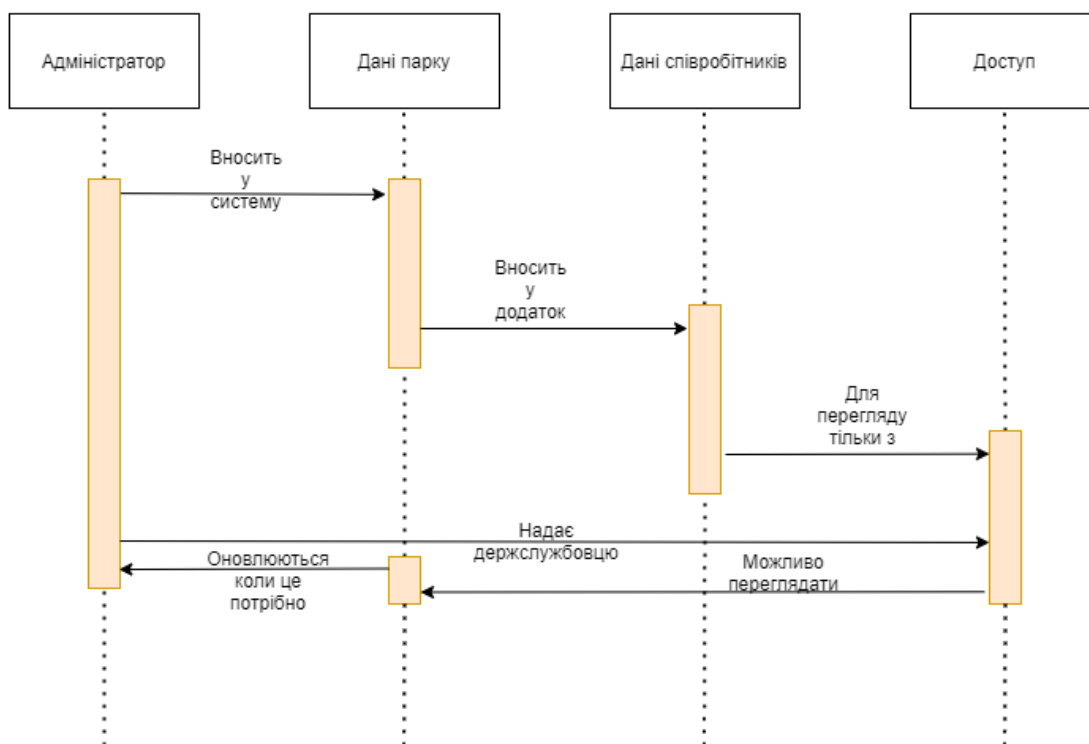


Рис. 2.29. - Приклад діаграми послідовності

Та останньою є діаграма, яка представлена у вигляді логічної моделі баз даних, вона показує зв'язки між ними таблицями та кожне поле, які будуть створюватися при роботі з базами даних в програмному середовищі DBeaver (рис. 2.30):

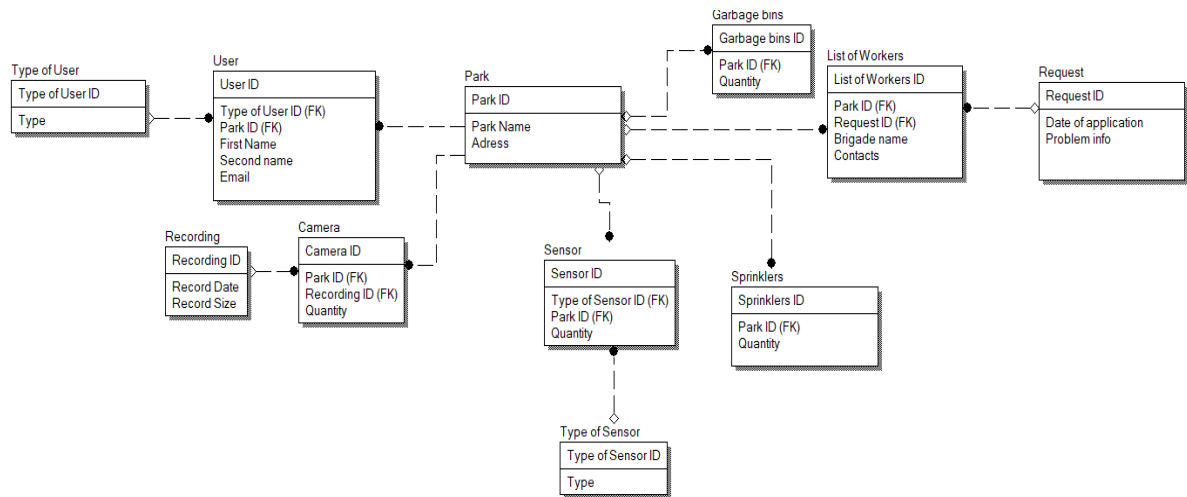


Рис. 2.30. - Приклад діаграми класів

Загалом, вона є прикладом діаграми класів із зазначеними усіма таблицями та класами, які у подальшому можуть використовуватися в додатку для заповнення форм та додання нових даних відповідно до парку та прив'язаних до нього класів девайсів чи користувачів. Також, треба зазначити, що можуть вноситись зміни при додаванні чи видаленні певних елементів системи.

На даному етапі це уся робота, котра повинна була бути виконаною перед відтворенням бази даних.

Висновок до розділу 2

IoT платформи, та принципи побудови працездатного проекту, було досліджено аналоги сервісів, які надають можливість взаємодіяти з цими платформами для відтворення проекту, після чого розглянуто їх переваги та недоліки для вирішення, яку саме платформу буде використано для створення початкового відтворення проекту. Це надає більше можливостей в становленні зв'язку з сервісними програмами пристроїв системи для того, щоб легше отримувати дані для моніторингу цілого парку. Тому, звісно, було розглянуто аналоги пристроїв, які б могли стати елементами системи оптимізації парку, виявлено найкращі варіанти, які відповідають потребам проекту. Також, серед розглянутих елементів були знайдені запасні варіанти, які можливі у використанні для модернізації при подальшому розвитку проекту. Загалом, серед аналогів системи винайдено елементи, які вже є взаємопов'язані та ті, котрі будуть

пов'язані у подальшій розробці. Та й хоча система має вже достатньо пристроїв є варіанти майбутнього вдосконалення, яке зможе поширювати роботу проекти чи створювати схожі по всій країні. Було розроблено план-схему парку, яка слугує опорною точкою в уявленні проекту та подальшому переносі усіх елементів до бази даних; після схеми було ознайомлення з нотацією BPMN та створення функціональну модель діяльності проекту та концептуальну схему бази даних, відповідно, до догляду за зеленими зонами. Опрацьоване використання середовища Process Modeler та створення функціональної моделі систем в нотації IDEF0 та IDEF3. Також, створення різного виду діаграми, які зазначають діяльність роботи системи на кожному рівні; ознайомлення з методологією нотації DFD та вивчення процесу моделювання для обраної сфери за допомогою інструментального середовища AllFusion Process Modeler. Ще треба зазначити вивчену мову об'єктно-орієнтованого моделювання – UML (Unified Modeling Language), в якій було відтворено моделі діяльності компанії.

РОЗДІЛ 3. СТВОРЕННЯ БАЗИ ДАНИХ ТА РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Створення бази даних

Для створення та розробки таблиць баз даних було обрано програмне середовище DBeaver SQL Lite. Програмне середовище, яке має функціонал по створенню нових таблиць та записів чи їх імпорту з Erwin Data Modeler, в якому вони вже були розроблені разом із зв'язки між ними.

На початку, як і говорилося було спроектовано базу даних (рис. 3.1) за допомогою пакету Erwin Data Modeler:

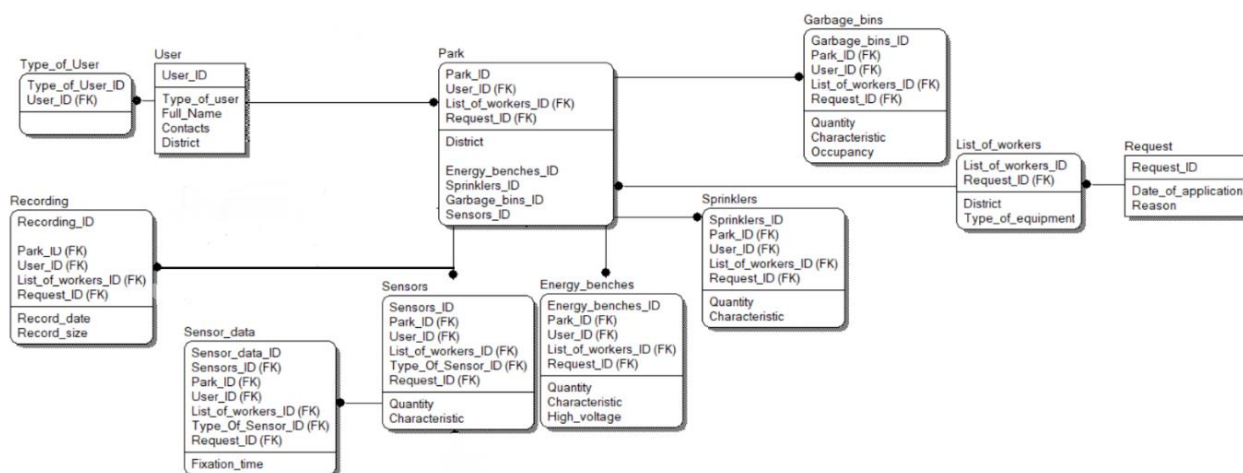


Рис. 3.1. - Створення таблиць та атрибутів

Першим кроком проектування було створення логічної моделі та налагодження зв'язків у ній, після чого вже можна було переходити до фізичної моделі баз даних, попередньо задавши головні ключі (рис. 3.2) кожної таблиці.

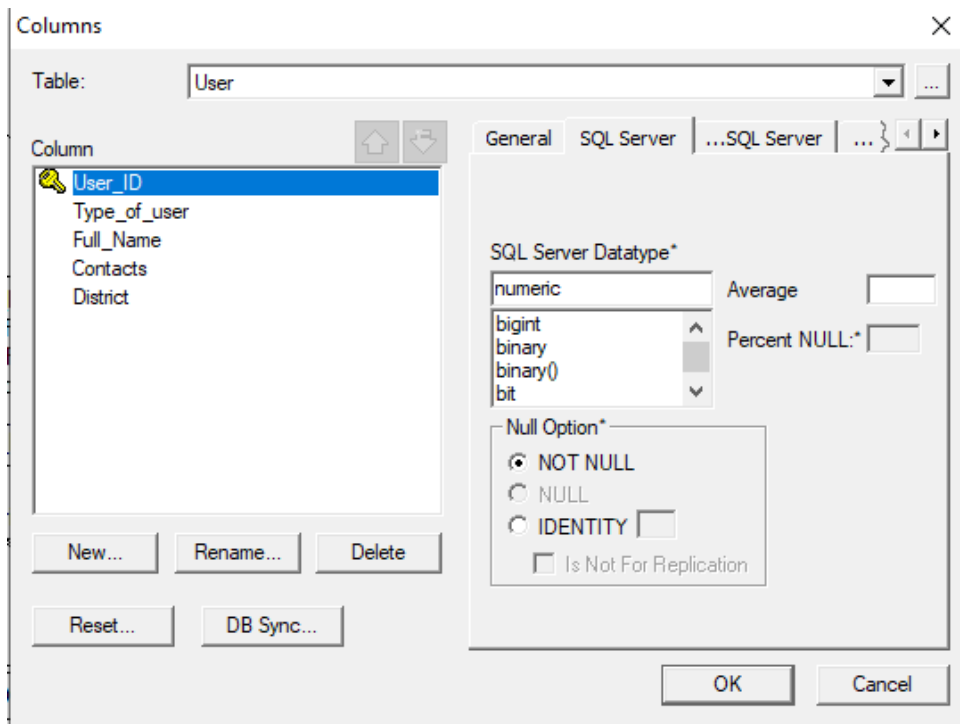


Рис. 3.2. - Встановлений зв'язок між таблицями.

Отримавши DDL файл с моделлю системи почалась робота з програмним середовищем DBeaver, яка базується на SQL Lite. Специфічна та трохи складна до задання параметрів мова, але завдяки програмному середовищу створення було легшим та більш автоматизованим. Тому після пересення моделі, вносячи таблиці в SQL Script, ми отримали зв'язані між собою усі таблиці (рис. 3.3) бази даних паркової зони:

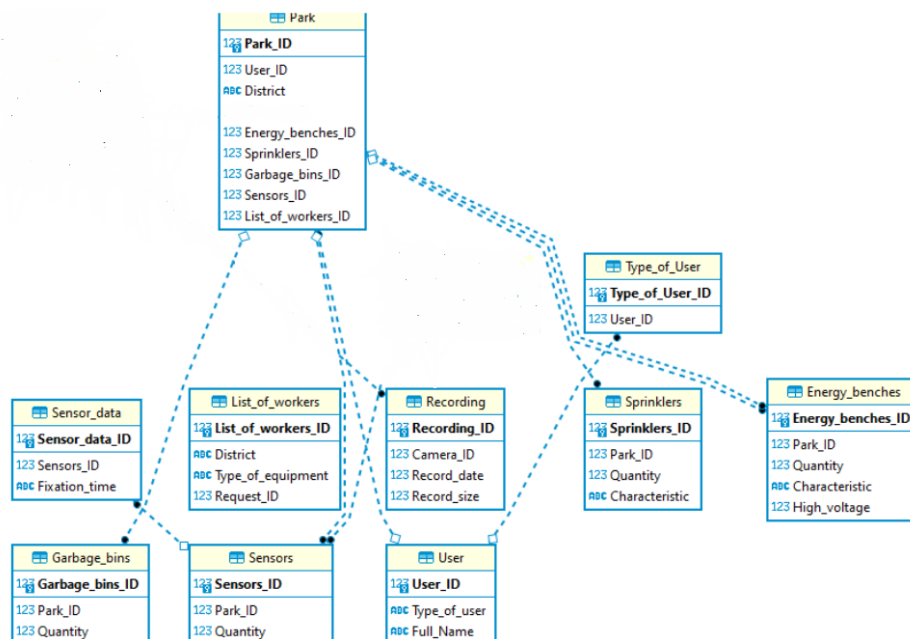


Рис. 3.3. - Зв'язані таблиці баз даних

Також, в самому DBeaver є можливість працювати та обирати кожну таблицю зі списку (рис. 3.4), котрий формується при їх додаванні через команди чи саморуч:

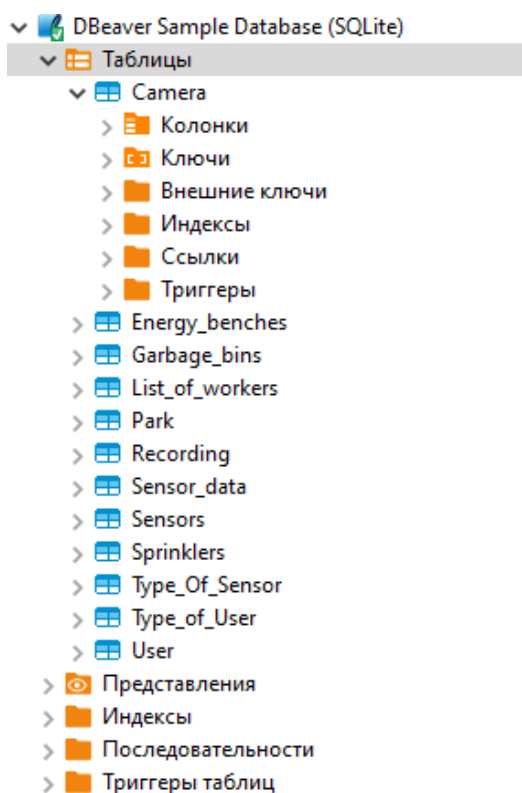


Рис. 3.4. - Перелік усіх доданих таблиць

Ще треба зазначити, що в даному програмному середовищі не використовується синтаксис з DDL файлу, тому його треба було переробити (приклад у додатку Б), та після вивчення базових навичок у мові SQL Lite дані до таблиць були внесені згідно типів їх даних.

3.2 Імпортування баз даних

Як вже було зазначено у попередніх розділах для створення інтерфейсу та подальшому імпорту бази даних було використано сервіс по роботі з базами даних QuintaDB. Він дозволяє автоматизувати бізнес-процеси, керувати, візуалізувати та ділитися даними в індивідуальному додатку. А, також, налаштовувати крос платформні програми за допомогою простої та потужної бази даних. Виконувати численні завдання, розрахунки та аналіз даних за допомогою інтуїтивно зрозумілих конструкторів та інструментів. Загалом, його можна вважати одним

великим конструктором баз даних для налаштування зав'язків та створення робочого прикладу системи схожих проектів.

Початковим етапом роботи з інтерфейсом, звісно буде першочергове імпортування бази даних до сервісу (рис. 3.5). QuintaDB дозволяє робити це декількома шляхами: налаштування самотужки, створення форми по вже існуючим у базі шаблонам чи завантаження Excel файлу, який автоматично створить таблиці та форми заповнення для них. Щоб відтворити останній варіант було збережено файли таблиці у вигляді Excel файлу з вже відомого програмного додатку DBeaver [10]. Після чого файли були перенесені до сервісу, де відповідно заданих показників та ID було створено таблиці з відповідними формами.

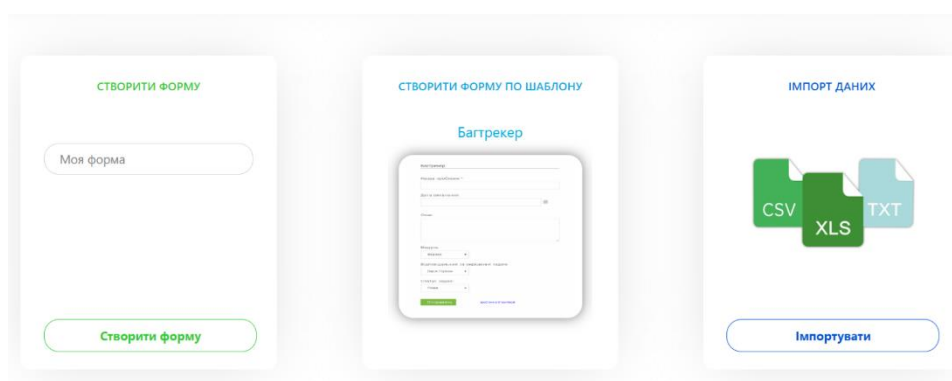


Рис. 3.5. - Імпортування створеної бази даних

Хоча й імпортування таблиць системи до сервісу є легким, краще знати налаштування самотужки також, бо при потребі додати ще одну таблицю, вона може не зв'язатися з вже існуючими, тому створення власних таблиць та форм теж потрібно для подальшої роботи. Тож, обираючи створення форм власноруч треба задати параметри таблиці такі як назва, дані, які будуть зберігатись, та обрати зв'язок в залежності від зав'язків, які були створені раніше в програмних додатках DBeaver та Erwin Data Modeler.

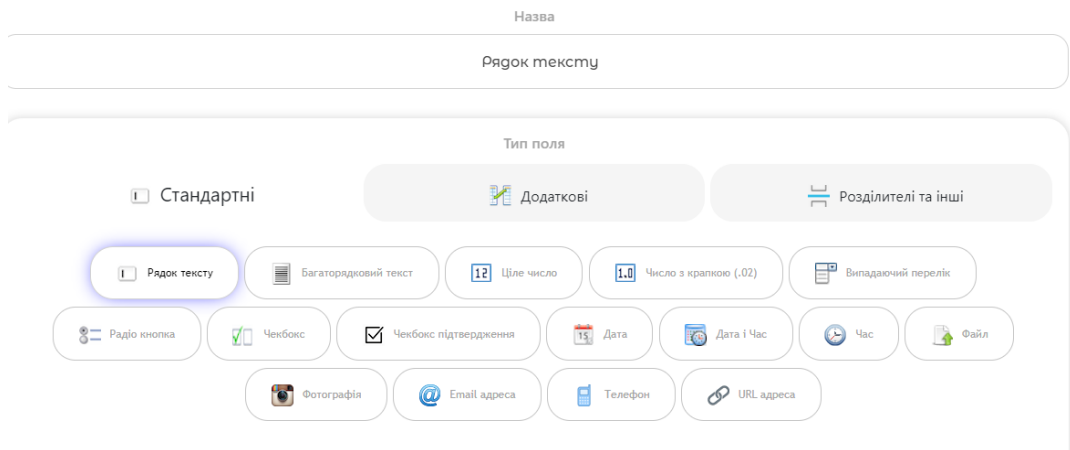


Рис. 3.6. - Налаштування параметрів таблиці

Таким чином надаючи початкові характеристики (рис. 3.6) першій таблиці треба урахувати, що подальший зв'язок буде з'єднано через обране поле, бо сервіс вже має внутрішню базу даних, яка не є видимою для усіх користувачів, а тільки для розробника. Зроблено це для того, щоб не навантажувати таблиці ще однією формою, яка не є важливою при додаванні інформації звичайним користувачем. Тому при створенні вже другої таблиці треба обирати тип зв'язку (рис. 3.7) з перелічених, також, з можливістю різного відображення даних при заповненні форм.

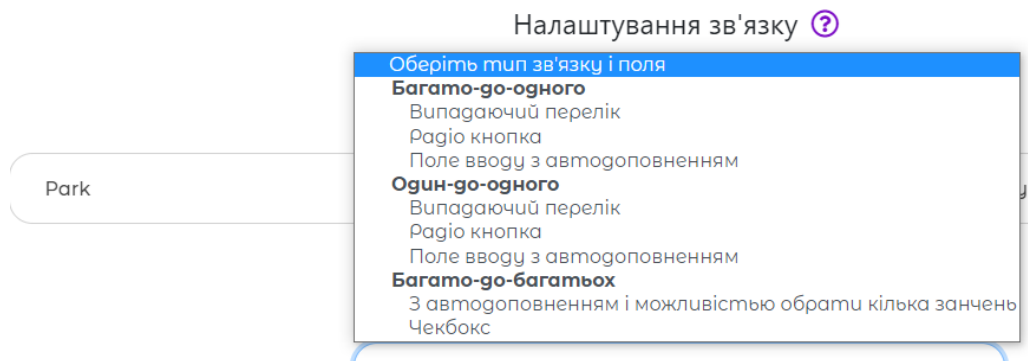


Рис. 3.7. - Налаштування зв'язків

Додавши вже хоча б дві таблиці треба переходити до налаштування форм (рис. 3.8), тобто вікна в якому вносяться дані чи показники залежно від таблиці до якої вона прив'язана. Для цього є спеціальна вкладка з конструктором форм, де можна додати нове поле вводу даних, оновити вже існуюче чи взагалі видалити. Кожна форма, як і при створенні таблиць (рис. 3.9) у минулих програмах має окремий тип даних, на сайті є перелік цих даних залежно від потреб, що дає

можливість змінювати та налаштовувати таблицю по різному для окремих показників.

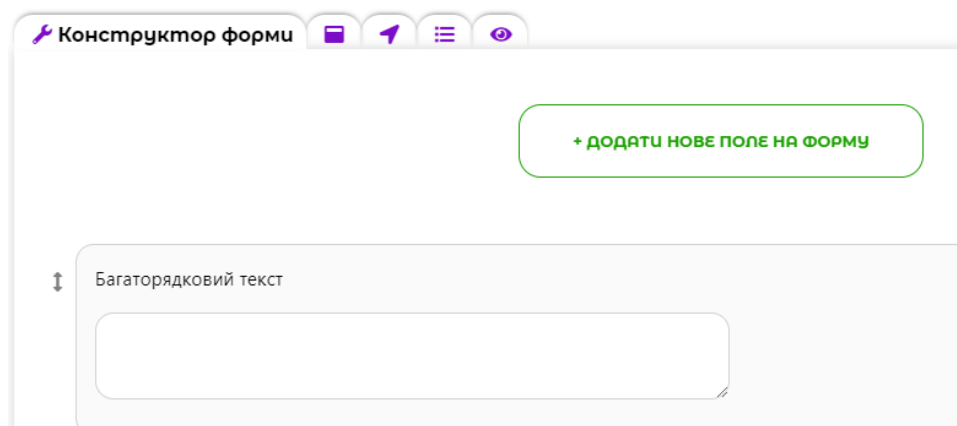


Рис. 3.8. - Налаштування форм для таблиць

Наприклад, дата спрацювання датчиків руху влаштованих у ліхтарі. Для такої задачі можна обрати різний тип встановлення дати, чи взагалі обрати час з датою, але для багатой кількості спрацювань це не є раціональними даними, тому було обрано знати повну дату без часу, але з кількістю спрацювань, які будуть прив'язані до типу сенсора, який в свою чергу відсилає на таблицю з датчиками та їх розташуванням у конкретних парках.

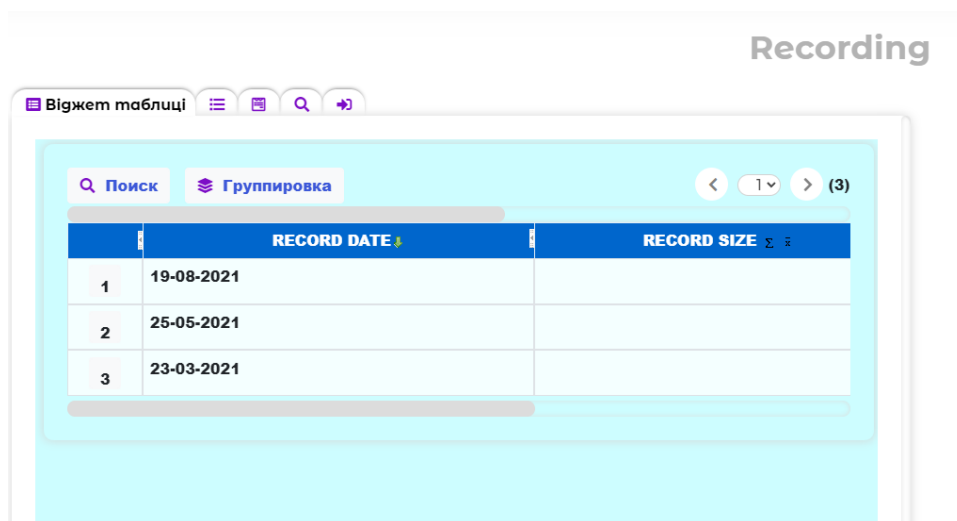


Рис. 3.9. - Вигляд налаштованої таблиці

Окрім дати та часу є ще інші типи даних такі, як числові показники, текстові, посилання, прив'язки до карт з розташуванням об'єкту та інше.

3.3 Реалізація інтерфейсу

Створення таблиць, налаштування їх зв'язку та заповнення є тільки початком в створенні сервісу. Далі іде налаштування інтерфейсу, налаштування можливостей таких як, перегляд карти з розташуванням парку, додавання різного рівня користувачів та дизайн, тобто розташування меню з вкладками таблиць та форм. Оскільки цей проект є унікальним, власними зусиллями було створено логотип веб-порталу та вигадано назву Green Area (рис. 3.10). Його було закріплено у шапці сайту, та надано можливості переходу до головної сторінки при натисканні.

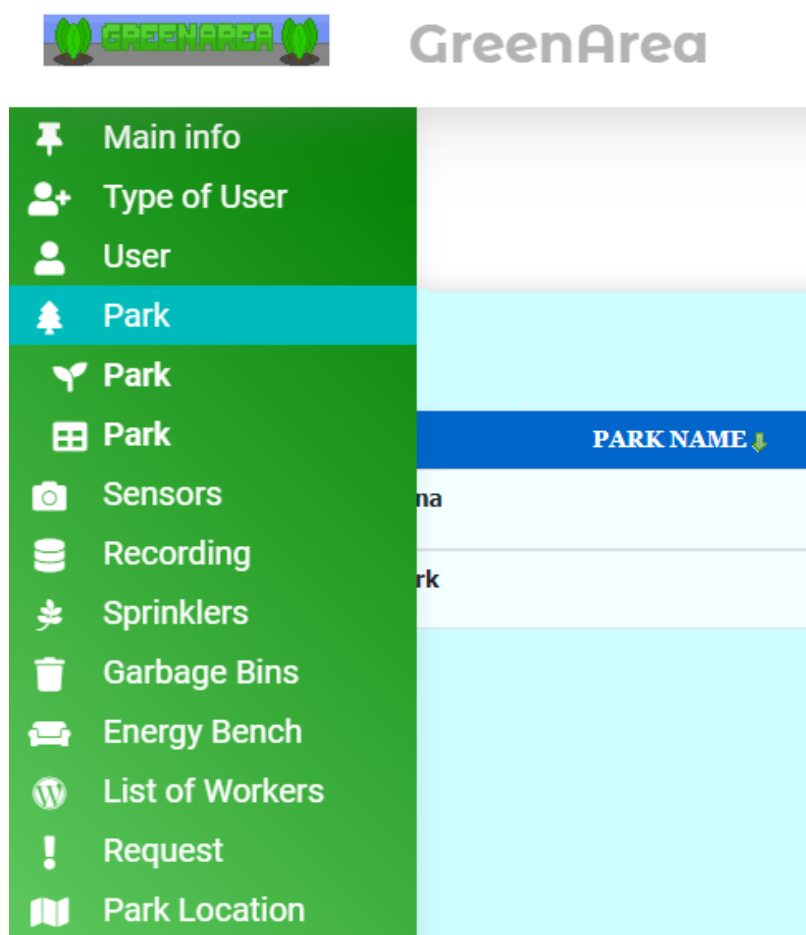


Рис. 3.10. - Інтерфейс до веб-порталу

Налаштування таблиць та форм має окреме місце в сервісі (рис. 3.11), яке надає можливість передивитись вже налаштовані та оновлювати їх окремо, але завжди маючи спробу продивитись результат. Після налаштування цього, також, вже можна розташувати самі таблиці по меню інтерфейсу, який поряд та яка сторінка буде головною.

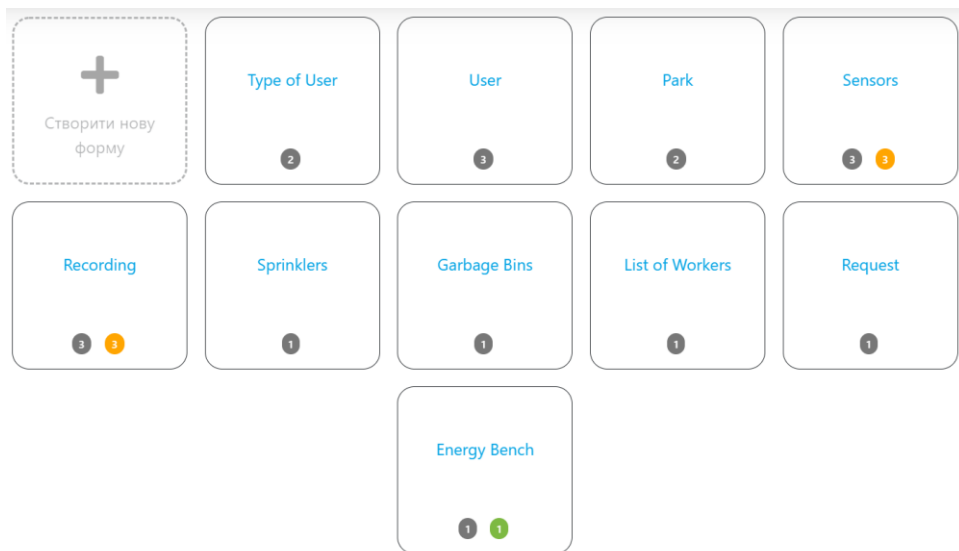


Рис. 3.11. - Вигляд таблиць у розділі додавання форм

Для цього сервіс має також окреме налаштування (рис. 3.12), де можна додати головну сторінку, нову таблицю, форму чи додати щось до вже існуючих.

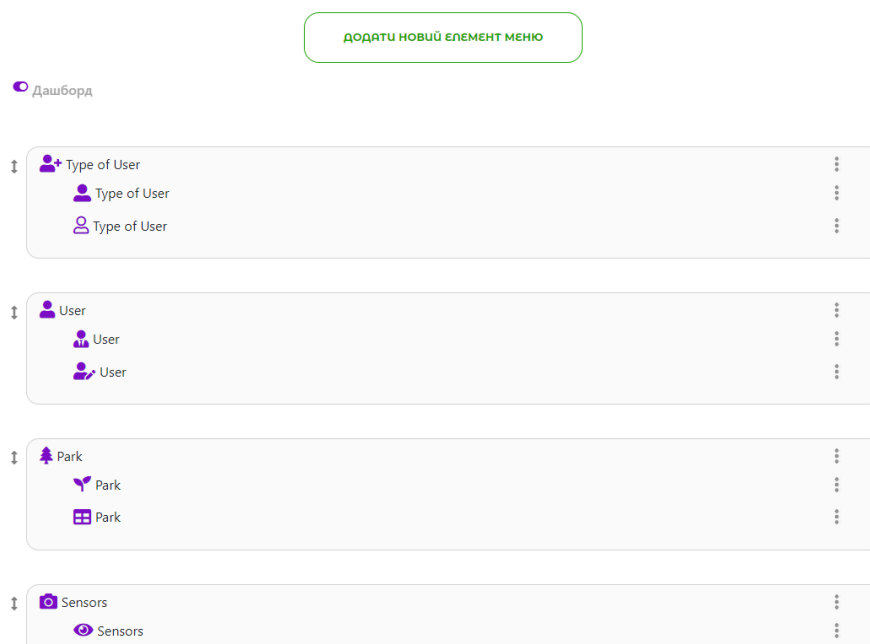


Рис. 3.12. - Налаштування меню інтерфейс

Оскільки це не є чимось, що займає багато часу та чимось дуже важливим, наступним кроком буде створення головної сторінки, котра буде зустрічати кожного користувача при потраплянні на веб-порталу. Головною сторінкою не обов'язково має бути тільки інформація чи фотографії парку, вона може бути більш інформативною та передавати саму сутність системи, бо сам портал створений для у більшості для робітників, а не для відвідувачів парку. Тому при переході на цю сторінку буде показуватись діаграми з даними по встановленим датчикам та кількість їх спрацювань по певних днях. Це може допомогти

вираховувати завантаженість парку, а вночі бути охороною системою, якщо парк усе ж таки буде зачинятись, враховуючи обладнання, котра там буде впроваджене.

Взагалі, сервіс надає можливість налаштувати пріоритетність заповнення даними та створення спеціального критерію пошуку у таблиці, якщо це є потрібним. Зазначається, що у ПРО версії цього сервісу більший функціонал, але для курсової роботи, було використано безкоштовні версію з певними обмеженнями у кількості полів, записів та діаграм, які далі використовуються для візуалізації даних з таблиць на головній сторінці веб-порталу.

Тому, усього було створено три різних діаграми [12], які беруть дані з таблиць, які були додані до перегляду в інтерфейсі. Стовпчаста (рис. 3.13) та кругова діаграма (рис. 3.14) використовуються для показу кількості задіяних датчиків руху у парку та їх загальне розташування, та лінійна діаграма, котра вказує скільки було спрацювань обраного дня у робочий час.

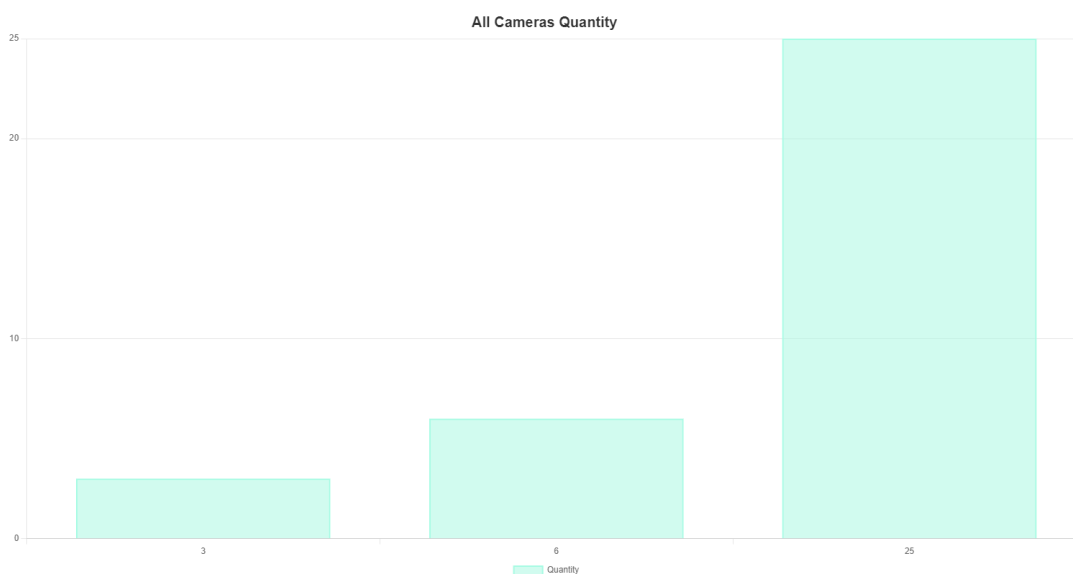


Рис. 3.13. - Діаграма кількості сенсорів у парку

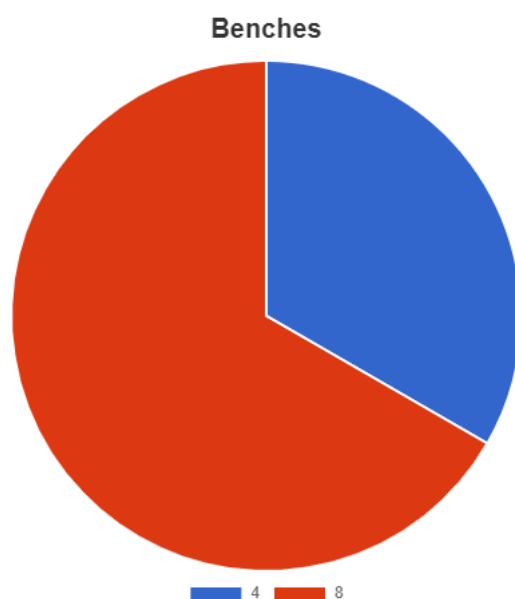


Рис. 3.14. - Кругова діаграма активних датчиків руху

Як вже було зазначено останньою діаграмою була обрана лінійна діаграма спрацювання датчиків руху з ліхтарів (рис. 3.15). Таким чином, головна сторінка веб-порталу відображає графічно зображені дані для зрозумілого представлення користувачем з чим він має справу та що ще можна було б додати в залежності від потреб користувача та додаванні нових девайсів.

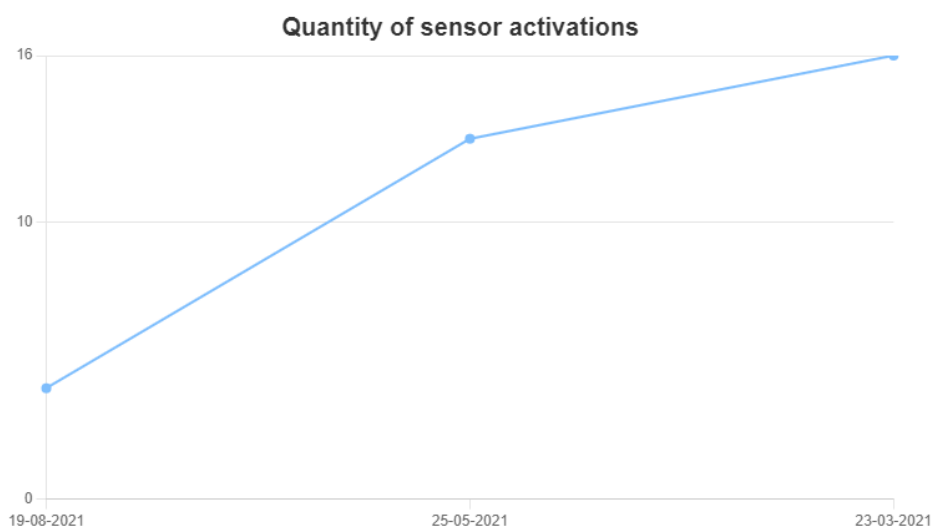


Рис. 3.15. - Лінійна діаграма спрацювання датчиків руху

Хоча, це ще є не фіналом цієї сторінки, бо ще є можливість додати карту (рис. 3.16) з позначеними парками до відповідного адресу, які були внесені до таблиць де треба використовувати прив'язку до локації для повного уявлення масштабу системи, бо сам парк великий за площею. Тому цей парк було відзначено на карті завдяки адресу взятому з таблиці “Парк”.

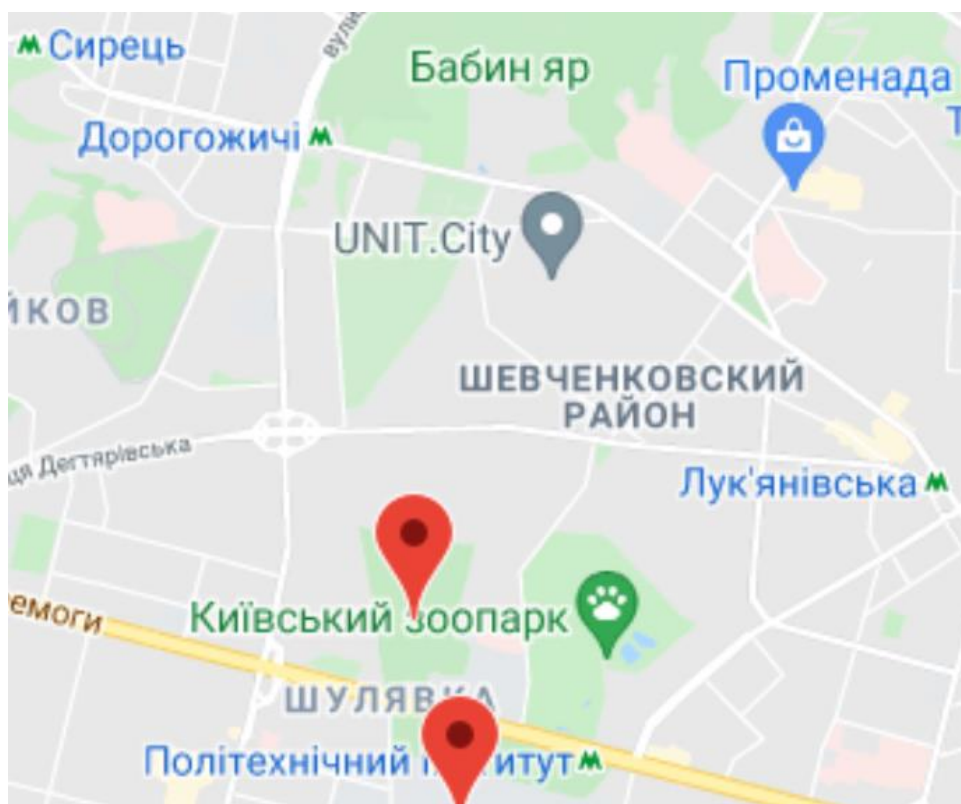


Рис. 3.16. - Позначений парк імені Пушкіна на карті

Також, було додано лічильник (рис. 3.17) нових записів про запити до бригад, щоб мати інформацію скільки запитів взагалі було відправлено чи скільки було додано нових за 24 години (рис. 3.18), залежить від потреби користувача. Подібні лічильники слугують для своєчасного отримання інформації щодо нових звернень, щоб не шукати окрему таблицю на сайті, а вже після отримання інформації можна відсортувати запити по новизні та віднайти потрібний. Загалом, даний лічильник можна використати й для інших таблиць для повного моніторингу системи вже на головній сторінці.

 New requests per 24 hours

Рис. 3.17. - Лічильник нових запитів для бригад

	DATE OF APPLICATION	PROBLEM INFO
1	14-06-2022 14:20:48	Problem with connection
2	03-06-2022 18:00:00	Changing energy block from bins
3	01-06-2022 8:21:00	Disconnection between motion sensors and server.
4	08-07-2021 9:00:00	Sensors broke down
5	09-06-2021 12:16:48	Lost connection between benches

Рис. 3.18. - Таблиця запитів на виїзд до парку

У подальшій переробці веб-порталу з сервісу QuintaDB в повноцінний сайт чи додаток є можливим підключення даних з реально існуючих елементів системи, що надає більше можливостей по модернізації головної сторінки, бо вивід даних з активних зон поливу чи, наприклад, картинки з камер відеозапису, для швидкої перевірки відповідних до запиту зон парку.

3.4 Реалізація авторизації у системі

Кожна система має користувачів різного рівню, ця не є виключенням, бо є дані, які може переглядати тільки адміністратор чи супер користувач. В даному випадку такими даними виходять інформація про тип користувача, тобто робітника парку. Дану таблицю та відповідну до неї форму може використовувати тільки адміністратор, який і має право надавати конкретний тип робітника, в той час коли дані користувача (рис. 3.19), та його контакти є доступні для перегляду усім робітникам для розуміння до якого парку відноситься людина та які контакти для з'єднання має. Хоча, при бажанні, це також можна змінити, та взагалі заборонити доступ до цих даних усім, окрім адміністратора.

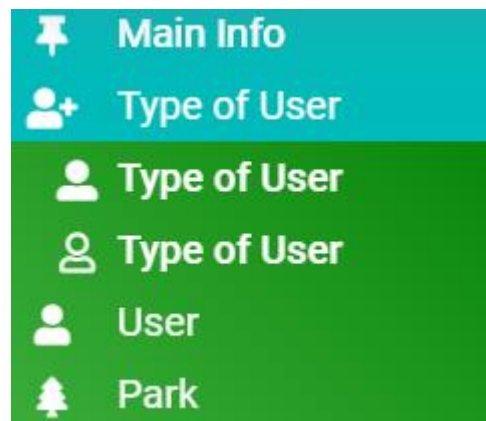
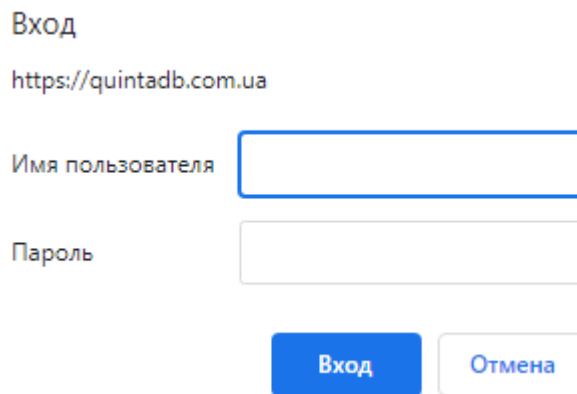


Рис. 3.19. - Меню з вкладками веб-порталу

Але не залежно від рівня користувача кожен має авторизуватися для відвідування веб-порталу (рис. 3.20). Вхід до порталу виконується з використанням налаштованих адміністратором паролю та логіну, які є словом - Park.



Вход

https://quintadb.com.ua

Имя пользователя

Пароль

Рис. 3.20. - Вхід до веб-порталу

Перше, що буде зустрічати користувача на сайті це головна сторінка (рис. 3.21), котра вже була налаштована раніше. Наступними кроками можуть бути переходи для перегляду таблиць та їх наповнення. Кожен зі звичайних користувачів зможе переглядати таблиці та змінювати їх дані, за потребою, тобто додавати нові записи, тим самим оновлюючи статистику.

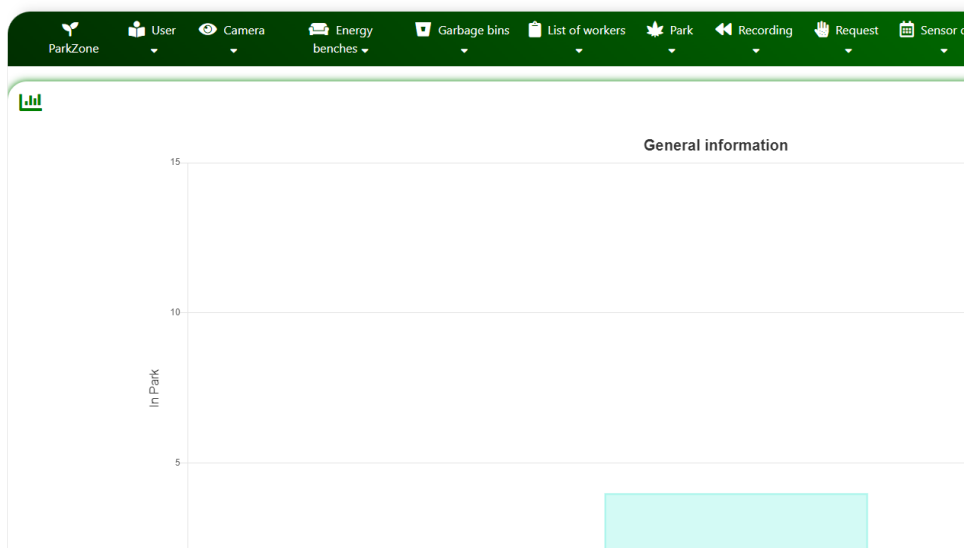
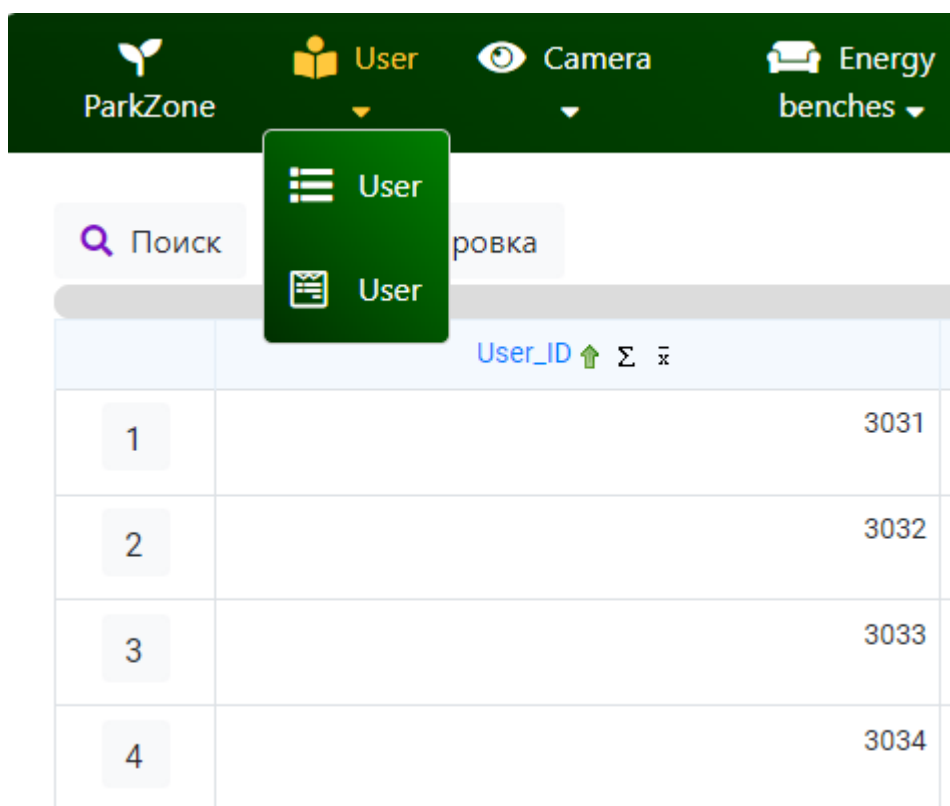


Рис. 3.21. - Діаграми на головній сторінці

Для перегляду інших таблиць з даними треба обрати відповідний запис, для цього кожна з вкладок має свою особисту позначку, що дає змогу розрізняти їх не тільки по назві (рис. 3.22).



	User_ID ↑	Σ	̄x
1	3031		
2	3032		
3	3033		
4	3034		

Рис. 3.22. - Таблиця з записами

Щоб ввести новий запис до таблиці треба перейти до заповнення форм даної таблиці, також, за відповідною вкладкою (рис. 3.23). Форми мають різні формати заповнення даних, зазвичай це зрозуміло з назви чи можливих варіантів для вибору, але якщо намагатися порушити правила внесення значень та інформації, то буде видана помилка, яка попросить перевірити коректність вводу даних. Це зроблено, аби вберегти систему від порушення зав'язків та її загального зламу.

Рис. 3.23. - Форма для додання нових записів

3.5 Таблица запитів

Однією з головних функцій сайту має бути не тільки моніторинг показників, але й можливість моніторингу запитів на перевірку обладнання, яке ці показники і надає. Тому для цього є окрема таблиця з запитами, яка зберігає інформацію про дати заявки цього запиту, до якого району належить бригада робітників та яка в цілому проблема сталась, тобто причину запиту. Усі ці дані додаються до таблиці “Request” (рис. 3.24), після чого можна прив’язати цей запит до певної команди, яка в свою чергу пов’язана з парком.

request

	DATE OF APPLICATION	PROBLEM INFO	BRIGADE
1	08-07-2021 9:00:00	Sensors broke down	Solomesky district

Рис. 3.24. - Таблица “Request”

А прив’язка до парку здійснюється через зв’язок таблиця парку (рис. 3.25) та запиту, завдяки полю “Ім’я бригади”. До кожного парку є своя власна міська команда робітників з комунальних послуг, тому найзручніший зв’язок саме через

це поле, котре в свою чергу прив'язано до розташування парку, що надає можливості перейти одразу до карти та переглянути маршрут до нього.

Park



PARK ID	PARK NAME	ADDRESS	BRIGADE NAME
1	31 KPI Park	Peremohy, 37k, KPI	Solomesky district
2	3 Pushkina	avenue Victory Kyiv, Ukraine 02000	Solomesky district

Рис. 3.25. - Таблиця з даними парку

Ще треба зауважити, що карта має й окреме вікно, яке зображує позначками, які парки вже внесені до бази даних, що знаходиться поблизу, та, взагалі, це карта Google (рис. 3.26), що вже для багатьох зрозуміло та легко у користуванні.

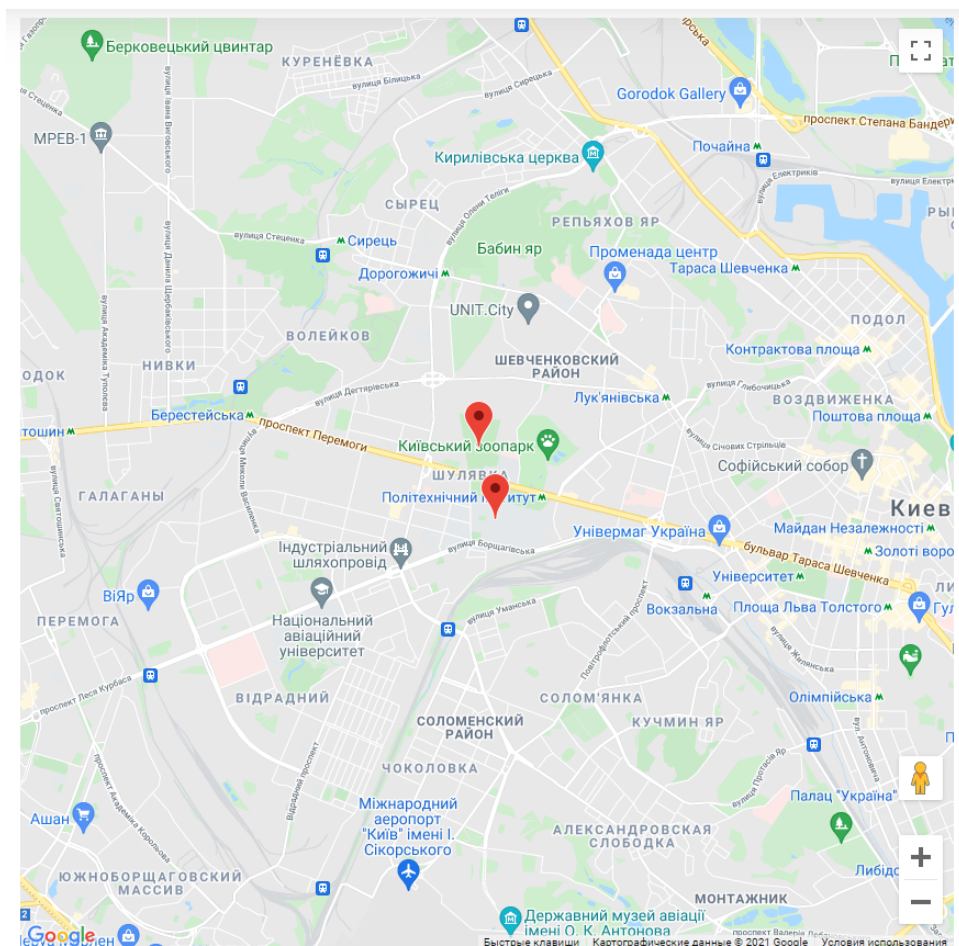


Рис. 3.26. - Карта з позначеними парками у таблиці

Тому автоматичне сповіщення бригади про потребу у виїзді до парку є однією з переваг даної системи, бо не треба нікому дзвонити, бо датчики та інші розумні елементи системи самі покажуть що вони не працюють, і коли це вже станеться - спеціалісти зможуть вчасно відновити системи та одразу перевірити

це на сайті, що економить багато часу та надає можливості зосередитися на інших елементах системи.

3.6 Таблиці діючих елементів системи

Основною ж частиною веб-порталу Green Area є таблиці усіх елементів системи (рис. 3.27). Завдяки їм, можна знати, скільки взагалі приладів є на території парку, скільки працює, які вже відпрацювали своє та які потребують налаштування. Заповнення форм не є особливістю у свій час коли зв'язки елементів між собою та їх індивідуальні дані є. Завдяки цьому, можна поєднати сервісні додатки пристроїв з системою та робити налагодження безпосередньо з самого сайти.

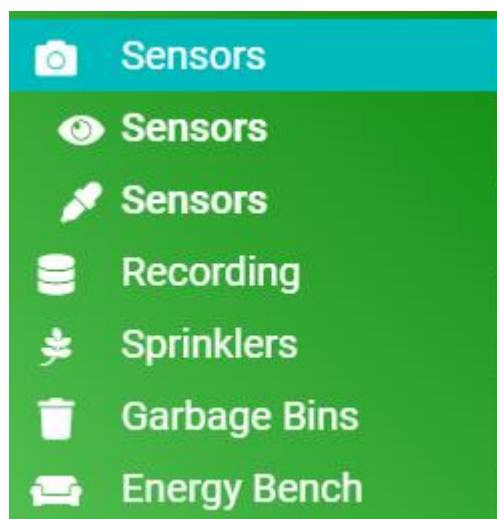


Рис. 3.27. - Меню елементів системи

Таблиці можуть автоматично відновлювати дані щодо спрацювання чи активації, коли у свій же час, адміністратор зможе контролювати полив рослин дистанційно під'єднавши додаток контролера поливу до всієї системи. Вмикати чи вимикати датчики руху (рис. 3.28) залежачи від потреби, бо, як вже було зазначено, датчики руху можна використовувати ще й як аналітичній засіб.

		Sensors	
	QUANTITY	TYPE OF SENSOR	
1	25	Motion	
2	6	Motion	
3	3	Laser	

Рис. 3.28. - Таблиця датчиків руху

Дані щодо сміттєвих баків також повинні оновлюватися автоматично, додавши лише одне поле, можна виявляти скільки баків заповнено, та де саме, якщо відтворити схему парку у подальших розробках. Розумні лавочки можуть зберігати дані з мережі, навіть без участі адміністратора чи робітників, просто використовуючи посилання з розкладом автобусів, а карта розташування вже й так буде демонструватися з самого веб-порталу за потребою.

Рис. 3.29. - Форма заповнення таблиці розумних лавочок

Кожні форми (рис. 3.29) заповнюються людиною та зв'язуються автоматично [13], можливо, у подальшому розвитку проекту, є доцільним зробити архів даних та запитів, котрий буде зберігатися на хмарі. Тим самим налаштувати, коли саме повинна бути архівація у формі, додавши поле дати та часу. Збереження такої інформації, як наповненість баків та запити є актуальними тільки до того, поки проблему не вирішено, а після це може бути просто архівом для перегляду, якщо будуть схожі питання чи непередбачені обставини.

Висновок до розділу 3

Під час виконання цього розділу, було вивчено концепцію та функціональні можливості сучасних засобів для проектування баз даних на прикладі пакету Erwin Data Modeler; опрацьоване проектувати бази даних за допомогою пакету Erwin Data Modeler. Перенесення для подальшого форматування таблиць до програмного середовища DBvaer. Створення інтерфейсу клієнтського додатку на сервісі QuintaDB та меню ІС з проектуванням форми введення даних для незалежних таблиць. Також, було набуто навичок по створенню діаграм у додатку, використовуючи дані з таблиць.

ВИСНОВКИ

В даній дипломній роботі було проведено роботу по розробці інтерфейсу до веб-портала для системи “Моніторинг роботи автоматизованого парку”. Для успішного відтворення першого варіанту цього проекту було проаналізовано аналогічні рішення за даним питанням та можливі технології з використанням яких реалізація буде можливою. Також, були створені контекстна, логічна та фізична моделі за допомогою засобів AllFusion Process Modeler та Erwin Data Modeler. Відповідно до цих моделей було спроектовано базу даних з таблицями у програмному середовищі DBeaver, яке базується на мові SQL Lite. Після чого за допомогою QuintaDB була розроблена тестова версія інтерфейса для веб-порталу з можливістю редагування даних у таблицях та перегляду основних характеристик девайсів паркової зони.

Створений інтерфейс дає можливість переглядати дані експортовані з бази даних у форматі Excel таблиць та виводи обрані дані до діаграма, котрі зазначені на головній сторінці. Розроблений веб-портал є тільки початком даного проекту, бо в подальшому, звісно, треба налаштувати зв'язок з базою даних без зайвих експертів та на пряму до повноцінного сайту. Ще, виводом може слугувати не лише діаграми з таблицями, а й розташування парку та усіх технологій, які використовуються на його території. Якщо, насправді, спробувати розробити цей проект у справжніх масштабах міста, то це значно може полегшити догляд за парками, та загалом, зберігати зелені зони, яких багато, у належному стані.

При подальшому розгляді проекту після його відтворення є можливим його вдосконалити. Наприклад, можна додати лінійний лазерний датчик руху, який при рості газону по периметру, буде розпізнавати це як перешкоду, та викликати розумних роботів газонокосарок, які зможуть без залучення людини до цього процесу підстригти газон та повернутися на своє місце. Визначення росту газону саме по периметру, допомагає закріплювати лазерні промені, що є економією самих лазерів та зменшення можливих помилкових перешкод, бо якщо газон буде рости по периметру так само буде і в центрі ділянки. Місцем зберігання можуть бути, наприклад, невеликі коробки розташовані по зелених ділянках парку з якими можна поєднати контролери поливу, щоб заощадити місце при плануванні та

захистити самі контролери. Приклад використання роботів газонокосарок вже був впроваджений в одному з центрів надання адміністративних послуг одного з селищ в Україні. Зволожувачі повітря, які використовуються в кафе та на деяких алеях міст, теж можна назвати гарним рішенням для подальшої модернізації даного проекту. Така технологія була впроваджена в парк коло драматичного театру міста Маріуполь, а зараз є на центральній вулиці Хрещатик нашої столиці. Зменшення ризику погіршення здоров'я від спеки у відвідувачів та довгі прогулянки під пекучим сонцем стають можливі з впровадженням саме цієї технології.

Підводячи підсумки, можна зазначити, що результатом даної дипломної роботи є проектування системи для екологічної безпеки міста, моделювання та створення бази даних з подальшою розробкою тестового інтерфейсу до веб-порталу, не складного та зрозумілого для користувача з робочою назвою "Greenarea".

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Розумні парки Європи, оновлено 2021 URL: <https://iprovsvita.com/top-5-najkrasyvishykh-parkiv-ievropy/> (Дата звернення 2.05.2022)
2. Поняття IoT-платформи на прикладі системи від Kyivstar, 2021 URL: <https://hub.kyivstar.ua/news/shho-take-iot-platforma-i-yake-ii-priznachennya/> (Дата звернення 2.05.2022)
3. Парк імені Пушкіна(загальна інформація, щодо заснування), 2022 URL: <https://uk.wikipedia.org/wiki/> (Дата звернення 3.05.2022)
4. Автоматичні системи поливу Nigarden system, 2021, URL: <http://avtopoliv.in/> (Дата звернення 3.05.2022)
5. Розумі лавочки Kiev Smart City System, 2022, URL: https://hotline.ua/dacha_sad/sadovaya-mebel/44618/(Дата звернення 3.05.2022)
6. Розумні сміттєві баки Smart can system, 2020, URL: <https://www.dw.com/uk/>(Дата звернення 3.05.2022)
7. Ліхтарів з датчиком руху, характеристики приладів з цінами, 2020 URL: <https://powerlux.com.ua/catalog/avtonomnoe-osveschenie>(Дата звернення 3.05.2022)
8. Рейтинг захисту IP, система оцінки приладів, 2019 URL: <https://asoft.by/chto-takoe-ip65-ili-kak-opredelit-stepen-zashchishchennosti-oborudovaniya> (Дата звернення 3.05.2022)
9. Сайт для створення схем Draw.io, проектування та відтворення, 2022 URL: <https://app.diagrams.net/> (Дата звернення 3.05.2022)
10. Проекти розумних систем автоматизації, 2022 URL: <https://www.anwit.kiev.ua/index.php/katalog/lazernye-datchiki-triangulyatsiya> (Дата звернення 3.05.2022)
11. Проектні рішення датчиків руху, 2022 URL: <https://rutd-ksk.com/datchik-dvizheniya-s-peredachey-na-mobilnyu-telefon/> (Дата звернення 3.05.2022)
12. Сервіси, які працюють з базами даних, 2022 URL: http://www.ubnt.su/solutions/big_networks/park.htm (Дата звернення 3.05.2022)

13. Запит на проекти автоматизованих систем в Україні, 2022 URL:
<https://local.com.ua/forum/topic/Дата звернення 3.05.2022>)

ДОДАТОК А

Схема план парку з розташуванням усіх обраних елементів автоматизації
паркової території.

Листів 1

Розробник

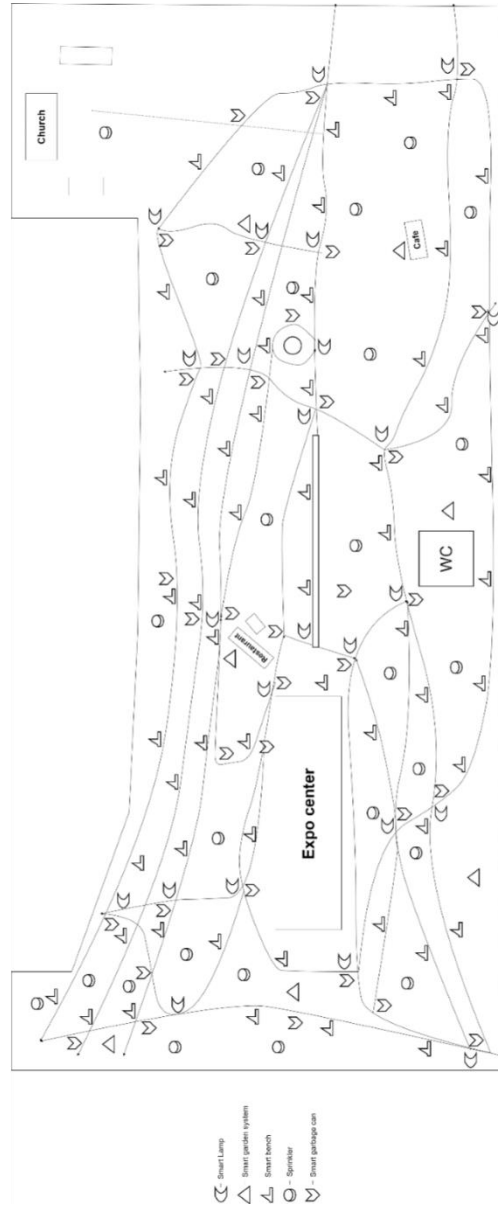
Керівник

Ярослав НІЗОВ

Мирослава ГЛАДКА

Київ – 2022

Схема план парку з розташуванням усіх обраних елементів автоматизації паркової території.



ДОДАТОК Б

Код для створення таблиці у програмному середовищі DBeaver базуючись на
DDL файлі

Листів 1

Розробник

Керівник

Ярослав НІЗОВ

Мирослава ГЛАДКА

Київ – 2022

Код для створення таблиці у програмному середовищі DBeaver базуючись на DDL файлі

```
CREATE TABLE Camera
(
Quantity      numeric NULL ,
Characteristic varchar() NULL ,
Park_ID       integer NOT NULL ,
Camera_ID     integer IDENTITY ( 1,1 )
)
go
ALTER TABLE Camera
ADD CONSTRAINT XPKCamera PRIMARY KEY CLUSTERED (Camera_ID ASC)
go
```

Код для створення таблиці після перетворення для мови SQL Lite:

```
CREATE TABLE Camera
(
Quantity      numeric NULL ,
Characteristic varchar(56) NULL ,
Park_ID       integer NOT NULL ,
Camera_ID     integer IDENTITY ( 1,1 ) ,
CONSTRAINT XPKCamera PRIMARY KEY (Camera_ID)
)
```

ДОДАТОК В

Частковий код реалізації веб-порталу на сервісі разом з таблицями баз даних

Листів 11

Розробник

Ярослав НІЗОВ

Керівник

Мирослава ГЛАДКА

Київ – 2022

Частковий код реалізації веб-порталу на сервісі разом з таблицями баз

даних:

```
<div id='portalheightBWONdK3XdM4WYWQnyiSkW' style='display:none'></div><div id='mth-portalrhbBWONdK3XdM4WYWQnyiSkW'>Зацените мой <a href='bBWONdK3XdM4WYWQnyiSkW'>клиентский портал</a></div><script type='text/javascript'>var portalurl_height = 'https://quintadb.com.ua/widgets/bVW5XwzCjfWOTjWQhdRHSu/get_portal_height/bBWONdK3XdM4WYWQnyiSkW.j';var portalrhbBWONdK3XdM4WYWQnyiSkW;(function(d, t) {var s = d.createElement(), options = {'userName':'','portalHashURL':'/apps/bVW5XwzCjfWOTjWQhdRHSu/portals/bBWONdK3XdM4WYWQnyiSkW/widget','portalHash':'portalrhbBWONdK3XdM4WYWQnyiSkW', 'autoResize':true,'height': '500px', 'heightURL': portalurl_height, 'formID': 'bBWONdK3XdM4WYWQnyiSkW', 'async':true,'host':'quintadb.com.ua', 'header':'show','ssl':true};s.src = ('https:' == d.location.protocol ? 'https://' : 'http://') + 'quintadb.com.ua/scripts/portal.js'; s.onload = s.onreadystatechange = function() {var rs = this.readyState; if (rs) if (rs != 'complete') if (rs != 'loaded') return; try { portalrhbBWONdK3XdM4WYWQnyiSkW = new mthPortal();portalrhbBWONdK3XdM4WYWQnyiSkW.initialize(options);portalrhbBWONdK3XdM4WYWQnyiSkW.display();portalrhbBWONdK3XdM4WYWQnyiSkW.addResizeScript(); } catch (e) {}}; var scr = d.getElementsByTagName(t)[0], par = scr.parentNode; par.insertBefore(s, scr); })(document, 'script');</script><iframe width='800px' height='500px' src='https://quintadb.com.ua/apps/bVW5XwzCjfWOTjWQhdRHSu/portals/bBWONdK3XdM4WYWQnyiSkW/widget' scrolling='auto' frameborder='0'></iframe>ALTER TABLE SprinklersADD CONSTRAINT R_8 FOREIGN KEY (Park_ID) REFERENCES Park(Park_ID)ON DELETE NO ACTIONON UPDATE NO ACTIONgoALTER TABLE UserADD CONSTRAINT R_2 FOREIGN KEY (Type_of_User_ID) REFERENCES Type_of_User(Type_of_User_ID)ON DELETE NO ACTIONON UPDATE NO ACTIONgoALTER TABLE UserADD CONSTRAINT R_3 FOREIGN KEY (Park_ID) REFERENCES Park(Park_ID)ON DELETE NO ACTIONON UPDATE NO ACTIONgoCREATE TRIGGER tD_Garbage_bins ON Garbage_bins FOR DELETE AS/* ERwin Builtin Trigger *//* DELETE trigger on Garbage_bins */BEGINDECLARE @errno int,@errmsg varchar(255)/* ERwin Builtin Trigger *//* Park Garbage_bins on child delete no action *//* ERWIN_RELATION:CHECKSUM="0001220c", PARENT_OWNER="", PARENT_TABLE="Park" CHILD_OWNER="", CHILD_TABLE="Garbage_bins" P2C_VERB_PHRASE="", C2P_VERB_PHRASE="", FK_CONSTRAINT="R_9", FK_COLUMNS="Park_ID" */IF EXISTS (SELECT * FROM deleted,Park WHERE /* %JoinFKPK(deleted,Park," = "," AND") */ deleted.Park_ID = Park.Park_ID AND NOT EXISTS ( SELECT * FROM Garbage_bins WHERE /* %JoinFKPK(Garbage_bins,Park," = "," AND") */
```

```

        Garbage_bins.Park_ID = Park.Park_ID
    )
)
BEGIN
    SELECT @errno = 30010,
           @errmsg = 'Cannot delete last Garbage_bins because Park exists.'
    GOTO ERROR
END
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
go

CREATE TRIGGER tU_Garbage_bins ON Garbage_bins FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on Garbage_bins */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insGarbage_bins_ID integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* Park Garbage_bins on child update no action */
    /* ERWIN_RELATION:CHECKSUM="0001623b", PARENT_OWNER="", PARENT_TABLE="Park"
    CHILD_OWNER="", CHILD_TABLE="Garbage_bins"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_9", FK_COLUMNS="Park_ID" */
    IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(Park_ID)
    BEGIN
        SELECT @nullcnt = 0
        SELECT @validcnt = count(*)
        FROM inserted,Park
        WHERE
        /* %JoinFKPK(inserted,Park) */
        inserted.Park_ID = Park.Park_ID
        /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
        select @nullcnt = count(*) from inserted where
        inserted.Park_ID IS NULL
    IF @validcnt + @nullcnt != @NUMROWS
    BEGIN
        SELECT @errno = 30007,
               @errmsg = 'Cannot update Garbage_bins because Park does not exist.'
        GOTO ERROR
    END
    END
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
go
CREATE TRIGGER tD_List_of_Workers ON List_of_Workers FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on List_of_Workers */
BEGIN

```

```

DECLARE @errno int,
        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* Request List_of_Workers on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00025eb6", PARENT_OWNER="", PARENT_TABLE="Request"
CHILD_OWNER="", CHILD_TABLE="List_of_Workers"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_11", FK_COLUMNS="Request_ID" */
IF EXISTS (SELECT * FROM deleted,Request
WHERE
/* %JoinFKPK(deleted,Request," = "," AND") */
deleted.Request_ID = Request.Request_ID AND
NOT EXISTS (
SELECT * FROM List_of_Workers
WHERE
/* %JoinFKPK(List_of_Workers,Request," = "," AND") */
List_of_Workers.Request_ID = Request.Request_ID
)
)
BEGIN
SELECT @errno = 30010,
        @errmsg = 'Cannot delete last List_of_Workers because Request exists.'
GOTO ERROR
END
/* ERwin Builtin Trigger */
/* Park List_of_Workers on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="Park"
CHILD_OWNER="", CHILD_TABLE="List_of_Workers"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_10", FK_COLUMNS="Park_ID" */
IF EXISTS (SELECT * FROM deleted,Park
WHERE
/* %JoinFKPK(deleted,Park," = "," AND") */
deleted.Park_ID = Park.Park_ID AND
NOT EXISTS (
SELECT * FROM List_of_Workers
WHERE
/* %JoinFKPK(List_of_Workers,Park," = "," AND") */
List_of_Workers.Park_ID = Park.Park_ID
)
)
BEGIN
SELECT @errno = 30010,
        @errmsg = 'Cannot delete last List_of_Workers because Park exists.'
GOTO ERROR
END
/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END
go
CREATE TABLE Park
(
    Park_name        varchar(56) NULL ,
    Brigade_name     varchar(56) NULL ,
    Park_ID          integer IDENTITY ( 1,1 ) ,
    User_ID          integer NOT NULL ,
    District         varchar(56) NULL ,
    Camera_ID        numeric NULL ,
    Energy_benches_ID numeric NULL ,
    Sprinklers_ID    numeric NULL ,
    Garbage_bins_ID  numeric NULL ,
    Sensors_ID       numeric NULL ,

```

```

List_of_workers_ID integer NOT NULL ,
CONSTRAINT XPKPark PRIMARY KEY (Park_ID) ,
    CONSTRAINT R_12 FOREIGN KEY (User_ID) REFERENCES User(User_ID) ,
    CONSTRAINT R_19 FOREIGN KEY (List_of_workers_ID) REFERENCES List_of_workers(List_of_workers_ID)
);
INSERT                                INTO                                Park
(Park_name,Brigade_name,Park_ID,User_ID,District,Camera_ID,Sprinklers_ID,Garbage_bins_ID,Sensors_ID,List_of_
workers_ID)
VALUES('Pushkina','Solomnsky brigada',3,4,5,54,346,123,13,524);

INSERT                                INTO                                Park
(Park_name,Brigade_name,Park_ID,User_ID,District,Camera_ID,Sprinklers_ID,Garbage_bins_ID,Sensors_ID,List_of_
workers_ID)
VALUES('KPI','Shulyavka brigada',36,46,56,546,3466,1236,136,5246);

ALTER TABLE Garbage_bins
ADD CONSTRAINT XPKGGarbage_bins PRIMARY KEY CLUSTERED (Garbage_bins_ID ASC)
go
CREATE TABLE List_of_Workers
(
List_of_Workers_ID integer IDENTITY ( 1,1 ) ,
Park_ID            integer NULL ,
Request_ID        integer NULL ,
Brigade_name      varchar(56) NULL ,
Contacts          char(13) NULL
)
go
ALTER TABLE List_of_Workers
ADD CONSTRAINT XPKList_of_Workers PRIMARY KEY CLUSTERED (List_of_Workers_ID ASC)
go
CREATE TABLE Park
(
Park_ID           integer IDENTITY ( 1,1 ) ,
Park_Name        varchar(56) NULL ,
Adress           varchar(56) NULL
)
go
ALTER TABLE Park
ADD CONSTRAINT XPKPark PRIMARY KEY CLUSTERED (Park_ID ASC)
go
CREATE TABLE Recording
(
Recording_ID      integer IDENTITY ( 1,1 ) ,
Record_Date      datetime NULL ,
Record_Size      numeric NULL
)
go
ALTER TABLE Recording
ADD CONSTRAINT XPKRecording PRIMARY KEY CLUSTERED (Recording_ID ASC)
go
CREATE TABLE Request
(
Request_ID       integer IDENTITY ( 1,1 ) ,
Date_of_application datetime NULL ,
Problem_info     char(18) NULL
)
go
ALTER TABLE Request
ADD CONSTRAINT XPKRequest PRIMARY KEY CLUSTERED (Request_ID ASC)
go
CREATE TABLE Sensor
(
Sensor_ID        integer IDENTITY ( 1,1 ) ,
Type_of_Sensor_ID integer NULL ,
Park_ID          integer NULL ,

```

```

Quantity      numeric NULL
)
go
ALTER TABLE Sensor
ADD CONSTRAINT XPKSensor PRIMARY KEY CLUSTERED (Sensor_ID ASC)
go
CREATE TABLE Sprinklers
(
Sprinklers_ID      integer IDENTITY ( 1,1 ),
Park_ID           integer NULL ,
Quantity          char(18) NULL
)
go
ALTER TABLE Sprinklers
ADD CONSTRAINT XPKSprinklers PRIMARY KEY CLUSTERED (Sprinklers_ID ASC)
go
CREATE TABLE List_of_workers
(
  Brigade_name     varchar(56) NULL ,
List_of_workers_ID integer IDENTITY ( 1,1 ),
District          varchar(56) NULL ,
Type_of_equipment varchar(56) NULL ,
Request_ID        integer NOT NULL ,
CONSTRAINT XPKList_of_workers PRIMARY KEY (List_of_workers_ID),
  CONSTRAINT R_28 FOREIGN KEY (Request_ID) REFERENCES Request(Request_ID)
);
INSERT INTO List_of_workers (Brigade_name , List_of_workers_ID ,District ,Type_of_equipment, Request_ID )
VALUES('Solomnsky brigada',524,'Kievsky','Sensor',88);
INSERT INTO List_of_workers (Park_ID , Quantity ,Characteristic ,Sensors_ID, Type_Of_Sensor_ID )
VALUES('Shulyavka brigada',5246,'Kievsky','Sensor',87);
CREATE TABLE Request
(
  Brigade_name     varchar(56) NULL ,
Request_ID        integer IDENTITY ( 1,1 ),
Date_of_application date NULL ,
Reason           varchar(56) NULL ,
CONSTRAINT XPKRequest PRIMARY KEY (Request_ID)
);
INSERT INTO Request (Brigade_name,Request_ID,Date_of_application,Reason)
VALUES('Solomnsky brigada',88,'2021-12-10','Broke sensores');
INSERT INTO Request (Brigade_name,Request_ID,Date_of_application,Reason)
VALUES('Shulyavka brigada',87,'2021-10-15','Broke cameras');
INSERT INTO Request (Brigade_name,Request_ID,Date_of_application,Reason)
VALUES('Solomnsky brigada',80,'2021-05-25','Sensors shooting down');

```

ДОДАТОК Г

Презентація дипломної роботи

Листів 3

Розробник

Керівник

Ярослав НІЗОВ

Мирослава ГЛАДКА

Київ – 2022

ДИПЛОМНА РОБОТА

“Система управління та обслуговування комунікаціями в парку імені Пушкіна м.Києва”

Керівник кандидат технічних наук,
асистент Гладка Мирослава Вікторівна
Виконав студент групи ІР-41
Нізов Ярослав

Мета роботи: Система управління та обслуговування комунікаціями в парку імені Пушкіна м. Києва.



Поставлені задачі:

- дослідження території проекту;
- розгляд аналогів елементів системи та вибір найкращих;
- відтворення план схеми проекту;
- створення макету бази даних;
- реалізація сервісу моніторингу парку

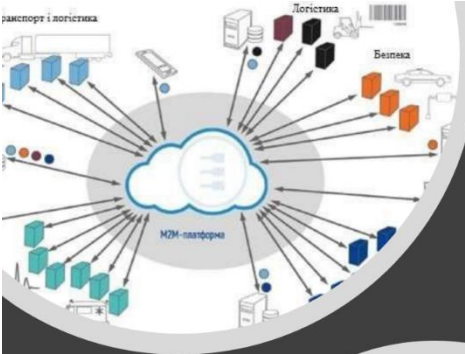
Головні особливості системи

- Синергія роботи елементів між собою.
- Дистанційне керування.
- Автоматизація роботи працівників та парку.
- Можливість оновлювати елементи системи та додавати нові пристрої.
- Використання інтернет технологій.



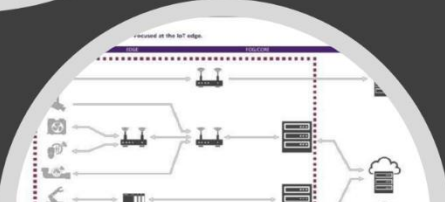

Вже існуючі проекти:

- Американський проект компанії Gemalto
- Розумні лавочки у парку КПІ
- Екостанції розташовані по всіх районах Києва
- Студентський сквер НУБІП з розумними лавочками та ліхтарями



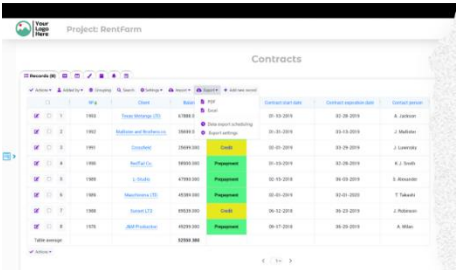
ІоТ-платформи:

- EdgeX Foundry від Linux
- Everyware Cloud від компанії Eurotech

QuintDB

- Внутрішня база даних.
- Можливість налаштування зв'язків між таблицями.
- Зручна модифікація інтерфейсу.
- Додаткові можливості платформи.



ID	№	Тип	Вартість	Статус	Датум початку дії	Датум закінчення дії	Контакт особа
1	1001	Транспорт (Т)	120000	Активний	01.01.2019	31.08.2019	A. Johnson
2	1002	Безпека та захист (Б)	80000	Активний	01.01.2019	31.12.2019	J. Martinez
3	1003	Логістика	200000	Склад	01.01.2019	31.09.2019	J. Lopez
4	1004	Безпека	150000	Припинено	01.01.2019	31.06.2019	K. J. Smith
5	1005	Інфра	170000	Припинено	01.01.2019	30.09.2019	S. Anderson
6	1006	Безпека (Б)	90000	Припинено	01.01.2019	30.01.2019	T. Taylor
7	1007	Системні ІТ	180000	Склад	01.12.2019	30.12.2019	J. Adams
8	1008	ММР-платформа	100000	Припинено	01.01.2019	30.09.2019	A. White

Аналоги елементів системи, що розглядалися:

- Технології, що використовуються для з'єднання: Wi-Fi, Bluetooth.
- Пристрої, які можуть під'єднуватись: смартфони, персональні комп'ютери, планшети та інші.

План схема парку з розташованим обладнанням

