

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерної інженерії

Пояснювальна записка

до кваліфікаційної роботи на ступінь вищої освіти магістр

на тему: **«ПРОЕКТУВАННЯ АРХІТЕКТУРИ ГЕОІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ЛОКАЦІЙ ОБ'ЄКТІВ »**

Виконала: студентка 6 курсу, групи КСДМ–61
спеціальності

123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Урсол Т.С.

(прізвище та ініціали)

Керівник Торошанко Я.І.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2022

ВСТУП

Ми ми живемо в неймовірну епоху щодо того, що відбувається з вторсировиною, коли вона прибуває до місцевого центру вашого міста. Лише в 1970-х роках переробка стала популярною, програми переробки запровадили в містах під час початку зеленого руху, а «програми збору відходів із кількох матеріалів» почали встановлюватися.

Сьогодні система збору та переробки вторсировини стає все більш впорядкованою в міру розвитку технологій та обладнання. Хоча концепція може здатися складною, весь процес можна розбити на 3 основні кроки:

1. Кожне місто та громада дещо відрізняються у способах збору відходів, але більшість міст мають обов'язкові закони про переробку, які вимагають роботи центру збирання відходів та роботи центру. Обов'язково ознайомтеся з правилами та положеннями щодо переробки відходів у вашому місті, щоб знати правила переробки у вашому районі. У містах також існують різні правила щодо того, що можна, а що не можна переробляти. Важливо розуміти, які контейнери не підлягають переробці або які можуть забруднити партію переробки через залишки їжі;
2. Опинившись у центрі, контейнери висипаються на великі машини, і за допомогою працівників вміст сортується за типом, наприклад, газети, пластмаси, банки тощо. Робітники також допомагають відокремлювати чисту вторсировину від забрудненої на різні купи;
3. Вторсировина насправді вважається товаром. Після сортування вторсировини виробники прийдуть до центрів переробки, щоб придбати відсортований вторсировину. Потім вони відвозять їх на переробку, розбивають на сировину і перетворюють на нові продукти. Найбільш затребуваним пластиком, який шукають виробники, є пластик, позначений цифрою 1, 2, 4 або 5. Перш ніж купувати пластикові контейнери, такі як ванна для масла, перед покупкою подивіться на номер, який на ньому позначений.

Споживачі відіграли величезну роль у цьому процесі через підвищення обізнаності про перероблену упаковку. Через це вартість, яку виробник буде платити за перероблені товари, з роками зростає. Виробники, які купують у центрах переробки, також платять за те, щоб центр утилізації працював.

1 Геоінформаційні системи зараз – теорія та імплементація

Як програмний продукт ГІС – це інструмент, який дозволяє користувачам генерувати інтерактивні запити, досліджувати просторові інформації, редагувати інформацію на картах і представляти результати всіх цих операцій. До ГІС відносять ряд різноманітних технологій, процесів і методів, що використовуються в науці, техніці та бізнесі:

1. Координатно-часове прив'язування об'єктів у геодезії, картографії, геології, мореплавстві;
2. Обробка та агрегація фотографічних зображень з космосу для наукових і військових цілей;
3. Обробка геофізико-геодинаміки, застосування в народному господарстві (укладання міських, обласних та державних інвентаризацій територій) та інші.

З цієї причини ГІС є однією з найшвидших зростають сегменти ринку високих комп'ютерних технологій, до яких входить величезна кількість найбільша компанія[1, с.1]

Усе програмне забезпечення ГІС, незалежно від виробника, складається з системи управління базою даних, яка здатна обробляти та інтегрувати два типи даних: просторові та атрибутивні дані.

1. Просторові дані стосуються реальних географічних об'єктів, що представляють інтерес, таких як вулиці, будівлі, озера та країни, а також їх відповідне розташування.
2. Окрім розташування, кожен із цих об'єктів також володіє певними ознаками інтересу або атрибутами, такими як назва, кількість оповідань, глибина чи населення.

Програмне забезпечення ГІС відстежує як просторові, так і атрибутивні дані і дозволяє нам пов'язувати два типи даних разом для створення інформації та полегшення аналізу.

Як апаратне забезпечення, ГІС складається з комп'ютера, пам'яті, запам'ятовуючих пристроїв, сканерів, принтерів, блоків глобальної системи

позиціонування (GPS) та інших фізичних компонентів. Якщо комп'ютер розташований у мережі, мережу також можна вважати невід'ємним компонентом ГІС, оскільки вона дозволяє нам обмінюватися даними та інформацією, які ГІС використовує як вхідні дані та створює як вихідні дані.

Як інструмент, ГІС дозволяє нам зберігати, аналізувати та обмінюватися багатьма даними та інформацією. Від відносно простого завдання складання карти шляху урагану до більш складного завдання визначення найбільш ефективних маршрутів збору сміття в місті, ГІС використовується в державному та приватному секторах. Онлайн- та мобільні послуги картографування, навігації та визначення місця розташування також персоналізують та демократизують ГІС, додаючи карти та картографування до мас[2].

Сьогодні геоінформаційні системи розрізняють:

1. За територіальним охопленням: глобальні, субконтинентальні, національні, регіональні, субрегіональні, місцеві;
2. За предметною областю інформаційного моделювання: урбаністичні (муніципальні), надрокористувацькі, гірничо-геологічні, екологічні (серед них є спеціальна підгрупа, т.к. особливо поширені, отримали земельні інформаційні системи).

Також геоінформаційні системи можна класифікувати за проблемною спрямованістю – вирішені науково-прикладні проблеми. Такими завданнями можуть бути інвентаризація ресурсів, аналіз, оцінка, моніторинг, управління та планування, підтримка прийняття рішень, геомаркетинг.

Вони також розрізняють між багатомасштабними, або незалежними від масштабу геоінформаційними системами, на яких базуються множинні або багатомасштабні уявлення просторових об'єктів [1, с.2].

1.1 Сфери використання геоінформаційних систем

Сфера використання геопросторових даних величезна: вона охоплює кожен сферу чи галузь, де географічне положення має значення. У список входять власне

географія, екологія, туризм, морські науки, сільське господарство, лісове господарство, маркетинг і реклама, військові сили, флот, авіація, правоохоронні органи, матеріально-технічне забезпечення та транспорт, астрономія, демографія, охорона здоров'я, метеорологія та багато інших.

Ось кілька типових прикладів застосування геопросторової технології:

1. Логістика. Відстеження товарів та забезпечення їх якості;
2. Перевезення. Визначення місця та часу прибуття, формування маршруту та навігації;
3. Метеорологія. Посилання прогнозів погоди для окремих територій;
4. Лісове господарство. Виявлення лісових пожеж і вирубки лісів та запобігання масштабним лісовим;
5. Сільське господарство. Оцінка стану рослинності на обраній місцевості;
6. Охорона здоров'я. Моніторинг районів епідемічних спалахів;
7. Екологія. Відстеження популяцій видів на певних територіях, запобігання та усунення лих;
8. Маркетинг і реклама. Націлювання реклами на відповідні регіони;
9. Нерухомість. Візуалізація та аналіз об'єктів нерухомості віддалено;
10. Страхування. Управління ризиками в досліджуваних областях (наприклад, за допомогою аналізу історичних геоприв'язаних даних) [3].

1.2 Підходи до розуміння ГІС

На додаток до визнання багатьох визначень ГІС, також конструктивно визначити три загальні та перекриваються підходи до розуміння ГІС — прикладний підхід, підхід розробника та науковий підхід. Хоча більшість користувачів ГІС, ймовірно, ідентифікують себе з одним підходом більше, ніж з іншим, вони не є взаємовиключними. Більше того, у міру розвитку ГІС та, в більш загальному вигляді, інформаційних технологій наступні категорії будуть відповідно трансформовані та переформовані.

1. Прикладний підхід до ГІС розглядає ГІС насамперед як інструмент. Це також, мабуть, найпоширеніший погляд на ГІС. З цієї точки зору

ГІС використовується для відповідей на запитання, підтримки прийняття рішень, підтримки інвентаризації географічних даних та інформації та, звичайно, для створення карт. Існують певні навички, які необхідно набути та необхідні для правильного використання та застосування ГІС. Прикладний підхід до ГІС більше стосується використання та застосування ГІС для вирішення проблем, ніж сама ГІС;

2. На відміну від попереднього прикладу, в якому ГІС використовується для відповіді або вирішення певного питання, підхід розробника до ГІС стосується розробки ГІС як програмної або технологічної платформи. Замість того, щоб зосередитися на тому, як ГІС використовується та застосовується, підхід розробника спрямований на вдосконалення, доопрацювання та розширення самих інструментів і технологій і в основному належить комп'ютерним програмістам і розробникам програмного забезпечення.
3. Науковий підхід до ГІС не тільки узгоджується з підходами до застосування та розробника, але також більше стосується ширших питань і того, як географія, пізнання, інтерпретація карт та інші геопросторові питання, такі як точність і помилки, мають відношення до ГІС і навпаки. Цей конкретний підхід часто називають географічною інформаційною наукою (GIScience), і він також цікавиться соціальними наслідками та наслідками використання та поширення технології ГІС. Від дослідження поширення помилки до вивчення того, як конфіденційність переозначається за допомогою ГІС та пов'язаних із ними технологій, GIScience одночасно є агентом змін і розуміння[2].

1.3 Веб-карти – термінологія та історія

Веб-картографування — це процес використання карт, що надаються географічними інформаційними системами

(ГІС) [4, с.2].

Існує більше термінів, які іноді плутають з веб-картографуванням і веб-ГІС. Вони включають Інтернет ГІС, GeoWeb (або Геопросторову мережу) та Розподілену ГІС.

Інтернет-ГІС – це термін, який часто використовується як синонім веб-ГІС. Це геопросторовий Інтернет-додаток, який може використовувати більше послуг, ніж лише Інтернет. Він ширший по відношенню до прикладних технологій, але не настільки поширений у використанні, як термін Веб-ГІС.

Веб-картографія використовує Інтернет, найбільшу в світі мультимедійну платформу, для доставки карт і послуг в першу чергу з технологічними питаннями, але також потрібні додаткові дослідження з картографічних теорій, серед інших питань. Це означає, що веб-картографування не тільки відповідає різноманітним Інтернет-протоколам, але й також використовує веб-протоколи. Це також означає, що нові технології та картографічні принципи необхідні для ефективного розробки карт для доставки веб-карт.

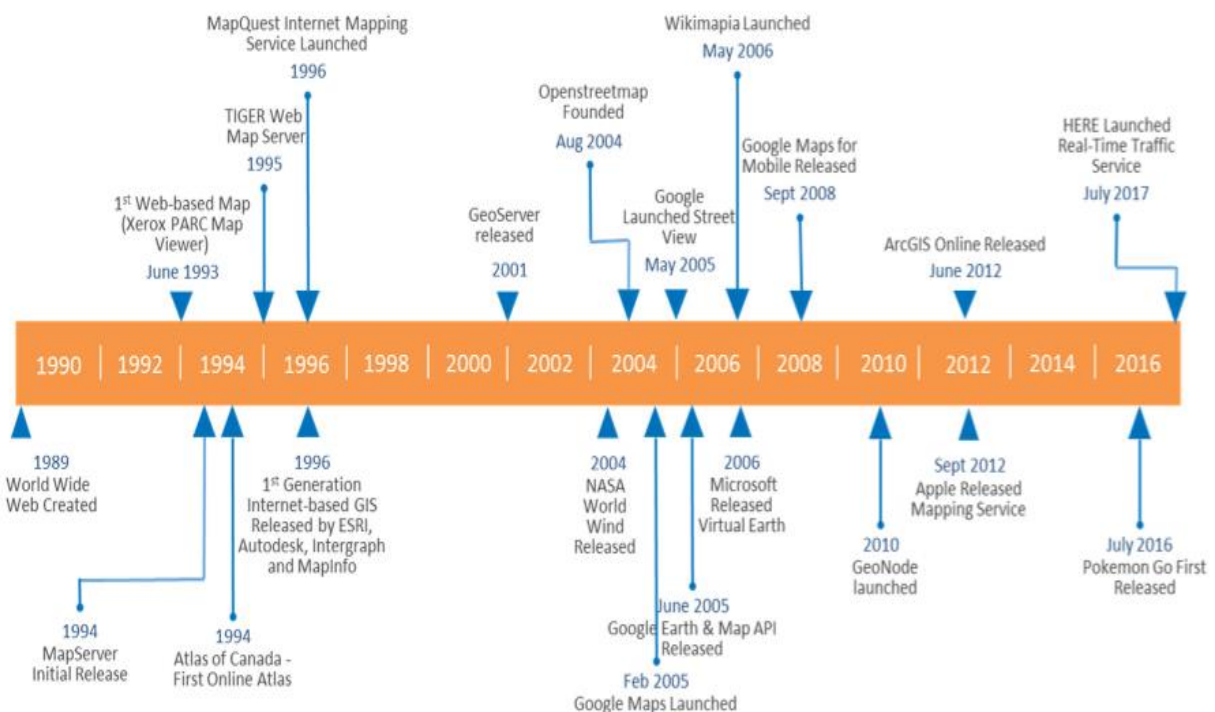
Веб-карти та пов'язаний з ними вміст, представлений у веб-середовищі, тобто у веб-браузерах, вимагають відповідний інтерфейс і часто додаткові функції для запитів і звітів. При проектуванні веб-карти, аудиторія, інтерфейс (наприклад, розмір, простір і колір) і функції швидкої реакції важливі фактори, які слід враховувати. Насправді, Інтернет – це найбільше широко використовуваний Інтернет-технології та веб-ГІС є найпоширенішою формою онлайн-ГІС. Отже ми будемо використовувати термін Веб-ГІС для представлення всіх онлайн-ГІС. Для GeoWeb можна знайти два різних визначення:

1. Злиття геопросторових даних з негеопросторовими (фото, відео, новини тощо);
2. Синонім терміну розподілена ГІС, тобто широко поширене геопросторове середовище, що дозволяє спільно використовувати сумісні компоненти [4, с.4].

Веб-картографування та використання геопросторової інформації в Інтернеті швидко розвивалися останні кілька десятиліть. Майже кожен у світі використовує

картографічну інформацію, незалежно від того, чи усвідомлює це. Майже кожен мобільний телефон тепер має служби геолокації, а також кожному подію та об'єкту земля має місце розташування. Використання геопросторових даних про місцезнаходження швидко розширилося завдяки розвитку Інтернету. Величезні обсяги геопросторових даних доступні і щодня фіксуються в Інтернеті та використовуються у веб-додатках і картах для перегляду, аналізу, моделювання та моделювання. У цій статті розглядається розвиток веб-картографування з першої статичної онлайн-карти зображень до сучасних високоінтерактивних веб-сервісів для картографування з кількох джерел все частіше переходять на платформи хмарних обчислень. Усе середовище веб-картографування фіксує інтеграцію та взаємодію між трьома компонентами, знайденими в Інтернеті, а саме: геопросторова інформація, люди та функціональні можливості.

Малюнок 1.3.1 ілюструє коротку історію основних подій веб-картографування на часовій шкалі.



Розробка веб-картування почалася відразу після створення Інтернету, насамперед у формі опубліковані онлайн-карти, причому PARC Map Viewer є першою веб-картою. Відносно, веб-картографування пройшло короткий шлях з моменту свого

початку, але досягло рівня, якого, можливо, і немає уявлялося два десятиліття тому, пояснюючи, головним чином, досягненнями у веб-технологіях.

Інтенсивний розвиток та зріст популярності практик «живого», інтерактивного веб-картографування триває приблизно з 2003-2004 років. Нині цей напрямок має значні перспективи подальшого розвитку завдяки таким чинникам: а) протягом 2000-х років Інтернет остаточно став глобальним, масово доступним, потужним, поліфункціональним та комфортним засобом комунікації; б) розширився спектр застосування GPS-технологій*, масового розповсюдження набули індивідуальні прилади GPS, дані з яких можна експонувати у веб-просторі в режимі реального часу; в) крім вже існуючих платних (пропрієтарних) систем веб-картографії, виникли нові спеціалізовані онлайн-технології та безкоштовне програмне забезпечення з відкритим кодом, доступніше і простіше у використанні; г) дедалі доступнішою для громадськості стає ексклюзивна геопросторова інформація (аерофотозйомка, супутникові знімки тощо), дані геоінформаційних систем (ГІС)* та можливість користування цими системами (див. скріншоти 1, 2). Вже у 2005 році аналіз перспектив розвитку та можливостей використання комплексу цих технологій спонукав провідних експертів у галузі, таких як А. Тернер та П. Мейер, назвати його «неогеографією» (neogeography) і заговорити про створення «геопросторового вебу» (geospatial web).

Сьогодні даний напрямок викликає жвавий інтерес дослідників, наукових центрів, громадських та міжнародних організацій, урядів, державних служб порятунку і активно розробляється. Найбільш перспективний вектор розвитку сучасної веб-картографії полягає в об'єднанні навколо розв'язання певних питань суспільного життя професійних високотехнологічних систем і мережного краудсорсингу. Інакше кажучи, йдеться про синтез на базі єдиної онлайн-платформи суто науково-дослідницької, професійно обробленої, але здебільшого періодичної інформації і «живих», приватних, недостатньо перевірених відомостей, які зате надаються в режимі реального часу, характеризуючи ситуацію «тут і зараз». Це оптимальний симбіоз для раннього попередження НС і швидкого реагування на них, і він вимагає певної моделі інституційної взаємодії, базованої на: а) принципах мережності,

волонтерства, краудсорсинговості і відкритості; б) тісній взаємовигідній колаборації з міжнародними організаціями, державними органами, науковими і приватними структурами[17].

Початок розвитку як правило пов'язують з тим, коли Google купили організацію Keyhole Inc і 2004-го року було запущено веб-карти Google Earth, а після цього і сервіс Google Maps. Вже після було запущено сервіс Virtual Earth від Microsoft, зараз вже відомий як Bing Maps. Повільний, але впевнений зріст публічності супутникових знімків та даних онлайн-ГІС в цілому почався саме с Google Earth та Google Maps. У 2007 року Гарвардський Університет в рамках Harvard Humanitarian Initiative почав дослідницьку програму «Crisis Mapping and Early Warning», що мала за ціль дослідження потенціалу публічного краудсорсингу даних для веб-карт у цілях моніторингу надзвичайних ситуацій та а також ліквідації наслідків цих ситуацій.

На зараз у Інтернеті є тисячі онлайн-ГІС – веб-сервісів, що дають доступ до картографічну інформацію через звичайний браузер, або самостійні програми- «оболонки». Більшість із них надають супутникові знімки й різні карти, а також усіяку просторово прив'язану інформацію (наприклад, визначні пам'ятки, автомобільні пробки й т.п.). Можна окремо виділити веб-орієнтовані картографічні сервіси (web-based maps), які являють собою веб додаток і працюють у браузері (OpenStreetMaps, Wikimapia) і сервіси, які мають власну оболонку-навігатор (virtual globus), але при цьому картографічні дані, використовувані в них, завантажуються з мережі Інтернет (Bing Maps)[17].

За ініціативи Open Geospatial Consortium було створено та застосовано стандарти інтерфейсів онлайн-ГІС і специфікацій кодування гео-даних (WMS, WCS, WFS), що є метриками функціональності веб-карт. Сьогодні багато великих виробників програмного забезпечення для ГІС (ESRI, MapInfo, Autodesk, Intergraph) вже надають доступ до власних картографічних серверів. Також є чимало програмного забезпечення з відкритим кодом, що мають більший функціонал, підтримкою великої кількості форматів вхідних даних (GRASS, gvSIG, GDAL, Mapserver,

OpenLayers, GIS-Lab.info) і, звичайно, має нескінченні можливості для масштабування.

За допомогою веб-сервісу Google My Maps користувачі отримують можливість створювати власні слої на основі карт Google Maps, зберігати масиви локацій та інформації про них, а також ділитися ними з іншими користувачами картографічних сервісів Google Maps і Google Earth. Можна додавати об'єкти інфраструктури, створювати маршрути на основі доданих локацій. На даний момент Google Map Maker доступний більш ніж в 190 країнах і ця кількість росте. В архітектурі системи передбачена верифікація даних, отриманих від вільних користувачів, проте великим недоліком є те, що дані мають бути перевірені сертифікованими працівниками Google до публікації.

Одним з відомих представників краудсорсингової веб-картографії є OpenStreetMap, або OSM – це некомерційний онлайн-ГІМ проект, що підтримується силами тисяч волонтерів-користувачів, що забезпечують оновлення та розширення бази даних цього сервісу з різних куточків світу. Як джерела даних також використовуються записи з персональних GPS-трекерів, знімки аерофотографії, відеозаписи, супутникові знімки і панорами вулиць, до яких надають доступ певні організації. В OpenStreetMap для створення карт використовується принцип Вікіпедії, тобто кожний зареєстрований користувач може додавати свої дані та коригувати вже існуючі.

В 2012 році було опубліковане дослідження відповідно до якого в Німеччині OSM має обсяг інформації більший за сервіс TomTom/TeleAtlas на 27 %. Часто зазначають що, що оскільки для OSM першочерговим джерелом на є точні координати GPS-треків, а не супутникові знімки, що часто неточно прив'язані до геолокацій, їхні карти нерідко є значно коректнішими за аерокосмічні[17]. Станом на травень 2012 року кількість учасників спільноти OSM досягла 500 000.

У січні 2010 р., після початку землетрусу на Гаїті активні користувачі OSM створили цілий набір спеціалізованих кризових веб-карт ресурсів, що включали неймовірно точну та детальну веб- карту району надзвичайної ситуації, де було позначено майже всі зруйновані будинки, вузли комунікації, джерела води,

тимчасові табори біженців тощо. Після цього популярність OpenStreetMap очікувано зросла – завдяки ним був уведений у широке використання термін «crisis mapping», для OSM надали доступ до своїх супутникових знімків таким організаціям як NOAA, GeoEye, DigitalGlobe, Eros, CNES/Spot Image, JAXA/ALOS, Google, World Bank. Користувачі OSM також допомогли рятувальним органам під час лісових пожеж у Російській Федерації 2010 року та землетрусам і цунамі в Японії.

Практично всі найбільші геопортали на різних умовах надають власну інформацію та/чи програмне забезпечення криз-картографічним ресурсам, а час від часу беруть безпосередню участь у ліквідації наслідків стихійних лих чи гуманітарних катастроф (наприклад, участь проекту OpenStreetMaps в ліквідації наслідків НС на Гаїті, в Японії та РФ).

Щодо власне кризової веб-картографії, то крім глобальних дослідницьких проектів та ініціатив постійно виникають і розвиваються локальні платформи, наприклад, Sudan Satellite Sentinel Project (<http://www.satsentinel.org/>), або новозеландська CERA (докладно – нижче), а також спеціалізовані з глобальним охопленням – Citivox (<http://citivox.com/>) або Development Seeds (<http://developmentseed.org/>), орієнтовані на створення й підтримку кризових онлайн-карт на замовлення, Tomnod (<http://tomnod.com/>), російський Космоснимки (<http://kosmosnimki.ru/>) тощо. Таких проектів десятки, і їх спільною рисою є те, що отримані в рамках проекту дані візуалізуються в комплексі з аерокосмічними геопросторовими даними, на т.зв. «супутникових картах».

В контексті таких тенденцій розвитку «чисті» криз-краудсорсингові платформи в їх сучасному вигляді навряд чи можуть розглядатися як перспективна стратегічна модель розвитку криз-картографії через свою концептуальну, а, відтак, технологічну «неповноту». Наприклад, в класичному Ушахіді – одному з найуспішніших на сьогодні проектів кризового краудсорсингу, що практично став його символом – при всіх його перевагах для онлайн-візуалізації даних передбачена звичайна топографічна карта з мінімумом геопросторової інформації. Не будучи достатньо пристосованим для залучення/відображення високотехнологічної

верифікованої інформації (ГІС, GPS, аерокосмічні візуальні дані тощо), Ушахиді репрезентує скоріше певну проміжкову фазу розвитку веб-картографії.

Водночас, вже існують переконливі спроби синтезу. Такою, наприклад, є веб-сторінка новозеландської The Canterbury Earthquake Recovery Authority, офіційної урядової «агенції, що очолює та координує відновлювальні роботи після руйнівних землетрусів вересня 2010 та лютого 2011 року» у регіоні Кентербері і його центральному місті Крайстчерч. У складі всесвітньої мережної організації Crisis Mappers нині працює велика (більш ніж 1 600 членів) волонтерська група, що займається питаннями синтезу професійної та краудсорсингової інформації на єдиних платформах і зокрема розробкою єдиних стандартів та рекомендацій для таких веб-ресурсів[17].

1.4 Сьогодення ГІС

Основною ідеєю і «надзавданням» веб-картографування на даному етапі його розвитку є створення глобальної, відкритої, інтерактивної, багатоцільової онлайн-інфраструктури, що працює в «режимі реального часу» (дані завжди є свіжими) і базована на тісній колаборації урядів, науково-експертних кіл, а також – найголовніше – пересічних людей по всій земній кулі, об'єднаних (чи не об'єднаних) в ті чи інші онлайн-спільноти, недержавні організації, рухи тощо. Центральний принцип архітектури даної системи – динамічна, синхронізована в часі і просторі, комплексна, багаторівнева візуалізація всього масиву даних, отриманих з різних джерел*, на платформі спеціальних картографічних веб-ресурсів.

На даний момент світова система (інфраструктура) масової веб-картографії знаходиться на початковому етапі свого формування і має децентралізовану, багатошарову архітектуру, в рамках якої поєднуються як глобальні (Google Maps, Open Street Map, Bing Maps), так і локальні (Sudan Satellite Sentinel Project, CERA, Яндекс Карты і Яндекс Народная карта) сервіси широкого та власне кризового (Development Seeds, Citivox, Tomnod) призначення, що можуть бути

пропріетарними і некомерційними, дослідницькі центри (програма «Crisis Mapping and Early Warning» в рамках Гарвардської Гуманітарної Ініціативи), різноманітні онлайн-спільноти, блоги, форуми (Green Map System, iRevolution – блог Патріка Мейера і багато інших).

При цьому вже з'явилися мережні організації, що відіграють роль глобальних координаційних центрів і дискусійних майданчиків, – в сегменті кризової веб-картографії такою, наприклад, є Crisis Mappers (повна назва –The International Network of Crisis Mappers), що на сьогодні об'єднує більш ніж 1 700 організацій і близько 2 000 веб-к/артографів у 157 країнах. Під егідою організації з 2009 року проходять щорічні Міжнародні конференції кризової картографії (International Conference of Crisis Mappers - ICCM) [17]

Одна з основних технологічних проблем розвитку сучасної веб-картографії полягає в тому, що на сьогодні відсутні надійні відпрацьовані технології верифікації краудсорсингових даних. Це дає незначний відсоток помилок і неточностей, але досі невідомо про випадки навмисного стороннього викривлення даних чи цілеспрямованої масштабної дезінформації в рамках означених проектів. Однією з основних проблем інституційного розвитку сучасної веб-картографії (включаючи кризову) є дефіцит системного, взаємовигідного і достатньо легітимізованого співробітництва між волонтерським, некомерційним її сектором і урядовими та іншими офіційними структурами відповідного профілю. Водночас реалізуються численні спільні проекти, відкриті веб-картографічні ресурси open source широко використовуються міжнародними, урядовими та приватними структурами (OpenStreetMap), поряд з державними чи пропріетарними професійними центрами геоінформаційного аналізу (ГІС) виникають альтернативні некомерційні онлайн-сервіси, що серед інших послуг пропонують вільне ПЗ (GRASS, gvSIG, GDAL, Mapserver, OpenLayers, GIS-Lab.info). Такі тенденції свідчать про обопільну зацікавленість і дозволяють твердити, що подальший взаємовигідний симбіоз глобальної веб-картографії та офіційних структур, її подальша інституалізація, швидке зростання і структурне укрупнення стане другим етапом і найімовірнішим сценарієм її розвитку [17].

Однією з центральних проблем розвитку сучасної веб-картографії, що впливає з самої її концепції та архітектури, є верифікація краудсорсингових даних, тобто перевірка інформації, що надходить не з офіційних чи інших надійних джерел, а від пересічних користувачів (безпосередньо з їхніх телефонних номерів чи через інтерактивні веб-сервіси). Такі щоразу унікальні відомості, надані практично без затримки у часі й зазвичай прямо з місця подій, є одним з ключових ресурсів веб-картографії, її наріжним каменем, без якого вона багато в чому втрачає сенс. Але зрозуміло, що саме ці дані (до того ж, нерідко анонімні) найскладніше перевірити, хоча насамперед вони можуть містити найбільше дезінформації, помилок, неточностей тощо.

У той же час, на сьогодні відсутні надійні технології верифікації краудсорсингових даних. П. Мейер, одних з провідних експертів у галузі, після детального вивчення питання дійшов висновку, що найоптимальнішим шляхом вирішення питання на даний момент є «збалансоване застосування декількох стратегій одночасно», яке «подеколи є потужним засобом перевірки краудсорсингової інформації з мінімальними затримками у часі».

Тривають процеси конвергенції між мережними веб-картографічними сервісами open source і урядовими та приватними структурами. Наприклад, відкритий сервіс OpenStreetMap офіційно використовують такі структури як Організація Об'єднаних Націй, сайт президента США, Федеральне космічне агентство Російської Федерації, Microsoft Bing, Оксфордський університет, французька газета «Liberation», Flickr, MapQuest, Вікіпедія, Вікімапія, американські рятувальники, Космознімки (kosmosnimki.ru) і багато інших[17].

Водночас, експерти визнають, що у глобальному вимірі взаємодія ця поки що не має системного, регулярного і достатньо легітимізованого характеру, а відтак – лишається відкритим питанням. У лютому поточного року Інститут досліджень національної стратегії при Національному університеті оборони США представив об'ємне дослідження-довідь «Конструктивне зближення: картографія і гуманітарна допомога». Зокрема, у студії робиться такий висновок: «Урядові структури різних рівнів починають розуміти цінність кризової онлайн-картографії,

що надає інформацію практично в режимі реального часу. Статичні веб-сторінки більше не задовольняють потреб ані виборців, ані рятувальних органів. Але нові динамічні джерела інформування, поряд з цим, порушують звичні механізми планування і реагування.

У першу чергу вищесказане стосується кризової картографії, оскільки сфера її застосування є найбільш критичною та актуальною як для урядів, так і для громадськості. Сьогодні кризовий сегмент світової веб-картографії є найрозвинутішим і акумулює в собі найсучасніші технологічні рішення. На даний момент у світі йде інтенсивна робота над елементами нової моделі цивільного захисту, де акцент робиться на ранньому попередженні НС та миттєвому реагуванні на них на основі перманентного багатоканального моніторингу і картографічній візуалізації даних онлайн. Вже зараз ця модель широко застосовується в умовах окремих НС та гуманітарних катастрофах, що, по суті, є процесом її доопрацювання та апробації. Враховуючи в позитивні результати даних випробувань, а також сталий та інтенсивний інтерес до кризової веб-картографії з боку міжнародних організацій, урядів та авторитетних науково-дослідницьких кіл – у середньостроковій перспективі триватимуть процеси вдосконалення цієї моделі та її імплементації (безумовно, з поправкою на місцеві особливості) в якості важливого ресурсу урядових та міжнародних систем цивільного захисту в різних країнах світу.

Модернізація систем цивільного захисту за описаними вище напрямками є сьогодні одним з прогресивних і перспективних глобальних трендів, включення в який нині є стратегічно доцільним для будь-якої країни світу. Але для України – з точки зору готовності суспільства та вітчизняних державних органів – приєднання до цього процесу буде додатково ускладнюватись наступними чинниками:

- сьогодні практично немає суто українських онлайн-спільнот з питань веб-картографії та кризового краудсорсингу;
- участь українських користувачів в глобальних проектах такого роду є мінімальною, відповідно, відсутні нормативні механізми потенційного

співробітництва між державою і недержавними профільними організаціями та проектами.

З іншого боку, Україна має достатній ступінь технологічної готовності до розгортання повноцінних масових веб-картографічних ресурсів, зокрема: а) порівняно високий рівень і добрі перспективи проникнення Інтернет (включаючи мобільний, що є основною апаратною платформою для масової веб-картографії); б) власні науково-технологічні ресурси для розробки і розвитку ГІС; в) наявність власного орбітального космічного апарата «Січ-2», однією зі штатних функцій якого є моніторинг надзвичайних ситуацій. Існує також багатий світовий досвід, широкий спектр відкритого програмного забезпечення, міжнародні онлайн-мережі та некомерційні організації, відкриті до співробітництва на різних рівнях.

1.4 Сучасні системи транспортування даних

Для вирішення проблеми транспорту даних корпоративних обсягів з кількох джерел були розроблені системи (стандарти), що визначають послідовності дій роботи з масивами даних для оптимізації процесів їх періодичного транспорту, обробки та оновлення звітів. Базовими системами передачі даних є ETL та ELT системи.

- ETL (Extract, transform, and load) - система передачі даних, яка використовується для збору даних з різних джерел, обробки їх відповідно до вимог корпорації та завантаження їх до кінцевого сховища даних. Обробка даних, зазвичай включає різні операції, такі як фільтрування, сортування, агрегування, поєднання даних, очищення даних, видалення дублікатів та перевірка даних. Часто три фази ETL виконуються паралельно, щоб заощадити час. Наприклад, під час вилучення даних процес трансформації може працювати над вже отриманими даними та готувати їх до завантаження, а процес завантаження може почати

працювати над підготовленими даними, а не чекати завершення усього процесу вилучення [9, с.1058]. У першій фазі дані відправляються до місця постановки, певного вузлу мережі. Попередні обробки проводяться в області постановки так, що продуктивність вихідної системи не погіршується. Крім того, якщо пошкоджені дані будуть скопійовані безпосередньо з джерела до сховища даних, відновлення буде проблемою. Область постановки дає можливість перевірити отримані дані до того, як вони перейдуть у сховище даних. Область постановки також займається інтеграцією систем, що мають різні СУБД, апаратні ресурси, операційні системи та протоколи зв'язку;

- ELT (Extract, load, and transform) - відрізняється від ETL виключно тим, де відбувається обробка. У ELT перетворення відбувається в кінцевому сховищі даних. Замість того, щоб використовувати окремі ресурси обробки, для трансформування даних використовуються можливості обробки кінцевого сховища даних. Це спрощує архітектуру, видаляючи двигун трансформації з архітектури даної системи. Ще одна перевага цього підходу полягає в тому, що масштабування кінцевого сховища даних також збільшує продуктивність ELT. Однак ELT працює добре лише тоді, коли система є достатньо потужною для ефективного перетворення даних. ELT є оптимальним вибором, коли швидкість передачі даних є важливішою за детальність її перевірки та за відсутності суттєвої різниці між середовищем джерела даних та кінцевого сховища.

Перший крок отримання даних є спільним для обох систем. Існує три види отримання даних:

- Full Extraction (транспорт усього об'єму даних одним пакетом, характерно для ELT);
- Partial Extraction- without update notification (транспорт даних при попередньому розділі на пакети);
- Partial Extraction- with update notification (транспорт даних при попередньому розділі на пакети для великих обсягів).

Фазу завантаження даних до сховища також можна розділити на три типи:

- Initial Load — перезавантаження усіх таблиць сховища;
- Incremental Load — оновлення сховища за умови змін, перевірка на зміни відбувається періодично;
- Full Refresh — повне перезавантаження визначених таблиць сховища.

Існує безліч інструментів, які автоматизують процес організації ETL. Серед них вирізнити три основних типи:

- інструменти зі зберігання даних, що беруть на себе весь процес зберігання даних, надаючи область постановки, сховище та інструменти для обробки та інтеграції даних, серед них - MarkLogic, Oracle data integrator, Amazon RedShift;
- ETL-креатори, що представлені як програмні додатки або високорівневі бібліотеки мов програмування, вони забезпечують лише організацію процесу завантаження та обробки даних, областю постановки може бути окремий виділений вузол або оперативна пам'ять. Серед них - Airflow, Spark, Luigi, pandas;
- хмарні ETL-інструменти, що пропонують інструменти з'єднання зі сховищем та трансформації, а такою хмарне сховище для області постановки. Серед них - Informatica–PowerCenter, IRI Voracity, HEVO.

1.5 Інформаційні системи

Інформаційні системи є прикладною реалізацією автоматизації звичних бізнес-процесів, з плином часу ці системи набули додаткових можливостей та характеристик [10, с.17].

- 1) TPS (Transaction Process System) - перша інформаційна система, створена для збору, зберігання та модифікацій даних про бізнес-процеси до бази даних. Існує багато різних типів систем обробки транзакцій, таких як нарахування заробітної плати, контроль ресурсів, облік замовлень, кредиторська заборгованість, дебіторська заборгованість та інші. Обробка транзакцій дає цінний внесок у багато інших систем організації, таких як інформаційні системи управління та системи підтримки прийняття рішень. TPS служить основою для цих інших систем. TPS відстежує звичайні операції, але не надає особливої підтримки для прийняття рішень. Є два способи обробки транзакцій: за допомогою пакетів та в режимі реального часу. У системі пакетної обробки операції накопичуються протягом певного періоду часу і обробляються як одиниця, або партія. У системі обробки в реальному часі транзакції обробляються негайно, оскільки вони відбуваються без будь-якої затримки для накопичення транзакцій. Обробка в режимі реального часу також називається обробкою транзакцій в Інтернеті або OLTP;
- 2) MIS(Management information system) - використовується для прийняття рішень згідно висновків аналітичних звітів, що зроблені на основі збережених даних, тобто на базі системи TPS. Три компоненти MIS забезпечують більш повне та цілеспрямоване визначення, де система пропонує інтеграцію та цілісний вигляд, а також наявність вводу, виводу, процесів обробки даних та зворотного зв'язку, інформація означає оброблені дані подані у вигляді звіту для допомоги у плануванні, а менеджмент - кінцевий користувач, який приймає рішення, тобто усі представники управління організації. Головними можливостями MIS є:

- a) Отримання даних;
 - b) Обробка даних;
 - c) Зберігання результатів;
 - d) Пошук результатів;
 - e) Поширення результатів.
- 3) DSS(Decision support system) - створена для автоматизації прийняття стандартних організаційних рішень та планування. DSS використовує підсумкову інформацію, винятки, закономірності та тенденції, використовуючи аналітичні моделі. Система підтримки прийняття рішень допомагає у прийнятті рішень, але не обов'язково дає саме рішення. Ті, хто приймає рішення, збирають корисну інформацію з необроблених даних, документів, особистих знань та/або бізнес-моделей для виявлення та вирішення проблем та прийняття рішень. DSS має три основних атрибути [11]:
- a) СУБД - для вирішення проблеми необхідні дані можуть надходити з внутрішньої або зовнішньої бази даних. В організації внутрішні дані генеруються системою, такою як TPS чи MIS. Зовнішні дані надходять з різних джерел, таких як газети, інтернет-сервіси даних, бази даних (фінансові, маркетингові, людські ресурси);
 - b) система управління моделями - зберігає та отримує доступ до моделей, які менеджери використовують для прийняття рішень. Такі моделі використовуються для проектування виробничих потужностей, аналізу фінансового стану організації, прогнозування попиту на товар чи послугу тощо;
 - c) допоміжні інструменти - онлайн-допомога; випадające меню, інтерфейси користувача, графічний аналіз, механізм виправлення помилок, що полегшує взаємодію користувачів із системою.
- 4) EIS(Executive information system) - система, що відрізняється поєднанням внутрішньої та зовнішньої інформації та прийняттям рішення на основі його аналітики. Призначені для використання

старшими керівниками безпосередньо для підтримки підтримки непрограмованих рішень у стратегічному управлінні. Інформація, що ці системи надають, часто є зовнішньою, неструктурованою і навіть невизначеною. Точний обсяг та контекст такої інформації часто заздалегідь невідомий;

- 5) OLAP(Online analytical processing) - новий тип інформаційних систем, метою створення якого стала оптимальна організація обробки багаторозмірних запитів даних. Бази даних, які бізнес використовує для зберігання всіх своїх транзакцій та записів, називаються OLTP. Ці бази даних зазвичай мають записи, які вводяться по одному. Часто вони містять інформацію, яка є цінною для організації. Бази даних, які використовуються для OLTP, однак не були розроблені для аналізу. Тому отримання відповідей з цих баз даних є витратним у плані часу та зусиль;

Системи OLAP були розроблені, щоб допомогти витягнути важливу інформацію з даних з високою продуктивністю. Це пояснюється тим, що бази даних OLTP оптимізовані для великого завантаження. Одним з найвідоміших прикладів систем OLAP є Microsoft Azure Analysis Services [9, с.1062];

- 6) Data mining - новий підхід до інформаційних систем, що пропонує отримання цінних аналітичних висновків та нової інформації з даних, які раніше не були залучені до аналітичних процесів та не впливали на прийняття рішень, проте зберігалися. Системи data mining продуктивно використовуються у наступних сферах [13]:

- a) маркетингові дослідження;
- b) управління ризиками;
- c) детекція фроду (дій шахраїв).

Реалізація систем data mining може бути здійснена через процедури на основі СУБД, машинне навчання систем штучного інтелекту, побудову

візуалізацій (інфографік, графіків, таблиць), статистичних розрахунках, дій менеджерів у організованій інформаційній системі та спеціалізованих методів для певних областей діяльності;

7) BI(Business intelligence) - інформаційна система, що включає методи та інструменти для обробки інформації, візуалізації, створення готових звітів та використання штучного інтелекту для аналітики. Термін «Business Intelligence» розвинувся із систем DSS і зміцнився завдяки такій технології та застосуванню, як OLAP. Система бізнес-аналітики - це в основному система, яка використовується для пошуку залежностей з існуючих даних. BI відрізняють дуже великим переліком можливостей, що надають їх реалізації [11]:

а) зберігання даних та управління:

- сховище даних;
- спеціалізовані звіти для кожного випадку;
- контроль якості даних;
- Data mining.

б) поширення даних:

- звітування;
- колаборація;
- управління звітами;
- інфографіки;
- розрахунки метрик.

с) запити, звіти та аналіз

- спеціалізовані звіти;
- звітування продуктивності.

1.6 Бази даних

База даних може розглядатися як своєрідна електронна картотека; вона містить оцифровану інформацію ("дані"), яка зберігається в постійному зберіганні якогось типу. Користувачі можуть вставляти нову інформацію в базу даних і видаляти, змінювати або отримувати наявну інформацію в базі даних, видаючи запити або команди програмному забезпеченню, яке управляє базою даних, тобто системою управління базами даних (коротко СУБД) [11]. СУБД - програмне забезпечення, яке керує базою даних - ефективно служить своєрідним посередником між логічним та фізичним рівнями системи: запити користувачів на доступ до бази даних виражаються через логічну базу даних та реалізуються СУБД. Однією із загальних функцій, що надаються СУБД, є захист користувачів від деталей фізичного рівня системи (так само, як системи мови програмування також захищають користувачів від деталей фізичного рівня системи). Головним компонентом, що відрізняє СУБД між собою, точніше - створює ряди відмінностей - є оптимізатор, що відповідає за шлях, який проходить збір даних з пам'яті для користувацького запиту.

Іншими важливими функціями СУБД є контролю:

- контроль безпеки необхідні для забезпечення законності запитів користувачів, у тому сенсі, що користувач запитує операцію, до якої йому дозволено здійснювати дані, до яких йому дозволено отримувати доступ. У кожній СУБД є свої особливості процедур створення акаунтів користувачів та розподілу доступів;
- контроль паралельності пов'язаний з можливістю того, що база даних одночасно може використовувати декілька користувачів;
- контроль цілісності пов'язані із гарантуванням правильності даних у базі даних (наскільки можна надати будь-яку таку гарантію);
- контроль відновлення займається підтримкою резервного копіювання даних та процедурами відновлення втрачених даних з копій.

Основною будівельною одиницею бази даних є таблиця. Таблиця складається з полей, що визначають тип даних, призначення та обмеження колонки (може бути незаповненою, автоматичне заповнюється значення, що залежне від попереднього) та записів, що складаються з рядка, який включає один екземпляр значення для кожної колонки. Таблиця є логічним представленням даних у базі. Фізичні представлення даних у базі сильно відрізняються у різних СУБД та мають різні переваги. Ще однією важливою логічною одиницею баз даних є відносини даних між собою, що здійснюються за допомогою ключів - однакових колонок індексів у різних таблицях. За типами відносин та фізичного представлення даних СУБД поділено на різні типи:

- реляційні. Організують бази, побудовані на рівноправних зв'язків між таблицями на основі ключів - спільних колонок. Оновлення реляційних баз даних відбувається з чітко визначеною періодичністю, а продуктивність залежить від побудови запитів до баз даних. Приклад - PostgreSQL;
- NoSQL(Not only SQL) - побудовані на особливих типах зв'язків між таблицями, NoSQL СУБД були розроблені через потребу у зберіганні великих обсягів даних за умов обмеженості фізичних ресурсів та частих оновленнях, через це головний вплив на їхню продуктивність спричиняє апаратне забезпечення серверів. NoSQL стало збірною назвою усіх СУБД, що надають альтернативний до реляційного підхід до організації баз. Існують наступні види NoSQL СУБД:
- стовпчикові. Оптимізований для читання та запису стовпців даних на відміну від рядків даних. Орієнтоване на стовпчик сховище для таблиць баз даних - це спроба зменшити витрати на введення / виводу для бази даних. Оскільки витратність вводу / виводу знижена, загальний об'єм для зберігання знижується. Однією з головних особливостей стовпчикових СУБД є їх здатність компресувати дані. Замість того, щоб дані записувались у традиційній орієнтації рядків, у цих базах даних використовується орієнтація стовпців. Кожен стовпець буде пов'язаний з ключем стовпця. Прикладами таких СУБД HBase, Vertica;

- документі. Дизайн даних у базах даних документів набуває напівструктурованої форми, як у форматах JSON або XML. Схема для бази даних документів гнучка, що дає розробнику інструменти для масштабування програм, підтримуваних базою даних. Вимоги до дизайну та вільність у вимогах до схеми дозволяють забезпечити високу пропускну здатність в документних СУБД. Приклад - MongoDB;
- графові. Ці СУБД фокусуються на тому, як дані стосуються інших точок даних. Вони зберігають спрямований зв'язок між наборами даних, які називаються ребрами. Ці ребра відображаються у вигляді графу. Графові бази даних використовуються у випадках, коли дані не можна віднести до реляційної структури, або ієрархії, проте зв'язки існують між усіма полями - як у соцмережах, онлайн-кінотеатрах, де зв'язком між профілями користувачів може виступати як інший користувач, так і спільна музика, спільноти тощо. Приклад - Neo4j;
- ключ-орієнтовані. Використовують у середовищах з високими витратами на операції зчитування та з високими потребами у швидкості, наприклад у веб-додатках. Ці СУБД створюють об'єктно-орієнтовані бази з реляційними зв'язками та зберігають дані на SSD-накопичувачах або у операційній пам'яті. Як правило, дані СУБД використовують не для зберігання повного обсягу даних, а для зберігання результату великого запиту отриманого з сервера, проте на стороні клієнта. Приклад - Redis.

За типами організації бази даних можна розділити на такі категорії:

- централізовані. Інформація (дані) зберігається в централізованому місці, і користувачі з різних місць можуть отримати доступ до цих даних. Цей тип баз даних містить процедури застосування, які допомагають користувачам отримувати доступ до даних навіть із віддаленого місця. Для перевірки та підтвердження кінцевих користувачів застосовуються різні види аутентифікації, також реєстраційний номер надається процедурами програми, яка веде облік даних про використання даних (логування);

- розподілені. На відміну від концепції централізованої бази даних, розподілена база даних має внески як із загальної бази даних, так і з інформації, що зберігається місцевими комп'ютерами. Дані не в одному місці і поширюються на різних сайтах організації. Ці сайти з'єднані один з одним за допомогою комунікаційних посилань, що допомагає їм легко отримувати доступ до розподілених даних. Можна уявити розподілену базу даних як ту, в якій різні частини бази даних зберігаються в декількох різних місцях (фізичних), поряд з процедурами застосування, які реплікуються і розподіляються між різними точками мережі;
- персональні. Дані збираються та зберігаються на персональних комп'ютерах, що є невеликим і легко керується. Дані, як правило, використовуються одним і тим же відділом організації та мають доступ до невеликої групи людей;
- кінцевого користувача. Кінцевий користувач, як правило, не переймається транзакцією або операціями, які проводяться на різних рівнях, і знає лише той продукт, який може бути програмним забезпеченням або додатком. Тому це спільна база даних, спеціально розроблена для кінцевого користувача, як і менеджери різних рівнів. У цій базі даних зібрано загальну інформацію про всю інформацію;
- комерційні. Це платні версії величезних баз даних, розроблені унікально для користувачів, які хочуть отримати інформацію про допомогу. Ці бази даних є предметом конкретного характеру, і не можна дозволити собі підтримувати таку величезну інформацію. Доступ до таких баз даних забезпечується через комерційні посилання;
- NoSQL. Вони використовуються для великих наборів розподілених даних;
- операційні. Інформація, що стосується операцій підприємства, зберігається всередині цієї бази даних. Такі бази даних, як маркетинг, відносини з працівниками, обслуговування клієнтів тощо, вимагають такого роду баз даних;
- реляційні;

- хмарні. Зараз у день дані спеціально зберігаються над хмарами, відомими також як віртуальне середовище, або у гібридній хмарі, у публічній чи приватній хмарі. Хмарна база даних - це база даних, оптимізована або побудована для такого віртуалізованого середовища. Існують різні переваги хмарної бази даних, деякі з яких - це можливість платити за пропускну здатність і пропускну спроможність на основі кожного користувача, і вони забезпечують масштабованість за потребою, а також високу доступність;
- об'єктно-орієнтовані. Об'єктно-орієнтована база даних - це сукупність об'єктно-орієнтованого програмування та реляційної бази даних. Існують різні елементи, створені за допомогою об'єктно-орієнтованих мов програмування, такі як C ++, Java, які можна зберігати у реляційних базах даних, але об'єктно-орієнтовані бази даних добре підходять для цих елементів. Об'єктно-орієнтована база даних організована навколо об'єктів, а не дій, даних, а не логіки.

Запити до бази даних формуються за принципами запитів до системи мовами програмування, тому що база даних, власне, є одним з видів системи програмування. Мови запитів так само включають інструкції, змінні, типи, оголошення, імена, значення, оператори зчитування та порівняння. Різні діалекти мов запитів пропонують певні обмеження та особливості у використанні запитів, як-то можливість створювати тимчасові змінні, оголошувати процедури, використовувати різнорівневі агрегації підзапитів. Більшість обмежень, що надають діалекти мов запитів, спрямовані на згаданий вище захист користувачів від деталей фізичного рівня та, відповідно, гарантію безпечності результатів запитів та їх впливу на стан бази даних - наприклад, у MS SQL є можливість створити нескінченний запит при з'єднанні таблиць за ключем однієї колонки з обох боків, що може призвести до перевантаження бази або сервера, якщо мова йде про поєднання таблиць з різних баз. Подібні винятки є сферою відповідальності не тільки розробників СУБД та діалекту і адміністратора сервера, або іншого автора запиту, що має для цього дозвіл, як-то аналітика.

1.7 Мова програмування Python

Python - це мультіпарадігменна мова програмування загального призначення з відкритим кодом з підтримкою об'єктно-орієнтованого, функціонального та процедурні стилів програмувань. Вона використовується як для автономних програм, так і для модулів програм у найрізноманітніших областях і, як правило, вважається однією з найбільш широко використовуваних мов програмування у світі [14, с.11].

Проте основна частина ознак цієї мови вказує на її схильність до об'єктно-орієнтованого стилю програмування. Ця мова не потребує явного оголошення змінних та є регістрозалежною.

Визначною особливістю мови Python є те, що вона не використовує фігурні або інші дужки для розділу блоків програми, цю роль виконує табуляція, що створює належну потребу у спеціалізованому текстовому редакторі, а точніше у середі розробчі. Проте вкладений блок також може з'являтися в одному рядку із заголовком заяви

якщо він складається лише з простих (безсполучникових) інструкцій.

Python містить такі структури даних як списки (lists), кортежі (tuples) і словники (dictionaries). Списки - схожі на одномірні масиви (але ви можете використовувати список, що включає списки - багатовимірний масив), кортежі - незмінні списки, словники - теж списки, але індекси можуть бути будь-якого типу, а не тільки числовими. Масиви в Python можуть містити дані будь-якого типу, тобто в одному масиві може перебувати числові, строкові та інші типи даних.

Написані інструкції виконуються послідовно, одна за одною, якщо у кодї програми не передбачено умовного розгалуження за допомогою операторів.

Зараз існує дві основні версії мови програмування - 2.X та 3.X, відповідно 3.X, постійно оновлюється та покращується. У цьому проекті буде використано Python 3.7.

Функції можна також створити за допомогою форми вираження лямбда, яке створює новий об'єкт функції і повертає його до виклику пізніше, замість того, щоб присвоїти його імені [12, с.84].

Спискові, кортежні та словарні вирази надають можливість ініціалізувати об'єкти цих класів одночасно з циклом обробки даних.

Вирази генератора досягають власних ефектів, не формуючи фізичний список, щоб утримувати всі результати. Генераторні вирази визначають набір результатів, але не зберігають весь список, щоб зберегти пам'ять; натомість вони створюють об'єкт генератора, який повертає елементи один за одним згідно протоколу ітерації, автоматично підтримуючи протокол ітерації попереднього елементу [12, с.51]

Також, мова програмування має наступні основні переваги:

- швидкість виконання програм написаних на Python дуже висока. Це пов'язано з тим, що основні бібліотеки Python
 - а. написані на C ++ і виконання завдань займає менше часу, ніж на інших мовах високого рівня;
- у зв'язку з цим можна писати свої власні модулі для Python на C або C ++;
- у стандартних бібліотеках Python ви можете знайти інструменти для роботи з електронною поштою, протоколами Інтернету, FTP, HTTP, базами даних тощо;
- скрипти, написані за допомогою Python виконуються на більшості сучасних ОС. Така переносимість забезпечує Python застосування в різних областях;
- Python підходить для будь-яких рішень в області програмування, будь то офісні програми, веб-додатки, GUI-додатки і т.д;
- над розробкою Python працювали тисячі ентузіастів з усього світу. Підтримкою сучасних технологій в стандартних бібліотеках ми можемо бути зобов'язані саме тому, що Python був відкритий для всіх бажаючих;
- код Python дуже легко читається, що робить його більш багаторазовим і доступним для обслуговування. Він має широкую підтримку передових програмних парадигм програмного забезпечення, таких як об'єктно-орієнтоване, функціональне та процедурне програмування;

– Python має чистий та елегантний стиль кодування. Він використовує синтаксис, подібний до англійської мови, і динамічно набирається. Отже, ви ніколи не оголошуєте змінну. Просте ініціалізація прив'язує ім'я до об'єкта будь-якого типу. Код Python значно менший, ніж еквівалентний код C ++ та Java. Це означає, що кожен тип є менш обмежений налагодженням і менше вимагає уваги. На відміну від компільованих мов, програми Python не потребують компіляції та посилання, що ще більше підвищує продуктивність розробника.

Обмеження з написання коду мовою Python, окрім синтаксису, контрольовані документом PEP 8, правила якого допомагають зробити код зручнішим для читання у співпраці. Цей документ дає умови кодування для Python при використанні вбудованих бібліотек.

1.8 Блокноти Jupiter як альтернативний спосіб використання програмної мови Python

Jupyter — це безкоштовний інтерактивний веб-інструмент з відкритим вихідним кодом, відомий як обчислювальний блокнот, який дослідники можуть використовувати для поєднання програмного коду, обчислювального результату, пояснювального тексту та мультимедійних ресурсів в одному документі. Обчислювальні ноутбуки існують протягом десятиліть, але, зокрема, Jupyter за останні кілька років зріс популярність. Цьому швидкому поширенню сприяла ентузіастна спільнота користувачів-розробників і оновлена архітектура, яка дозволяє ноутбуку розмовляти десятками мов програмування – факт, відображений в його назві, яка, за словами співзасновника Фернандо Переса, була натхненна мови програмування Julia (Ju), Python (Py) і R.

Один аналіз сайту обміну кодом GitHub нарахував понад 2,5 мільйона загальнодоступних ноутбуків Jupyter у вересні 2018 року, у порівнянні з 200 000 або близько того у 2015 році. Частково, каже Перес, це зростання пов'язане з удосконаленням веб-програм, які запускають такі програми, як Gmail і Google Docs; дозрівання наукового Python і науки про дані; і, особливо, легкість, з якою

ноутбуки полегшують доступ до віддалених даних, які інакше було б непрактично завантажувати

Блокнот Jupyter складається з двох компонентів. Користувачі вводять програмний код або текст у прямокутні клітинки інтерфейсної веб-сторінки. Потім браузер передає цей код внутрішньому «ядру», яке запускає код і повертає результати (див. наш приклад на go.nature.com/2uqq7ak). За підрахунками Переса, було створено понад 100 ядер Jupyter, які підтримують десятки мов програмування. Зазвичай кожен ноутбук може запускати лише одне ядро та одну мову, але існують обхідні шляхи. Один демонстраційний блокнот, наприклад, говорить на Python, Julia, R і Fortran.

Важливо, що ядра не повинні знаходитися на комп'ютері користувача. Коли майбутні користувачі LSST використовуватимуть ноутбуки Jupyter для аналізу своїх даних, код працюватиме на суперкомп'ютері в штаті Іллінойс, забезпечуючи обчислювальну мускулатуру, з якою не може зрівнятися жодний настільний ПК. Ноутбуки також можуть працювати в хмарі. Спільний проект Google, наприклад, надає інтерфейс на тему Google для ноутбука Jupyter. Це дає змогу користувачам співпрацювати та запускати код, який використовує хмарні ресурси Google — наприклад, графічні блоки обробки — і зберігати свої документи на Google Drive.

Найновішим варіантом Jupyter є JupyterLab, який був запущений як бета-версія в січні 2018 року і доступний (як і ноутбук Jupyter) або як окремий пакет, або як частина безкоштовного середовища науково-обчислювальної роботи Anaconda.

1.9 Мова розмітки XML

XML означає Extensible Markup Language. Це текстова мова розмітки, що походить від стандартної узагальненої мови розмітки (SGML).

Теги XML ідентифікують дані та використовуються для зберігання та організації даних, а не вказують, як їх відображати, як теги HTML, які використовуються для відображення даних. XML не збирається замінити HTML найближчим часом, але він відкриває нові можливості, застосовуючи багато успішних функцій HTML.

Є три важливі характеристики XML, які роблять його корисним у різноманітних системах і рішеннях

1. XML є розширюваним – XML дозволяє вам створювати власні теги або мову, що відповідає вашій програмі;
2. XML переносить дані, а не представляє їх – XML дозволяє зберігати дані незалежно від того, як вони будуть представлені;
3. XML є загальнодоступним стандартом – XML був розроблений організацією під назвою World Wide Web Consortium (W3C) і доступний як відкритий стандарт.

Короткий список використання XML говорить усе –

1. XML може працювати поза сценою, щоб спростити створення документів HTML для великих веб-сайтів;
2. XML можна використовувати для обміну інформацією між організаціями та системами;
3. XML можна використовувати для розвантаження та перезавантаження баз даних;
4. XML можна використовувати для зберігання та впорядкування даних, що може налаштувати ваші потреби в обробці даних;
5. XML можна легко об'єднати з таблицями стилів для створення практично будь-якого бажаного результату;
6. Практично будь-який тип даних може бути виражений як XML-документ.

Мова програмування складається з граматичних правил і власного словника, який використовується для створення комп'ютерних програм. Ці програми інструктують комп'ютер виконувати конкретні завдання. XML не може бути мовою програмування, оскільки він не виконує жодних обчислень або алгоритмів.

Зазвичай він зберігається в простому текстовому файлі і обробляється спеціальним програмним забезпеченням, здатним інтерпретувати XML.

За бажанням, документ XML може мати декларацію XML. Записується так –

```
<?xml version = "1.0" encoding = "UTF-8"?>
```

Де версія — це версія XML, а кодування визначає кодування символів, що використовується в документі.

Синтаксичні правила для XML-декларації

1. Оголошення XML чутливо до регістру і має починатися з "<?xml>", де "xml" пишеться у нижньому регістрі;
2. Якщо документ містить декларацію XML, то він має бути першим оператором XML-документа;
3. Оголошення XML суворо має бути першим оператором у XML-документі;
4. Протокол HTTP може замінити значення кодування, яке ви вводите в декларацію XML.

Синтаксис елемента — кожен XML-елемент потрібно закрити або початковими, або кінцевими елементами. Вкладеність елементів — XML-елемент може містити кілька елементів XML як дочірніх, але дочірні елементи не повинні перекриватися. тобто кінцевий тег елемента повинен мати те саме ім'я, що й останній невідповідний початковий тег. Кореневий елемент – XML-документ може мати лише один кореневий елемент. Чутливість до регістру — імена XML-елементів чутливі до регістру. Це означає, що назви початкового та кінцевого елементів мають бути точно в одному регістрі.

Атрибут визначає одну властивість для елемента, використовуючи пару ім'я/значення. XML-елемент може мати один або кілька атрибутів.

Правила синтаксису для атрибутів XML

1. Назви атрибутів у XML (на відміну від HTML) чутливі до регістру. Тобто HREF і href вважаються двома різними атрибутами XML;
2. Один і той самий атрибут не може мати два значення в синтаксисі;
3. Назви атрибутів визначаються без лапок, тоді як значення атрибутів завжди мають бути в лапках.

Посилання зазвичай дозволяють додавати або включати додатковий текст або розмітку в XML-документ. Посилання завжди починаються із символу «&», який є зарезервованим символом, і закінчуються символом «;». XML має два типи посилань –

1. Посилання на об'єкт — посилання на об'єкт містить ім'я між початковим і кінцевим роздільниками. Наприклад, `&` де `amp` — назва. Назва посилається на попередньо визначений рядок тексту та/або розмітки;
2. Посилання на символи – вони містять посилання, наприклад `A`, містять решітку («#»), за якою слідує число. Число завжди відноситься до коду символу Unicode.

Назви XML-елементів та XML-атрибутів чутливі до регістру, що означає, що назви початкового та кінцевого елементів мають бути записані в одному регістрі. Щоб уникнути проблем із кодуванням символів, усі файли XML слід зберігати як файли Unicode UTF-8 або UTF-16.

1. Пробіли, такі як пробіли, табуляції та розриви рядків між XML-елементами та між XML-атрибутиами, будуть ігноруватися;
2. Деякі символи зарезервовані самим синтаксисом XML. Тому їх не можна використовувати безпосередньо.

2 Основи переробки сміття і стан переробки сміття в Україні

2.1 Практичні засади переробки сміття

Переробка – це процес збирання та переробки матеріалів, які в іншому випадку були б викинуті як сміття та перетворення їх на нові продукти. Переробка може принести користь вашій громаді та навколишньому середовищу.

Переваги переробки:

- Зменшує кількість відходів, що відправляються на сміттєзвалища та сміттєспалювальні заводи;
- Зберігає природні ресурси, такі як деревина, вода та мінерали;
- Підвищує економічну безпеку за рахунок використання вітчизняного джерела матеріалів;
- Запобігає забрудненню, зменшуючи потребу в зборі нової сировини;
- Економить енергію;
- Підтримує місцеве виробництво та зберігає цінні ресурси;
- Сприяє створенню робочих місць у переробній та виробничій промисловості.

Кроки до переробки матеріалів.

Переробка включає три кроки нижче, які створюють безперервний цикл, представлений знайомим символом переробки.

1. Збір та обробка Існує кілька методів збору вторсировини, включаючи збір на узбіччі, центри видачі та програми здачі або повернення коштів. Після збору вторсировину відправляють на утилізаційну установку для сортування, очищення та переробки на матеріали, які можна використовувати у виробництві. Вторсировину купують і продають так само, як і сировину, а ціни зростають і знижуються залежно від попиту та пропозиції в Україні і світі;
2. Виготовлення. Все більше і більше сучасних продуктів виробляються з переробленого вмісту. Поширені предмети побуту, які містять перероблені матеріали, включають наступне:

- a. Газети та паперові рушники;
 - b. Алюмінієві, пластикові та скляні контейнери для безалкогольних напоїв;
 - c. Сталеві банки;
 - d. Пластикові пляшки для прального порошку.
 - e. Перероблені матеріали також використовуються по-новому, наприклад, відновлене скло в асфальті для мощення доріг або відновлений пластик в килимах і паркових лавках.
3. Придбання нових продуктів, виготовлених із перероблених матеріалів. Ми допомагаємо закрити цикл переробки, купуючи нові продукти з перероблених матеріалів. Існують тисячі продуктів, які містять перероблений вміст.

Також варто звертати увагу на наступні деталі:

- Продукти, які можна легко переробити;
- Продукти, які містять вторинну переробку.

Нижче наведено деякі з використаних термінів:

Продукт із переробленим вмістом – продукт був виготовлений із перероблених матеріалів, зібраних у рамках програми переробки, або з відходів, утилізованих під час звичайного виробничого процесу. На етикетці іноді вказується, скільки вмісту було з перероблених матеріалів.

Вміст після споживання – дуже схожий на перероблений вміст, але матеріал надходить лише з вторинної сировини, зібраної від споживачів або підприємств за програмою переробки.

Продукт, що підлягає вторинній переробці – продукти, які можна зібрати, переробити та переробити на нові продукти після їх використання. Ці продукти не обов'язково містять перероблені матеріали. Пам'ятайте, що не всі види вторсировини можна збирати у вашій громаді, тому обов'язково проконсультуйтеся з місцевою програмою переробки перед покупкою[5].

Деякі з поширених продуктів, які можна виготовити з переробленого вмісту, включають:

- Алюмінієві банки;
- Автомобільні бампери;
- Килимове покриття;
- Ящики для злаків;
- Комікси;
- Яєчні коробки;
- Скляні ємності;
- Пляшки з пральним порошком;
- Моторне масло;
- Газети;
- Паперові рушники;
- Вироби зі сталі;
- Мішки для сміття.

2.2 Сортування сміття в Україні

З 1 січня 2018 року, згідно із Законом України «Про відходи», українці в обов'язковому порядку повинні сортувати сміття і викидати його у відповідні баки. В цьому ж році вирішили позбутися сміттєпроводів у житлових багатоповерхових будинках і внесли зміни в ДБН: скасували їх обов'язкове проектування при новому будівництві та реконструкції будівель.

Дані зміни в законодавстві повинні допомогти за 14 років зменшити обсяги щорічно викидаємого сміття з 95% (2016 рік) до 30% (2030 рік). Втім до 2018 року вдалося зменшити обсяги тільки на 1.2%.

За даними Міністерства екології та природних ресурсів, в Україні кожного року накопичується близько 17 мільйонів тонн побутового і промислового сміття. З цього сміття сортується лише 5% і всього 1% сміття спалюється на заводі

Енергія в Києві. Повніст. усі інше сміття потрапляє на легальні і нелегальні звалища ТПВ, яких в нашій країні десятки тисяч[6].

До кінця 2014 року в Україні функціонувало 5 сміттєспалювальних заводів: в Дніпрі, Києві, Рівному, Севастополі, Харкові. Проте до 2020 року з них в робочому стані лишився один київський завод Енергія, який не тільки знищує відходи, а й забезпечує тепловою енергією згоряння житлові будинки.

Завод у Рівному для спалення сміття збудували в 2013 році та функціонував він всього 6 місяців. Зараз він теж працює, у ролі цілого комплекс з переробки відходів, цей комплекс зараз за данимим Міністерства екології та природних ресурсів переробляє понад 120 тисяч тонн сміття за рік.

Також в Україні іункціонує 22 сортувальних лінії, задача яких - сортуванням побутових відходів. Тут із загального обсягу отриманих відходів фільтрують картон, пластикові пляшки, залізні, скляні банки, а потім відвозять їх ло центрів переробки.

Переробкою саме промислових відходів займаються окремі компанії. Зараз їх кількість становить близько 28: в Києві і Київській області, Одесі, Полтаві, Запоріжжі, Львові, Білій церкві, Миколаєві та інших містах. Ці організації займаються заховання небезпечних будівельних відходів, безпечною утилізацією відпрацьованих батарей та олій, екномною переробкою документації, переробкою продуктів харчування. Частина цих компаній такоє має можливість отримувати зі сміття електричну енергію.

До 2022 року збираються побудувати найбільший в Україні завод з переробки промислових відходів у Житомирі. Виконавцем замовлення є хорватська компанія Tehnix, яка вже інвестувала в будівництво понад 10 мільйонів євро. Після зведення та запуску цього заводу у дію нарешті буде можливість закрити Житомирське міське сміттєзвалище.

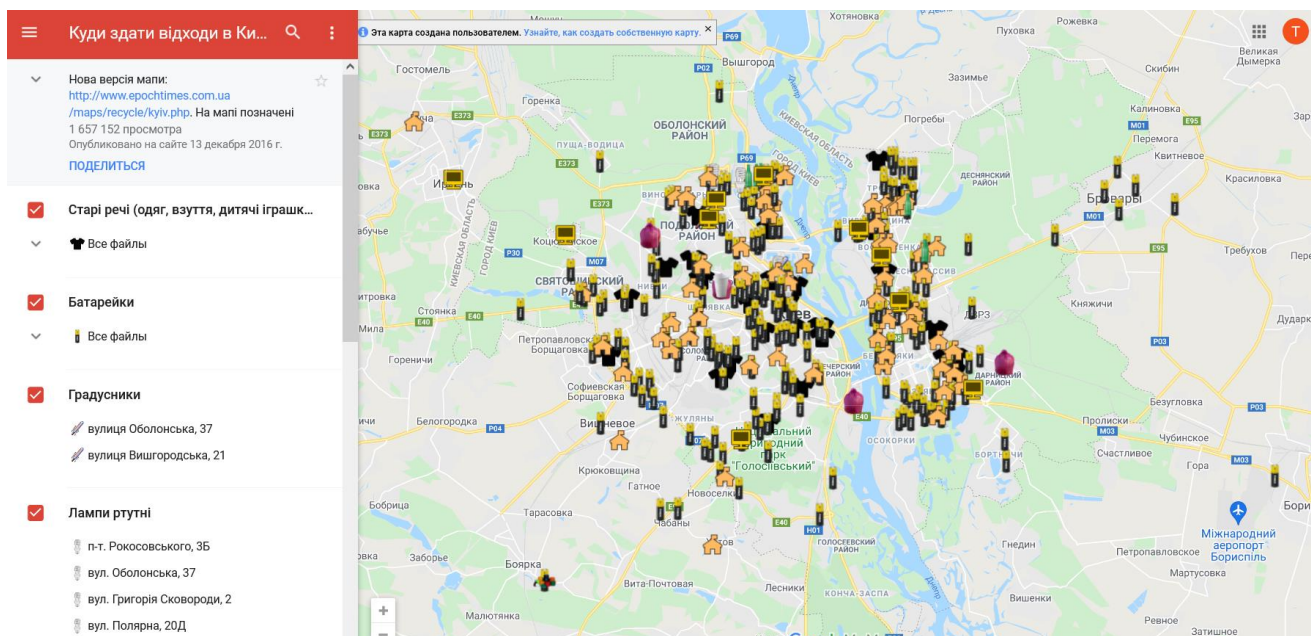
2.3 Огляд онлайн-карти «Куди здати відходи»

Видання «Велика Епоха» розробило всеукраїнський сервіс інтерактивних мап «Куди здати відходи». У ньому можна вибрати мапу, де позначено пункти

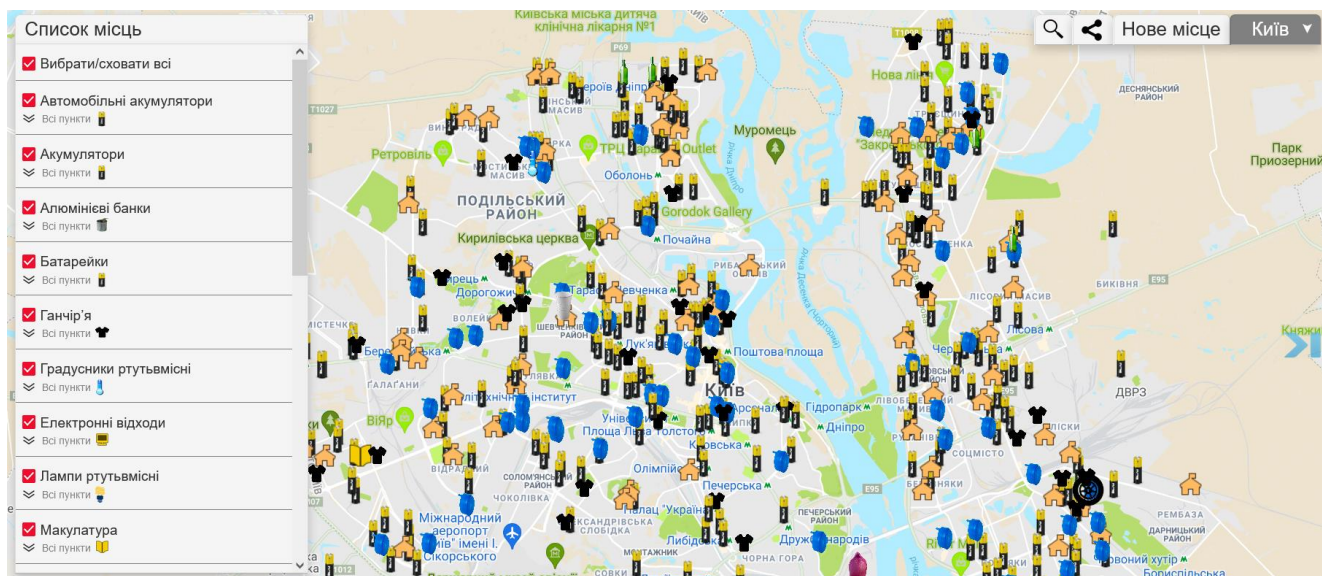
прийому різних видів відходів для таких міст: Києва, Харкова, Дніпропетровська, Львова, Одеси, Полтави. Для таких міст як Бровари та Херсон позначено пункти прийому батарейок.

На мапі у правому верхньому куті є кнопка перемикання міста — там можна вибрати будь-яке з цих міст. Раніше видання опублікувало мапу для кожного з цих міст окремо на сервісі мап «Гугл». Тепер всі ці мапи об'єднано в єдиному інтерфейсі.

На Малюнку 2.3.1 можна ознайомитись з мапою у старому інтерфейсі.



На Малюнку 2.3.2 можна ознайомитись із новою версією мапи.



Сервіс «Куди здати відходи» може здійснити мрію багатьох — жити в чистому та красивому місті і залишити його таким для майбутніх поколінь. Для цього потрібно не так уже й багато: навчитися сортувати відходи і відносити до спеціального пункту прийому, особливо небезпечне сміття (наприклад, старі батарейки та градусники, енергоощадні лампи). Інші види відходів, такі як пластик, шини, металобрухт, також приймають у спеціальних пунктах — вони йдуть на переробку[8].

3 Планування та проектування розподіленої геоінформаційної системи для визначення локацій переробки відходів

3.1 Теоретичні засади проектування розподіленої системи

Процес проектування систем розподіленої обробки даних включає наступні етапи [15, с.176]:

- виявлення інформаційних потреб кінцевих користувачів;
- концептуальне проектування;
- розробка архітектури системи;
- логічне проектування;
- налагодження і тестування прикладних програм;
- супровід.

Інформаційними потребами кінцевих користувачів у особі ДСМД є автоматизація та оптимізація процесів екологічного моніторингу стану поверхневих вод. На другому етапі здійснюється розробка схем архітектури та баз даних майбутньої системи. Усі надані у наступних пунктах розділу схеми створені у онлайн-редакторі Casoo. Серед переваг цього редактора є детальні інструменти організації схеми у групи та можливість створювати власні іконки програм, що використовуються у схемі. Етап розробки архітектури було розділено на 6 процесів:

- вибір апаратного забезпечення сервера даних;
- вибір програмного забезпечення сервера даних;
- організація баз даних та налаштування роботи сервера;
- проектування та розробка програмного додатку для транспортування даних;
- налаштування аналітичного програмного забезпечення;
- налаштування безпечного з'єднання з сервером даних.

Кожному з описаних процесів відведено окремий пункт.

Налагодження і тестування прикладних програм проведено під час їх розробки у відповідних процесах. У супровід входить детальний кошторис розробленої системи та напрями для її подальшого масштабування та функціонального розвитку.

3.2 Вибір апаратного забезпечення серверу даних

Вибір апаратного забезпечення серверу даних обраної архітектури був зумовлений, у першу чергу, наступними факторами:

- достатній обсяг пам'яті - від 2 терабайтів пам'яті на один носій та наявність RAID(0) контролеру для можливості розширення кількості носіїв та їх взаємодії;
- оптимальний обсяг оперативної пам'яті для швидкої обробки запитів до бази та роботи Microsoft Power BI;
- контролер бездротової мережі для можливостей віддаленого доступу;
- фізичні можливості для масштабування.

З огляду на ці фактори було обрано робочу станцію Dell Precision 5820 Tower Workstation. Це комп'ютер форм-фактору “вежа”. Згідно офіційної специфікації Precision 5820 Tower Workstation [16], дана модель має чотири основні переваги використання:

- потужна продуктивність, що гарантується завдяки процесорам Intel® Xeon® and Core® X та програмним забезпеченням Dell Precision Optimizer, що

містить оптимізовані драйвери для того апаратного забезпечення, що містить модель, та дає можливість його детального доналаштування;

- новітні технології - Radeon™ Pro та NVIDIA Quadro® серії графічних процесорів надають підтримку потужним процесам обробки даних, що відповідає потребам у масштабуванні для розширення функціоналу програмного додатку та оснащення звіту у Microsoft Power BI за допомогою алгоритму прогнозування. SSD контролер FlexBay також відповідає потребам масштабування та надає можливість встановлення SSD зберігачів та оптимізації їхньої роботи, що стане у нагоді при збільшенні місць забору проб та розширенні функціоналу програмного додатку, тобто збільшенню кількості операції запису, не рекомендованих для рухомих носіїв;
- інноваційний дизайн полягає у ергономічному корпусі - великій кількості задніх та передніх ручок, доступу до модуля живлення без потреби у інструментах та високому рівні фізичних засобів безпеки, що реалізовано у замках слотах жорстких дисках та зчитувачі смарт-карт;
- визначна надійність полягає у особливій системі охолодження.

З опціональних компонентів збірки вибрані:

- процесор Intel (R) Xeon Processor з частотою 3.6GHz;
- модуль живлення ємністю 120 Ватт;
- графічний процесор AMD Radeon Pro WX 3200;
- оперативна пам'ять RDIMM сумарною ємністю 16 гігабайт - два модулі DDR4 по 8 гігабайт;
- RAID-контролер MegaRAID SAS 9440-8i;
- жорсткий диск Nearline SAS AG-Enterprise ємністю 4 терабайти;
- карта віддаленого доступу Teradici Remote Workstation Access Host Card;
- контролер бездротових мереж Intel® Dual Band Wireless AC 8265 (та інтегрована мережева карта).

3.3 Вибір програмного забезпечення серверу даних

Програмне забезпечення, потрібне для встановлення на сервері можна розділити на дві частини: забезпечення з комплекту Dell Precision 5820 Tower Workstation та програми, яким потрібне наступне встановлення та налаштування.

Операційна система входить до комплекту робочої станції, Windows 10 Pro for Workstation, що повністю задовільняє потреби наступного програмного забезпечення та є зручним у адмініструванні.

Для управління сервером баз даних та його створення потрібно встановити СУБД PostgreSQL версії 9.6, разом з нею буде встановлено графічний інтерфейс для управління сервером баз даних - pgAdmin.

Для створення запланованих процесів транспорту даних буде використовуватися Планувальник Windows, що надходить у комплекті програмного забезпечення разом із оперативною системою.

Для редагування та функціонування скриптів на мові програмування Python буде встановлено Anaconda, менеджер пакетів та оточень та бібліотека більш ніж 7,500 програмних додатків з відкритим вихідним кодом. Anaconda надає свою власну середу розробки Spyder, яка має активну технічну підтримку та надає усі можливості для створення проектів на Python. Додаткові пакети будуть встановлені через менеджер пакетів pip та будуть детально описані у пункті 2.5.

Також буде встановлено інформаційну систему Microsoft Power BI. Настільна версія програми надається безкоштовно, для розміщення звітів на хостингу для віддаленого доступу до них потрібен аккаунт Microsoft Power BI. Додаткові вимоги до встановлення пакетів для інформаційної системи будуть описані у пункті 2.6.

Для налаштування дистанційного адміністрування сервера буде встановлено Microsoft Remote Desktop, програмне забезпечення для віддаленого до доступу до екрану комп'ютера за допомогою протоколу RDP.

3.4 Організація баз даних та налаштування роботи сервера

Для створення сервером баз даних та його керуванням буде використовуватися система управління базами даних PostgreSQL. Основною відмінністю цієї СУБД є те, що вона є повноцінним програмним рішенням для багатьох платформ, що робить її оптимальним рішенням для віддаленого сервера. Також PostgreSQL є проектом з відкритим вихідним кодом. Вирішальними особливостями, що вплинули на вибір цієї системи стали:

- підтримка геоінформаційних систем, що може спростити масштабування архітектури для автоматичного визначення координат місць забору проб при збільшенні їх кількості та, відповідно, ускладненні введенні кожного з координат місць вручну;
- можливість встановлювати доповнення до основного функціоналу СУБД, як, наприклад, утиліту для з'єднання таблиць з різних баз через зовнішні таблиці (PostgreSQL не підтримує з'єднання таблиць з різних баз);
- майже необмежені можливості масштабування самого сервера (найменше обмеження складає 1 гігабайт для розміру колонки);
- можливість створювати та зберігати послідовності (автоматизовані ряди) та використовувати їх як індекси таблиць - відповідно, розширені можливості індексування;
- майже унікальна можливість спадкування таблиць - коли одна, або більше дочірніх таблиць копіює структуру батьківської таблиці, проте з додавання своїх колонок, при запиті до батьківської є можливість отримати дані з усіх таблиць без з'єднань;
- можливість використовувати різні діалекти мов запитів та мов програмування як, Tcl, plsql, pgSQL / SQL, python, C++ тощо, що значно розширює можливості управління базою даних.

Для управління сервером використовувалися утиліти за замовчання - pgAdmin4, браузерне програмне забезпечення для управління серверами баз даних PostgreSQL. Використовувалася версія PostgreSQL 9.6, що задовольняє

потреби спроектованої архітектури, проте відсутність підтримки процедур може бути проблемою при подальшому масштабування, але оновлення СУБД проводиться досить швидко.

Вирішено використовувати 1 базу даних на сервері за таблицями:

- таблицю з даними локацій;
- таблицю з лейблами(що можна переробити) локацій с відносинами «one-to-many» с таблицею локацій;
- таблицю з описами локацій с відносинами «one-to-many» с таблицею локацій.

Повна схема серверу зображена нижче:

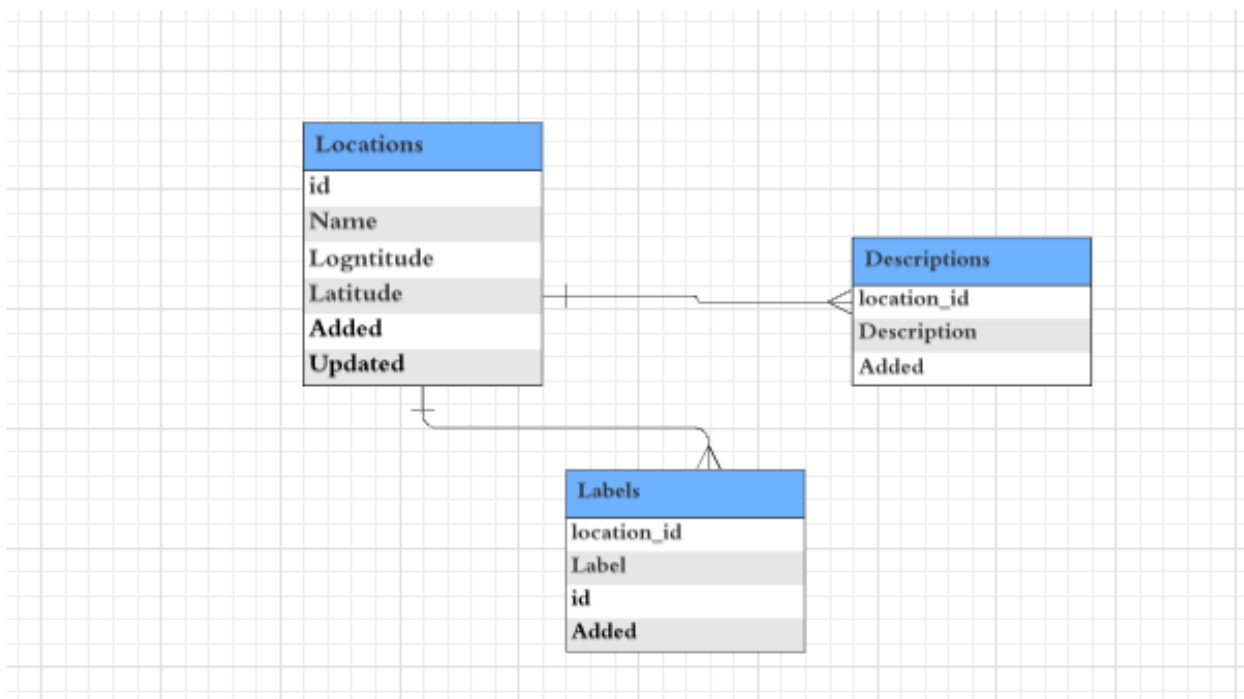


Рисунок 3.4.1

Після створення таблиць та баз відповідно до схеми, їх було заповнено тестовими даними. Дані для таблиць спочатку було згенеровано скриптом, написаним на мові програмування python з використання бібліотеки для наукових розрахунків numpy.

Наступним кроком у проектування сервера баз даних стало створення функцій, було додано наступні функції:

- додавання та видалення користувача;

- додавання та видалення локації;
- зчитування даних з csv-файлів оновлених спеціалізованим програмним забезпеченням для пристроїв аналізу стану поверхневих вод за занесення їх даних до усіх таблиць;
- набір функцій для оновлення таблиць баз організацій;
- обробка даних з усіх таблиць та заповнення даних таблиць відповідним станом.

Останнім кроком у організації сервера є створення вигляду, що буде формувати одну результуючу таблицю даних для використання її у ВІ системі. До вигляду входять дані про день, назву, значення та категорію індикатора, місце, групу та підгрупу.

3.5 Огляд різноманітних веб-картографічних платформ та вибір платформи

1. ArcGIS Online - це хмарний сервіс Esri з великою кількістю функцій. Він ідеально підходить для великих організацій, оскільки його легко інтегрувати. І найкраща частина всього – вона орієнтована на непрограмістів. Він створений провідною в світі компанією, що розробляє програмне забезпечення для ГІС, тому ви знаєте, що це клас вище за інших. Користувачам Esri подобається, тому що в ньому є все необхідне, як-от редагування, 3D-сцени та колекція на місцях. Створювати програми легко за допомогою Web App Builder (WAB). Ви можете розміщувати в хмарі Esri або ArcGIS Server. Потім ви можете редагувати його HTML, CSS і JavaScript. Javascript API від Esri відкриває масу можливостей для налаштування. Саме у візуалізації ArcGIS Online відстає від своїх конкурентів. Ви отримуєте менше контролю в порівнянні з Mapbox. Крім того, інтерфейс громіздкий у порівнянні з Mango Maps. Незважаючи на ці недоліки, він має деякі яскраві плями. City Engine — шпилька. Більше нічого в цьому класі немає. 3D-сцени пройшли довгий шлях. Це включає в себе LiDAR, який справді резонує

з вашою аудиторією. Нарешті, Esri створила цілу мову сценаріїв (Arcade) для налаштувань. ArcGIS Online тримається на таких важливих інструментах, як операційні інформаційні панелі, сюжетні карти та краудсорсинг. Survey123 — це форма, орієнтована на отримання даних у польових умовах. Тим часом ви можете відстежувати, куди їдуть польові бригади в режимі реального часу. Потім ви можете запланувати, де вони працюватимуть у додатку Workforce. Esri визнає потреби клієнтів, як і його портали відкритих даних. API вміщує розробку на замовлення, відкриваючи цілий світ можливостей;

2. Особливість сервісу Carto — це можливість працювати в Postgres + PostGIS як в веб-сервісі. Це означає, що ви отримуєте контроль над повністю керованою базою даних без необхідності налаштовувати, створювати резервні копії або виконувати оновлення. Незважаючи на те, що Carto призначений для програмістів, деякі з його платформ веб-картографування на основі рішень не потребують розробки. Розробники використовують Carto для створення власних інструментів на основі API. Використовуючи API SQL, ви взаємодієте (вставляєте, оновлюєте, видаляєте) зі своїми таблицями та даними за допомогою операторів SQL. Потім MAPS API запитує фрагменти карти для відображення. Дослідники даних, розробники та аналітики використовують PostGIS поза лаштунками для просторових запитів та аналізу. Наприклад, вони геозбагачують потоки даних на основі такої геометрії, як фінансові, демографічні та визначні місця. Сила Carto як платформи полягає в тому, як вона обслуговує основні дані у вигляді плиткових веб-карт. Має високий рівень контролю за картографією. Дисплей гладкий і приємний для ока. Інформаційні панелі готові до інтеграції та спільного доступу. Carto for deck.gl — це потужна платформа WebGL для великомасштабних наборів даних. Не кажучи вже про те, що ви також можете створювати програми за допомогою Google Maps і Mapbox GL. Carto спеціалізується на впровадженні аналізу у виробництво. Він

призначений для тих типів проектів, де ви глибоко копаєтеся в даних. Для створення просторових моделей на основі машинного навчання не потрібна розробка. Готові рішення, як-от вибір місця, планування території та маршрутизація транспортних засобів, вирішують повсякденні проблеми. Але це може бути трохи крутим, якщо ви не знаєте HTML, CSS, JavaScript і SQL. Також є можливість використання Cartoframes, які схожі на динамічні блокноти, які можна інтерактивно оновлювати «на льоту»;

3. Mapbox є одним з найпопулярніших сервісів. Foursquare, Pinterest та великі газети використовують Mapbox, тому що ви можете мапувати за допомогою чванства. Він орієнтований на програміста, як і Carto. Дисплей гладкий і приємний для очей. Mapbox зосереджено на плитках. Він швидкий, масштабований і повністю налаштований. Це робить його найкращим вибором для веб-сайтів із високою відвідуваністю. MBTiles є основою для зберігання наборів плиток. Mapbox має безліч функцій налаштування персоналізованих карт. Але якщо ви хочете повністю випробувати Mapbox, це все про API. Використовуючи API, ви можете програмно отримати доступ до інструментів і служб Mapbox. Нарешті, Mapbox Studio – це те, як конвертувати ваші дані в набори плиток і створювати стилі. Ось де Mapbox дійсно вибиває його з парку для налаштування веб-карти та керування даними. Чотири різновиди карт Mapbox – це карти вулиць, місцевості, трафіку та супутника. Ви можете інтегрувати будь-який з них у свою програму чи проект. Змінюючи компонент стилю, ви дійсно робите його власним. Нарешті, бібліотека JavaScript WebGL Mapbox — це те, як взаємодіяти з властивостями, параметрами та елементами керування на вашій карті. Mapbox надає будівельні блоки для створення карт, даних і просторового аналізу. Все це робиться в зручному браузері. Mapbox знає, як підвищити його на одну чи дві сходинки. Наприклад, він має API, які можна інтегрувати в такі програми, як локатори магазинів. Його вулична карта конкурує з

Google Maps. Ви можете зробити так багато в Mapbox, як-от AR-навігація, логістика та бізнес-аналітика;

4. У той час як інші платформи хмарного картографування зосереджені на тому, щоб створити якомога більше ударів, Mango Map використовує інший підхід. Вони пропонують найпростіший спосіб опублікувати веб-карту. Вони переосмислюють користувальницький інтерфейс, ставлять зручність перед складністю. На відміну від Mapbox і Carto, кодування не потрібно. Mango Map проста та ефективна. Ви не отримуєте багато функцій, таких як маршрутизація, сюжетні карти чи навіть 3D. Замість цього ви отримуєте інтуїтивно зрозумілий користувальницький інтерфейс із продуктом, не призначеним для програмістів. Ніяких API чи чогось подібного. Mango Map фіксує свою простоту завдяки гарному вибору стилів і шаблонів. Однією з чудових властивостей є те, що він групує точки в невеликому масштабі. Але коли ви вибираєте один, він автоматично розбиває точки, щоб ви могли бачити їх окремо. Mango Map також дозволяє додавати готові інструменти до ваших веб-карт. Простота – це найвища витонченість для карт Mango. Він не має API для персональних налаштувань;
5. GIS Cloud не є альтернативою ArcGIS Online. Але він має трохи атмосферу ArcGIS Online, без великої кількості ключових функцій. Інтерфейс вкладок є приємним на дотик, що надає йому вигляд і відчуття настільної ГІС. Хоча його база користувачів зросла до 70 000+ користувачів, це була трохи важка битва з усіма іншими слонами в кімнаті. GIS Cloud вас не здуває. Натомість, він здебільшого орієнтований на рішення. Наприклад, він зосереджений на зборі на місцях, редагуванні даних та краудсорсингу. Загалом, він виконує роботу для цих рішень для веб-картографування. Але це не займе зайву мілью, з деякими з їхніх найкращих прикладів на каналі GIS Cloud. Візуалізація – це не те, де сяє GIS Cloud. Звичайно, ви контролюєте маркування. Також можна стилізувати тематичні карти. Крім того, він

має приємний дотик до вкладок, що надає йому трохи настільної ГІС-програми. Якщо вам потрібен лише невеликий набір інструментів, скористайтеся ним. Але шукайте в іншому місці, щоб вийти за межі основ веб-картографування.

Було вирішено використовувати платформу Carto. Основними перевагами сервісу, що вплинули на вибір, були:

1. Можливість приєднати сервер даних PostgreSQL;
2. Блокноти Cartoframes, у яких можна писати аналітичні модулі на Python, яку мені, як автору системи, зручно використовувати, та яка підходить для гео-аналітики;
3. Можливість обирати між власним веб-картографічним движком Carto, Google Maps та AWS. Нажаль, для відображення детальної карти Києва підходить тільки Google Maps;
4. Переваги кастомного дизайну картових рівнів;
5. Можливість створити автоматизації на основі API;
6. Переваги швидкого масштабування;
7. Остання, проте досить важлива перевага – це опція створення картового рівня на основі даних з табличних файлів для тестів та налаштувань.

3.6 Налаштування безпечного з'єднання з сервером даних

Для захисту дистанційного доступу до серверу баз даних буде створено 4 заходи:

- перевірка доступу за допомогою сертифікатів SSL;
- аутентифікація користувачів;
- зберігання паролів у зашифрованому форматі за допомогою кодування MD5;
- використання протоколу SFTP для передачі файлів.

SSL-сертифікат - это цифровий підпис вузла у мережі, який потрібен для роботи протоколу захисту передачі даних. Тобто інформація, яка є об'єктом передачі, захищена від сторонніх користувачів. Ще за допомогою цього сертифікату можна перевірити справжність онлайн-ресурсу, так як у сертифікаті є

дані про ресурс. Тестові сертифікати для перевірки роботи передачі даних захищеним каналом можна отримати за допомогою утиліти командного рядка openssl, проте сертифікати, що використовуються реальними ресурсами - мають бути отримані у ліцензійних центрах.

Існує три види SSL-сертифікатів:

- DV (domain validation) SSL — підтверджує справжність домену, а також шифрує та захищає дані при передачі за допомогою протоколу https;
- OV (organization validation) SSL — окрім захисту даних гарантується належність домену до конкретної організації;
- EV (extended validation) — те саме, що й OV, проте перевіряється також комерційна діяльність компанії.

Для захисту серверу даних вистачить DV SSL-сертифікату. Сервери PostgreSQL мають вбудовану підтримку SSL-сертифікатів, вимагається лише наявність встановленої утиліти openssl на стороні клієнту та серверу та зміни значення опції ssl у файлі конфігурацій postgresql.conf на “on”.

Аутентифікація користувачів буде проводитися у програмному додатку та при відвідуванні СУБД. Профілі користувачів для програмного додатку будуть зберігатися у окремій таблиці у окремій таблиці на сервері, тому що там будуть перелічені як адміністратори серверу, так і лаборанти, що будуть відповідальні за ввід даних. Через повну розмежованість їх доступів (адміністратори мають заходити тільки безпосередньо до сервера, тоді як лаборанти виконують усі обмежені операції вводу-виводу через програмний додаток та інформаційну систему), тільки профілі адміністраторів будуть занесені до реєстру користувачів СУБД.

Паролі користувачів-лаборантів, будуть додатково захищені у таблиці кодування MD5. MD5 - це функція хешування, тобто згортання вхідного масиву у бітовий рядок. MD5 складається з 5 кроків:

- дописування одиничного байту 0x80, а потім - нульових бітів, до тех пор, пока довжина повідомлення не буде рівна 448 по модулю 512;
- дописування 64-бітного представлення довжини вихідного повідомлення;
- далі ініціалізується буфер;
- дані трансформуються за допомогою чотирьох логічних виразів;
- потім відбувається побітова трансформація та додавання шумів.

Протокол SFTP (SSH File Transfer Protocol) - це протокол, який використовується для встановлення безпечного з'єднання між віддаленим сервером та комп'ютером. Безпечна оболонка використовує шифрування відкритого ключа, щоб забезпечити надійну аутентифікацію користувача та безпечний зашифрований зв'язок через Інтернет. Підключення до серверів SFTP встановлюються або шляхом автентифікації пароля, або з використанням аутентифікації публічних / приватних ключів. Якщо аутентифікація пароля надійно передається, користувачеві потрібно лише ім'я користувача та пароль для входу на сервер SFTP. Інший метод - використання пар ключів. Генерується пара відкритих та приватних ключів, а відкритий ключ зберігається на сервері SFTP. Клієнт із приватним ключем під час входу в систему аутентифікується з сервером, і якщо ключі відповідають, клієнт SFTP отримає доступ до системи. Додаткову безпеку можна також додати до приватного ключа в приватний ключ. Основними перевагами цього протоколу є:

- Обмін даними. Він не використовує окремі канали, як це робить FTP. Це усуває незручності обслуговувати адміністраторів, яким інакше потрібно відкрити діапазон портів для звичайних FTP-з'єднань;
- Безпечність;
- Мета-дані - За допомогою SFTP користувачі можуть отримувати мета-дані, такі як дата, час, розмір, дозволи та іншу інформацію, в той час як деякі сервери FTP, особливо старіші, можуть мати проблеми з правильною інформацією, а інші не надають їх зовсім.

3.7 Збір даних локацій переробки відходів у Києві

Онлайн карта «Куди здати відходи в Києві» мала стару версію у вигляді кастомної карти на основі сервісу Google Maps. Так як стара версія все ще доступна та підтримується, кожен користувач отримує можливість завантажити геодані, на основі яких побудована карта. Ці геодані і будуть використовуватися як початкові дані для побудови нової веб-карти.

На Малюнку 3.7.1 можна ознайомитись із початковими даними.

```

<StyleMap id="icon-503-DB4436">
  <Pair>
    <key>normal</key>
    <styleUrl>#icon-503-DB4436-normal</styleUrl>
  </Pair>
  <Pair>
    <key>highlight</key>
    <styleUrl>#icon-503-DB4436-highlight</styleUrl>
  </Pair>
</StyleMap>
<Folder>
  <name>
    Старі речі (одяг, взуття, дитячі іграшки, ігри для розвитку дітей, книги, предмети гігієни, канцелярські товари)
  </name>
  <Placemark>
    <name>вул. Милославська, 49</name>
    <description>
      Адреса: Київ, вул. Милославська, 49<br>Що приймають: <br>Час роботи: вт і чт з 18:00 до 20:00<br>Телефон: (050) 611-19-25; (098) 721-64-67<br>Примітка: 2 поверх — відділення «Самопоміч». Нова версія мапи: http://www.epochtimes.com.ua/maps/recycle/kyiv.php.
    </description>
    <styleUrl>#icon-503-DB4436</styleUrl>
    <ExtendedData>
      <Data name="Адреса">
        <value>Київ, вул. Милославська, 49</value>
      </Data>
      <Data name="Що приймають">
        <value/>
      </Data>
      <Data name="Час роботи">
        <value>вт і чт з 18:00 до 20:00</value>
      </Data>
      <Data name="Телефон">
        <value>(050) 611-19-25; (098) 721-64-67</value>
      </Data>
    </ExtendedData>
  </Placemark>
</Folder>

```

Більшу увагу варто звернути на шлях отримання корисної інформації з результату експорту даних карти. Функція MyMaps Google Maps дозволяє створювати та зберігати прості карти (лінії та точки) і ділитися ними зі світом. Але спільний доступ здійснюється через мережеве посилання, а не через окремий файл KML, і ви не можете зберегти дані безпосередньо на своєму комп'ютері. Проте у

завантаженому KML можна знайти посилання для завантаження XML-файлу з усіма точками карти.

Після цього мною був створений невеликий скрипт на мові програмування Python, можливості якої описані раніше. Скрипт з використання вбудованих модулів мови програмування та модулю pandas дав можливість відкрити отриманий раніше файл, провести його парсинг та зберегти корисні для карти дані(адресу, опис локації, довготу та широту) у форматі csv(зручному для CartoDB сервісу платформи Carto).

Сніпет скрипту надано нижче:

```
import pandas as pd
from bs4 import BeautifulSoup

df = pd.read_table("Куди здати відходи в Києві — мапа видання «Велика
Епоха».kml.xml")
with open("map.txt", 'r') as f:
    data = f.read()

df = pd.DataFrame(columns = ['address','desc','longitude','latitude','label'])
for row in data.split('Placemark')[1:]:
    if row != '>\n    <':
        try:
            label = row.split('Що приймають: ')[1].split("<br>")[0]
        except:
            label = ""
        try:
            df = df.append({
                'address': row.split('<description><![CDATA[')[1].split("Нова версія
мапи")[0],
                'desc': row.split('<name>')[1].split("</name>")[0],
```

```
'label':label,  
  
'longitude':float(row.split('coordinates')[1].split(',')[0].replace('>\n','').replace('','')),  
  
    'latitude': float(row.split('coordinates')[1].split(',')[1])  
    },ignore_index=True)  
except:  
    print(row)  
  
df[['desc','longitude','latitude','label']].to_csv('map_data.csv', index = False)
```

Після повторного аналізу даних також було додано поле label, у якому записано ресурси придатні до переробки у локації(якщо інформація наявна).

З кінцевими даними після обробки можна ознайомитись на Малюнку 3.7.2

Data table

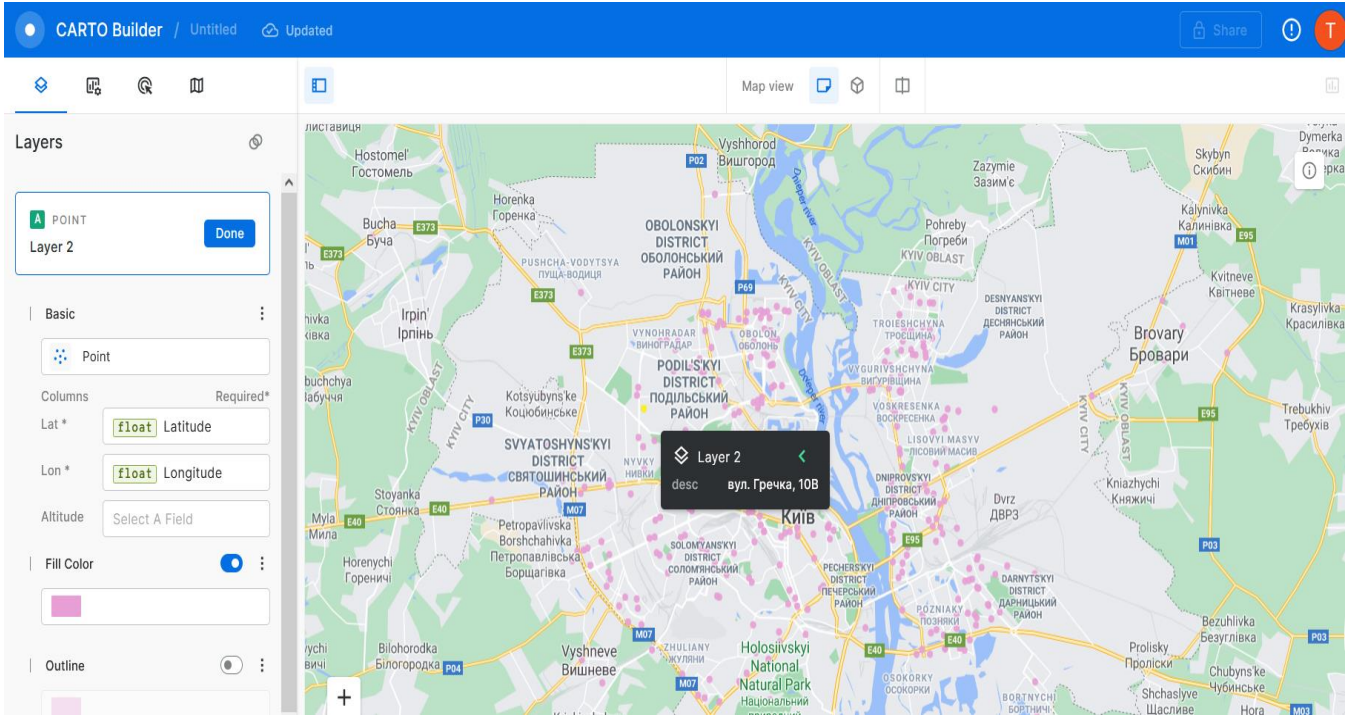
A kyiv_recycling1

| desc | label | longit... | latitu... |
|---------------------------|-----------|-----------|-----------|
| string | string | float | float |
| вул. Бучми, 6-А | Батарейки | 30.604052 | 50.431167 |
| вул. Васильківська, 28 | Батарейки | 30.492211 | 50.394412 |
| вул. Севастопольська, 1. | Батарейки | 30.656053 | 50.429595 |
| Ругаєвська набережна | Батарейки | 30.592375 | 50.439835 |
| вул. Кибальчича, 11-в | Батарейки | 30.598786 | 50.493022 |
| вул. Левандовська, 5-а | Батарейки | 30.545792 | 50.441742 |
| вул. Дніпроводська, 3 | Батарейки | 30.461867 | 50.567227 |
| Трага Київ — Олега (7 км) | Батарейки | 30.412559 | 50.346118 |
| вул. Маршала Тимошова | Батарейки | 30.504667 | 50.512892 |
| вул. М. Трубіліні, 1-6 | Батарейки | 30.407438 | 50.404909 |
| вул. Булгакова 11а | Батарейки | 30.404722 | 50.411108 |
| вул. Каптевільшівлі, 9а | Батарейки | 30.384338 | 50.422208 |
| вул. Полярна, 12/1 | Батарейки | 30.468387 | 50.518222 |
| вул. Бикова, 7-а | Батарейки | 30.621225 | 50.516401 |
| вул. Кірова, 57 | Батарейки | 30.794751 | 50.504947 |
| п-т Героїв Сталінграда | Батарейки | 30.513601 | 50.523344 |
| вул. Г. Дніпра, 226 | Батарейки | 30.495891 | 50.520797 |
| бул. Лесі Українки, 24 | Батарейки | 30.5354 | 50.429989 |
| вул. Скляренко, 5-6 | Батарейки | 30.464501 | 50.496948 |

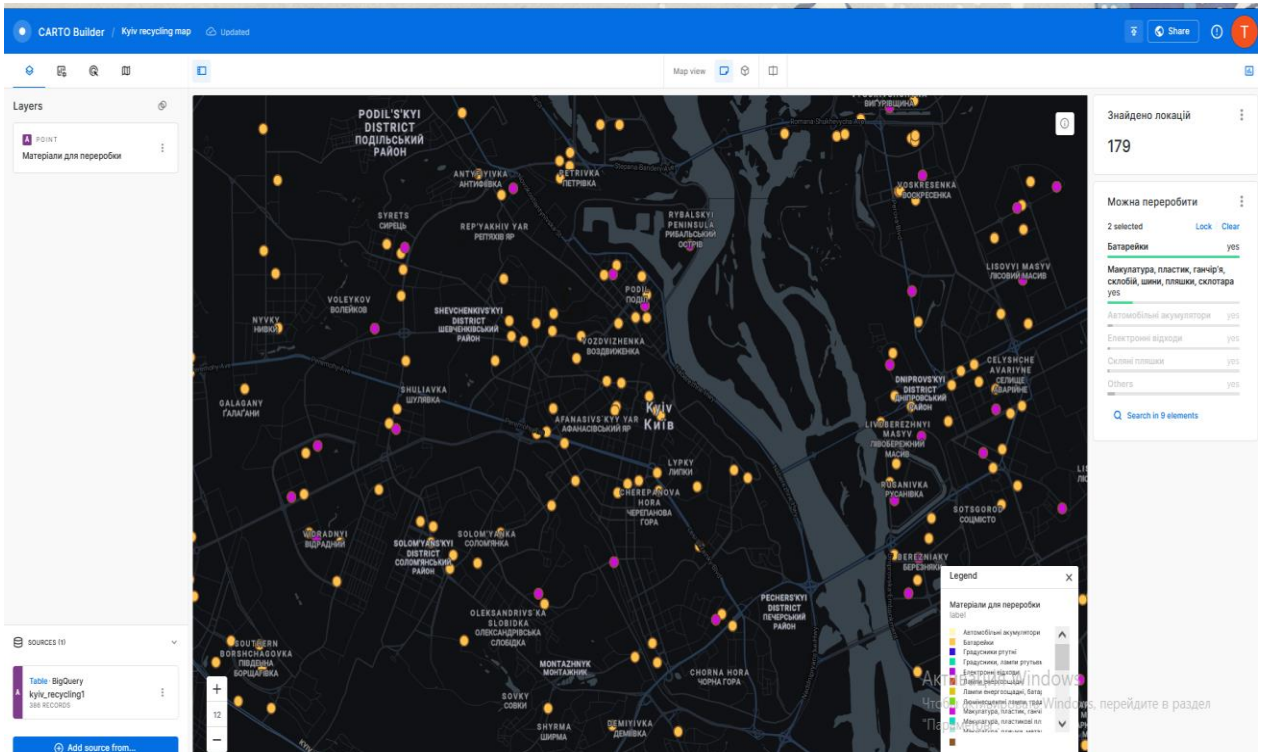
3.8 Створення карти Києва основі веб-картографічної платформи Carto

Після отримання файлу з даними координатів та адресами локацій, я вивантажила їх в редакторі Carto за допомогою опції Add new source.

Результат зображення даних на карті показано на Малюнку 3.8.1



Після перевірки правильності відображення даних я переформила дизайн карти. Результат дизайну карти можна побачити на Малюнку 3.8.2



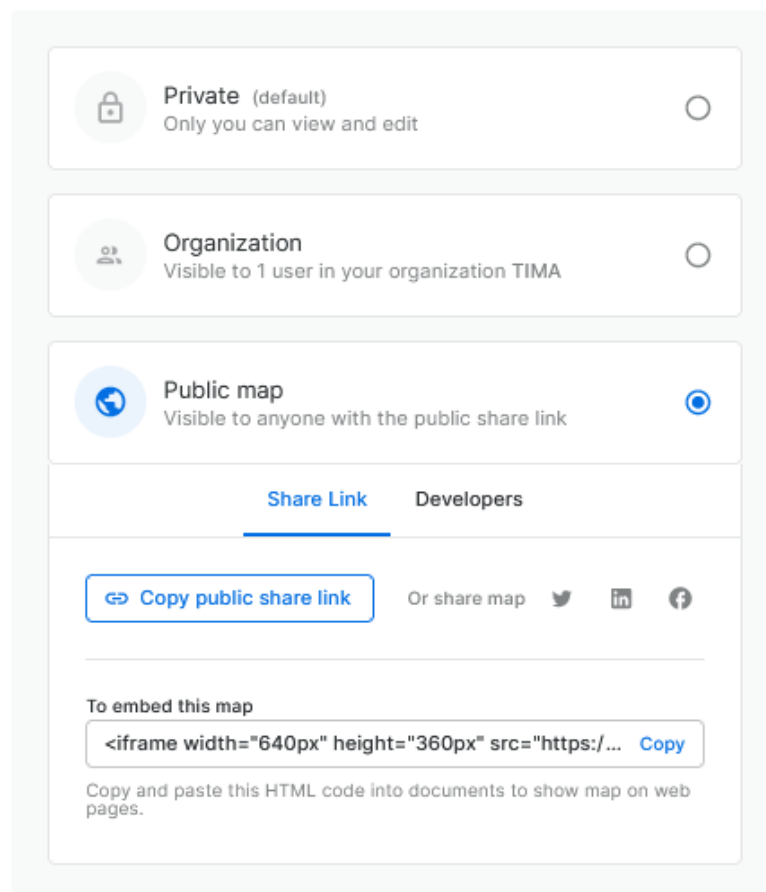
Також, на малюнку можна помітити нові елементи карти:

1. Легенда для лейблів локацій – кольори для точок кожного типу перероблюваних ресурсів;

2. Віджет «Знайдено локацій», вказує на кількість локацій, знайдених у виділеній ділянці карти;
3. Віджет «Можна переробити» вказує, які відходи можна переробити у виділеній ділянці карти та скільки локацій для переробки знайдено.

Після закінчення дизайну я опублікувати карти через кнопку Share як показано на Малюнку 3.8.3.

Sharing options



Done

3.9 Кошторис та обмеження розробленої системи

Ціна та обмеження кожного конкретного компонента розробленої розподіленої системи надані у таблиці:

Таблиця 3.9.1 - Кошторис

| Компонент | Ціна, \$ | Обмеження |
|---------------------------------------|----------|---|
| Dell Precision 5820 Tower Workstation | 1189.00 | 6 2,5-дюймових або 5 3,5-дюймових дисків; 18 ядер; до 600 Вт (графічний процесор); до 68 Терабайтів сховища; |
| Windows 10 Pro for Workstations | 127.00 | 4-х ядер |
| DDR4 2933MHz RDIMM ECC Memory | 130.00 | |
| Nearline SAS AG-Enterprise Hard Drive | 143.00 | 4 Терабайти |
| MegaRAID SAS 9440-8i | 146.00 | |
| Remote Access Host Card, Tera2 | 222.0 | |
| Intel® Dual Band Wireless | 47.00 | |
| Enterprise Carto pricing | 1639.00 | на один рік для одного активного користувача (адміністратора) |
| Microsoft Remote Desktop | | |

Продовження Таблиці 3.9.1

| Компонент | Ціна, \$ | Обмеження |
|----------------|----------|---|
| Anaconda | | |
| PostgreSQL 9.6 | | не має підключення до геоінформаційних систем; у версії відсутня можливість створювати процедури; немає можливості поєднувати таблиці з різних баз. |
| Сума | 1.868 | |

3.10 Передбачені можливості масштабування

Спроектвана архітектура пропонує чимало можливостей для масштабування та розширення функціоналу усіх її компонентів.

Серед них можна виділити наступні основні напрямки:

- 1) Спрощення операції додавання нових локацій через автоматизацію процесу введення координат за назвою місця. Для цього потрібно буде внести зміни до користувацького інтерфейсу, спроектувавши зручний спосіб відображення та перевірки координат за назвою. Після спроектованих змін потрібно буде внести їх до програмного додатку у модулі віджетів, а у модулі функцій відповідно змінити обгортку для функції додавання місця. У СУБД потрібно буде підключити підтримку геоінформаційних систем та змінити створену раніше функцію додавання місця;
- 2) Додати можливість оновлення списку локацій, їх лейблів та описів. Для цього потрібно буде провести роботу з проектування користувацького інтерфейсу для зручного введення усієї потрібної інформації. Після спроектованого нового вікна його потрібно буде додати у модуль віджетів, а у модуль функцій відповідні обгортки для функцій оновлення таблиць, що заздалегідь створені;

- 3) Придбання та встановлення додаткових накопичувачів за росту кількості локацій, додавання нових міст, нових типів локацій або зміні періодичності оновлення даних у сховищі. Встановлений на робочій станції RAID-контроллер надає можливість оперування кількома накопичувачами одночасно;
- 4) Додавання нових місць з локаціями переробки відходів, нових типів локацій, нових полей опису тощо;
- 5) Створення аналітичних додатків на основі Cartoframes та Python;
- 6) Додати прогнозу модель до звіту. Для цього потрібно:
 - a) Обрати та встановити пакет для побудови статистичних моделей мовою Python;
 - b) Написати скрипт для обробки вхідних даних, ініціалізації моделі та обробки її вихідних даних;
 - c) Протестувати модель та відлагодити відповідно її гіперпараметри та вхідні дані;
 - d) CSV-файл з вихідними даними додати до джерел даних CartoDB;
 - e) Спроекувати та реалізувати відповідну сторінку звіту;
 - f) Підключити Google Analytics після дослідження даної можливості та налаштувати збереження даних користувачів у окрему базу – це може бути використано для комерційного використання даних після анонімізації для спонсування кошторису карту. Також на основі даних користувачів можна випускати незалежні аналітичні дослідження для центрів переробки сміття, точки яких відображені на карті. Останній варіант використання цих даних – планування при оновленні даних з пріоритетами;
 - g) Додати скрипт моделі до переліку запланованих дій Планувальника Windows з визначеною періодичністю, що може суттєво відрізнятись від періодичності оновлення даних.

ВИСНОВКИ

Протягом виконання даного проекту було розроблено готову до використання розподілену систему для підтримки онлайн-ГІС для пошуку локацій переробки сміття у Києві. Було проведено аналіз сучасної ситуації онлайн-ГІС для переробки в Україні та оцінено потреби та обсяги для подальшої реалізації, масштабування, розвитку, фінансової підтримки системи. Проведено дослідження законодавчої бази переробки відходів в Україні.

Мною було проведено аналіз виробників веб-картографічних платформ та обрано найбільш оптимальний варіант. Було надано опис теоретичних засад інструментів, що використовувалися для проектування системи:

- мова програмування Python;
- СУБД Postgresql;
- веб-картографічна платформа Carto.

На основі вже існуючих аналогічних проектів було проведено порівняльну аналітику та зібрано реальні дані для тестування карти. Було створено та представлено варіанти схем майбутньої системи та реалізовано найоптимальніший варіант.

Обраний варіант архітектури було успішно реалізовано – розроблено та протестовано усі компоненти архітектури та зв'язок між ними. Запроваджено додаткові міри захисту даних та оптимізації їх передачі. Повну роботу системи було успішно протестовано зі реальними даними.

У якості подальшого супроводу надано перелік можливостей масштабування системи, кошторис з окресленими обмеженнями кожного компоненту.

Упровадження розробленої архітектури та власне онлайн-карти може бути практичним інструментом як для еко-ініціативних містян, так і для організаторів ініціатив з переробки відходів у Києві.

Останнім але не менш важливим за інші аспектом даної роботи було дослідження історії веб-картографії в Україні та світі та вивчення реальних кейсів

використання онлайн-ГІС у тому числі веб-карт, що використовують краудсорсингові джерела даних, для вирішення глобальних проблем людства. Вражаючі історії про кризовий веб-мапінг від Open Street Map є яскравим прикладом того як некомерційні веб-картографічні проекти можуть змінити світ на краще. Наявність подібних проектів, хоч і меншого масштабу, в Україні було однією з моїх мотивацій привнести у даний проект свою технічну експертизу.