МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Київський національний університет імені Тараса Шевченка

Навчально-науковий інститут «Інститут геології»

Кафедра геоінформатики

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

доктор технічних наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_ Зацерковний В.І.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 р.

**ДИПЛОМНА РОБОТА**

ВИПУСКНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ

"БАКАЛАВР"

**Тема: «Створення інформаційної бази даних для збереження і обробки геологічної інформації»**

|  |  |
| --- | --- |
| Виконав:  студент 4 курсу: Ганошенко Владислав Андрійович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (підпис) |
| Керівник:  д-р геолог.наук, проф.: Ковальчук Мирон Степанович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (підпис) |

Київ-2018

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Навчально-науковий інститут «Інститут геології»

Кафедра геоінформатики

Напрям 6.040103 «Геоінформатика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доктор технічних наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_ Зацерковний В.І.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**

на виконання дипломної роботи

Ганошенка Владислава Андрійовича

1. Тема дипломної роботи: **Створення інформаційної бази даних для збереження і обробки геологічної інформації,** затверджена протоколом № \_\_ засідання кафедри геоінформатики від «\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 р.

2. Термін виконання роботи: з 25.09.2017 по 25.06.2018 р.

3. Вихідні дані до роботи: геофізичні дані дослідження свердловин Полтавського району.

4. Зміст дипломної роботи: створено користувацький інтерфейс для взаємодії із базою даних для зберігання та обробки геологічної інформації. Реалізовано можливість зчитування інформації з документів різних типів. Запроваджено можливість проведення складних геофізичних розрахунків за допомогою простих маніпуляцій з даними, без використання стороннього програмного забезпечення. Розроблений механізм експорту даних з віртуальних баз та віддалених серверів. Проведено дослідження щодо шляхів використання додатку на підприємствах геологічного спрямування.

**Календарний план-графік**

|  |  |  |  |
| --- | --- | --- | --- |
| № пор. | Завдання | Термін виконання | Відмітка про виконання |
| 1. | Опрацювання спеціальної літератури за тематикою роботи | 25.09.2017 −  01.11.2017 | Виконано |
| 2. | Підготовка матеріалів першого розділу дипломної роботи | 02.11.2017 −  05.01.2018 | Виконано |
| 3. | Підготовка матеріалів другого розділу дипломної роботи | 05.01.2018 –  21.02.2018 | Виконано |
| 4. | Підготовка матеріалів третього розділу дипломної роботи | 21.02.2018 –  15.03.2018 | Виконано |
| 5. | Оформлення висновків дипломної роботи | 15.03.2018 –  28.04.2018 | Виконано |
| 6. | Оформлення графічних додатків дипломної роботи | 28.04.2018 –  06.05.2018 | Виконано |
| 7. | Підготовка до захисту дипломної роботи | 06.05.2018 –  20.06.2018 | Виконано |
| 8. | Попередній захист дипломної роботи | 21.06.18 | Виконано |
| 9. | Захист дипломної роботи | 27.06.18 | Виконано |

Завдання прийняв до виконання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ганошенко В.А.

/підпис/

Керівник дипломної роботи

д-р геолог.наук, проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ковальчук М.С

/підпис/

Дата отримання завдання “25” листопада 2018 р.

**РЕФЕРАТ**

Ганошенко Владислав Андрійович

Створення інформаційної бази даних для збереження і обробки геологічної інформації

Спеціальність: «Геоінформатика»

Київ – 2018

Стор. − , таблиць − , рисунків –

Ключові слова: Електронна таблиця, база даних, користувацький інтерфейс, C#

Мета роботи: створення програмного додатку для зберігання та обробки геологічної інформації за допомогою високорівневої мови програмування С#.

Основний зміст – створено користувацький інтерфейс для зберігання та обробки геологічної інформації. Реалізовано можливість зчитування інформації з документів різних типів. Запроваджено можливість проведення складних геофізичних розрахунків за допомогою простих маніпуляцій з даними, без використання стороннього програмного забезпечення. Розроблений механізм експорту даних з віртуальних баз та віддалених серверів. Проведено дослідження щодо шляхів використання додатку на підприємствах геологічного спрямування.

Наукові результати – зберігання та обробка геологічної інформації за допомогою розробленого програмного додатку дали змогу покращити продуктивність роботи під час інтерпретації даних і знизити час на проведення складних розрахунків без використання сторонніх програмних додатків.

Практичне значення роботи полягає у тому, що за допомогою розробленого програмного додатку можна автоматизувати велику кількість операцій при введенні, обробці, розрахунках та імпорті геологічної інформації.

|  |
| --- |
|  |

ЗМІСТ

**ВСТУП**…………………………………………………………….….…………....6

**РОЗДІЛ 1**. Загальна характеристика області досліджень…………………........9

Функціональні вимоги до додатку

Концептуальна структура баз даних

* 1. Проектування структури додатку та бази дани……………………………….6
  2. Особливості реалізації вводу даних з MS SQL бази даних….………………7

**РОЗДІЛ 2.** Засоби розробки програмного додатку.............................................

2.1 Microsoft Visual Studio 2017

2.2 Microsoft .NET Framework

2.3

**РОЗДІЛ 3.** Проектування форми додатку……….……………………………….8

2.1 Аналіз та методи вирішення задачі…………………………………….………3

2.2 Розробка алгоритму вирішення задачі…….……………………………………5

2.3 Розробка користувацького інтерфейсу…………………………...……………..7

**РОЗДІЛ 3**. Програмна реалізація додатку для маніпулювання даними бази даних………..7

3.1 Розробка меню……………………………………………………………………6

3.2 Елементи керування DataGridView……………………………………………..6

3.3 Елементи керування GroupBox…………………………………………………7

3.4 Розміщення елементів управління типу Label, Button, TextBox і ComboBox…………………………………………………………………………….7

3.5 Створення форм для елементів крісталічного фундаменту та кору вівітрювання………………………………………………………………………….7

3.5 Розміщення базових елементів таблиці на форму……………………………..8

3.6 Розміщення елементів групи опису порід, зони……………………………….8

3.7 Розміщення елементів групи вміст корисних копалин………………………..8

3.8 Вивід даних в dataGridView1…………………………………;;;………………9

3.8.1 Текст SQL-запиту…………………………………………………;…………..8

3.8.2 Програмування події кліку на кнопці «Застосувати»……….……………….9

3.8.3 Додавання методу button1\_Click() в метод Form1\_Load()…………………..2

3.9 Формування списку елементів керування comboBox1………………………..2

3.10 Створення форми «Form2.cs»………………………………………………….1

3.11 Програмування подій кліку на кнопках «Додати» та «Відмінити»…………2

3.12. Програмування методів відображення даних в елементах керування dataGridView1, dataGridView2, dataGridView3, dataGridView4, dataGridView5………………………………………………………………………..2

3.13. Програмування обробників подій зміни активної комірки в компонентах dataGridView1, dataGridView2, dataGridView3, dataGridView4, dataGridView5………………………………………………………………………..3

3.14. Побудова рядків полів для фільтра в елементах управління comboBox1 і comboBox2……………………………………………………………………………3

3.15 Програмування події виклику форми Form2 з форми Form1………………..2

3.16 Створення форми видалення запису в dataGridView1……………………….2

3.17 Розробка форм додавання даних в таблиці……………………………………2

3.17.1 Розробка форми додавання даних в таблицю «Свердловини»…………….2

3.17.3 Розробка форми додавання даних в таблицю «Назва породи»…………….2

3.18 Програмування події кліка на кнопці «Додати ...» з групи «Команди»…….2

3.19 Програмування події кліка на кнопці «Видалити» з групи «Команди»…….2

3.20 Програмування кліка по командах меню…………………………………….2

3.21 Запуск програми на виконання………………………………………………..2

**Розділ 5** Методика роботи користувача з програмною системою

ВИСНОВКИ………………………………………...………………………............55

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ…………………………….…..........57

ДОДАТОК А Лістинг коду програми

**ВСТУП**

Сучасні інформаційні технології – це методи й засобу для збору, зберігання, обробки й одержання інформації на основі сучасних засобів обчислювальної техніки.

Складовими частинами будь-якої інформаційної системи є БД і застосування для обробки даних. Поява персональних машин, сімейства операційних систем Windows і різного програмного забезпечення дозволяють автоматизувати ручні операції, вести будь-які види робіт з накопичення інформації, її обробки й одержанню різних вихідних форм.

У дипломній роботі ставиться завдання – розробити програмний додаток для роботи з інформаційною базою даних за допомогою мови C#, який буде обробляти та зберігати геологічну інформацію відповідно до сучасних вимог. Розробити застосування, що дозволяє здійснювати введення геологічної інформації з різних джерел через програмний інтерфейс користувача для виконання маніпуляцій щодо обчислень та виводу інформації у зручний для користувача вигляд. Додаток повинен бути спроектований з урахуванням реалізації запитів різного типу.

Цілями створення програмного додатку для роботи з інформаційною базою даних є:

1. Ефективна структуризація інформації, що дозволяє заощадити час і гроші.

2. Виключення або зведення до мінімуму повторюваних даних шляхом задання ефективної структури БД.

3. Забезпечення всім користувачам швидкого доступу до інформації бази даних.

4. Забезпечення розширення бази новими даними.

5. Забезпечення цілісності даних.

6. Запобігання несанкціонованого доступу до даних.

7. Полегшення створення застосувань, призначених для введення, редагування, виводу геологічних даних, а також ведення звітності.

Реалізація всіх перерахованих вище завдань повинна покладатись на систему керування базами даних.

Мета роботи: створення програмного додатку для зберігання та обробки геологічної інформації за допомогою високорівневої мови програмування С#.

Актуальність роботи: надійне зберігання та маніпулювання геологічною інформацією дозволяє покращити продуктивність під час обчислень та зменшити час на проведення механічних операцій.

Предмет дослідження: покращення стандартних методів введення, зберігання та обробки геологічної інформації з використанням високорівневої мови програмування C#.

Об’єкт дослідження: використання мови програмування C# для створення програмного додатку до інформаціїної бази даних .

**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

|  |  |
| --- | --- |
| **Скорочення** | **Розшифровка** |
| SQL | Structured Query Language |
| C# | C Sharp |
| DG | DataGrid |
| БД | База даних |
| ІС | Інформаційна система |

**РОЗДІЛ 1**

**ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБЛАСТІ ДОСЛІДЖЕНЬ**

Система управління базами даних (СКБД) - програмне забезпечення, яке здійснює управлінням і створенням баз даних. Популярними СУБД є MS SQL Server, Oracle Database, MySQL. В ході практичних занять вивчалися перші два програмних комплексу, перший з яких в результаті було обрано для реалізації програми.

Microsoft SQL Server - система керування базами даних (СКБД), розроблена корпорацією Microsoft. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства.

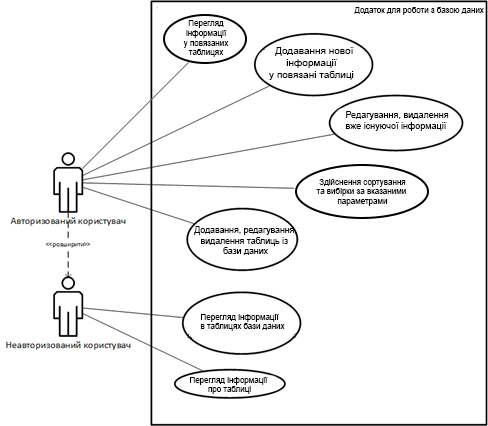
Мова високого рівня C # дозволяє реалізувати широкі можливості для взаємодії з базами даних MS SQL Server без необхідності підключення сторонніх бібліотек і розширень.

* 1. **Функціональні вимоги до додатку**

Розроблюваний додаток повинен відповідати сучасним стандартам та вимогам користувачів. В рамках виконання дипломної роботи було отримано дані по свердловинам родовища Балка Широка. Дані про свердловини…. Сучасний додаток для маніпулювання даними такого характеру має бути інтуїтивно зрозумілим для користувача та містити в собі стек технологій для виконання великої кількості операцій.

Функціональними вимогами до додатку, який буду здійснювати маніпулювання геологічної інформацією є:

* Можливість здійснювати підключення бази даних MSQL Server;
* Маніпулюваннями повязаними таблицями бази даних
* Можливість авторизації користувача
* Додаток має підримувати режим обмеженої функціональності для неавторизаваних користувачів
* Перегляд та редагування основої інформації у всіх таблицях баз даних
* Експорт обробленої інформації у зручний для користувача формат.
* Додаток повинен підтримувати режим фільтрації даних за відповідними критеріями.



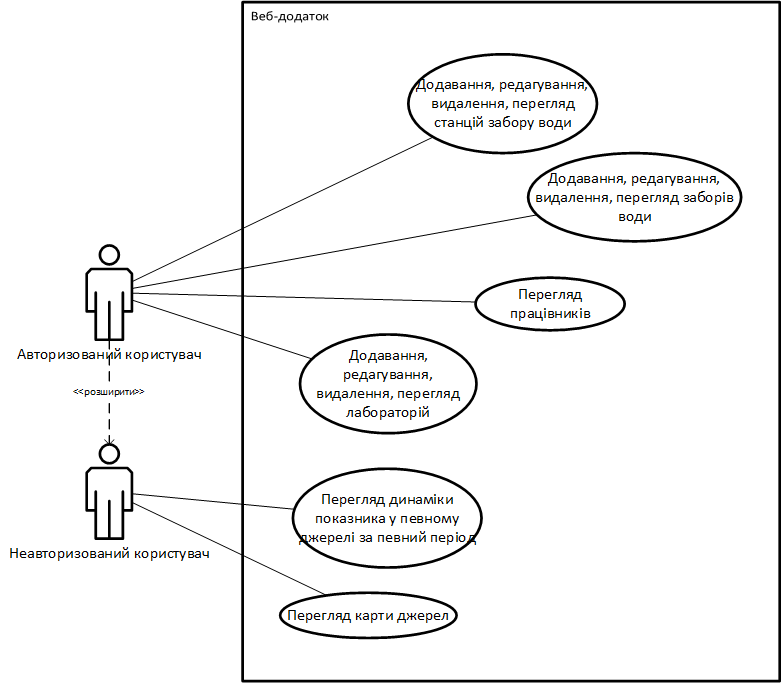


Рисунок 1.1.1 – Діаграма прецедентів для опису сценарію роботи web-додатку

Функції програми:

- висновок меню функцій, які виконуються програмою,

- введення, перегляд, редагування даних в базі даних (БД), що містить не менше двох взаємопов'язаних таблиць,

- обчислення значень полів в запитах,

- доповнення БД, видалення даних з БД,

- відбір (фільтрація) даних за умовою, що задається користувачем,

- сортування даних не менше ніж по двох полях,

- формування не менше двох звітів за даними з БД (з угрупованням, що обчислюються полями і обчисленням підсумкових даних), які можуть бути виведені на принтер,

- побудова на основі інформації з БД не менше двох графіків (гістограми, полігону або кругової діаграми), що характеризують задану предметну область,

- висновок форми з відомостями про програму (призначення, розробник).

**1.2 Концептуальна модель бази даних**

Створення цього додатку направлене на пониження складності пошуку та обробки геологічних даних на підприємстві. Базовими даними для бази даних є дані свердловин з родовища Балка Широка. Для того, щоб створити ефективну структури бази даних потрібно виділити основні відношення у яких ця інформація буде міститись. Ефективна структуризація інформації дозволяє заощадити час і гроші. Для створення такої структури потрібно опрацювати отримані дані та створити декілька взаємноповязаних відношень.

Відношення «Кора вивітрювання» зберігає у собі загальну інформацію про номер свердловини, координати, опис порід, зони та інформації про наявність корисних копалин.

Відношення «Свердлвини» містить інформацію про свердловину, а саме глибину, тип проведеного каротажу…….

Відношення «Містить інформацію» про зони…..

Дана структура бази даних розроблена для демонстрації можливостей додатку.

Така структура бази даних дозволить виключити або звести до мінімуму повторювання даних.

Зв’язки, що використовуються у відношеннях – «один-до-багатьох» та «один-до-одного».

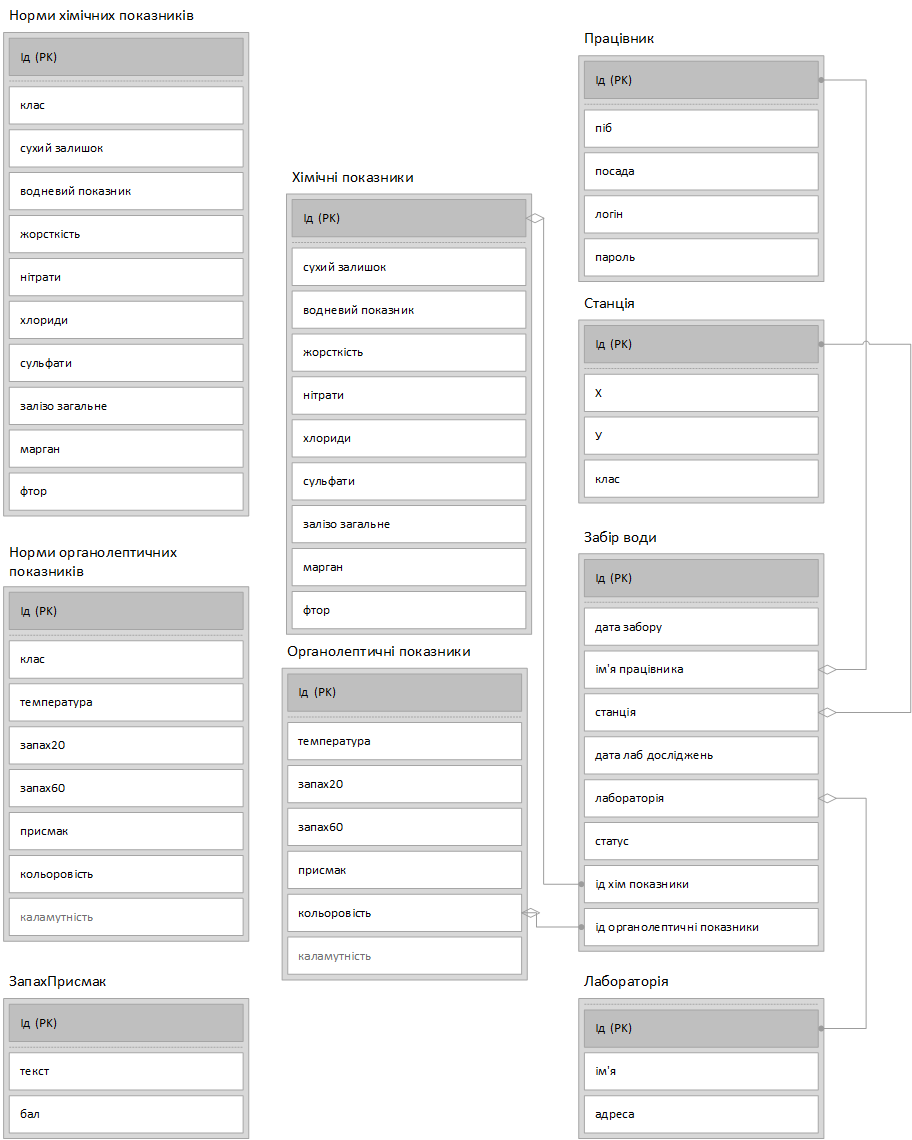


Рисунок 4.3.1 – Концептуальна схема бази даних

У роботі потрібно виконати такі завдання:

* спроектувати та реалізувати локальну базу даних типу Microsoft SQL Server, яка розміщується в окремому файлі. Ім’я бази даних “MyDataBase.mdf”;
* створити в базі даних 2 таблиці з іменами “Source” (Джерело) та “Emission” (Викиди). Кожна з таблиць повинна мати задані поля, які описуються нижче. Таблиці бази даних мають бути зв’язані між собою за деяким полем;
* розробити додаток, що оперує базою даних “MyDataBase.mdf”. Додаток реалізувати на мові C# за шаблоном Windows Forms Application;
* у додатку реалізувати основні команди оперування записами (даними) в базі даних: додавання, редагування, видалення, перегляд;
* реалізувати наступні обчислення: знаходження мінімальних, максимальних, середніх викидів для кожного джерела.

База даних містить 2 таблиці, які мають таку структуру.

Таблиця Source (джерело викидів).

[https://www.bestprog.net/wp-content/uploads/2017/08/table_source_u-300x62.jpg](https://www.bestprog.net/wp-content/uploads/2017/08/table_source_u.jpg)

Таблиця Emission (викиди для заданого джерела).

[](https://www.bestprog.net/wp-content/uploads/2017/08/table_emission_u.jpg)

Таблиці зв’язані між собою за полем ID\_Source.

* 1. **Проектування структури додатку**

В рамках дипломної роботи основною задачею є створення додатку для маніпуляції з геологічними даними. Для створення структури додатку такого типу потрібно визначити основні модулі та підмодулі програми. Для обробки геологічної інформації програма повинна підтримувати можливість вводу, зберігання, обробки та виводу геологічної інформації. Відповідно до цих критеріїв можна виділити наступні модулі програми: модуль який відповідає за ввід інформації у програму, модуль надійного зберігання інформації та модуль експорту інформації у різні формати відповідно до вимог кінцевого користувача.

Передбачається впровадження можливості вводу інформації різними шляхами, а саме: 1) ручний ввід інформації у таблицю бази даних. 2) імпорт інформації з таблиць MS Exel. 3) імпорт інформації с таблиць/баз даних MS Access.

Відповідно до цих вимог потрібно реалізувати введення інформації у додаток із різних джерел та приведення цієї інформації до спільного типу для наступної обробки. Головне вікно програми повинно бути інтуїтивно зрозумілим для користувача та відображати основний функціонал додатку. Користувач повинен мати можливість почати працювати з додаток через створення нової бази даних або таблиці безпосередньо у додатку або зробити імпорт даних із різних джерел, таких як таблиці MS Exel або MS Access. Для забезпечення всім користувачам швидкого доступу до інформації бази даних було прийнято рішення створити вікно авторизації. Якщо користувач не авторизований, він буде автоматично авторизований, як гість і зможе тільки переглядати інформацію. Якщо користувач авторизується у системі, то йому будуть доступні усі функції додатку.

Головне вікно програми буде поділене на три частини: 1) верхня частина – меню програми, яке містить елементи навігації та первинного доступу для початку роботи з програмою 2) меню навігації по додатку 3) головна панель відображення – область, де буде здійснюватись ввід та вивід інформації користувачу. Приклад основного вікна програми ви можете побачити на рисунку 1.

ТУТ ПРИМЕР ОКНА АВТОРИЗАЦИИ



Рисунок 1 – Головне вікно додатку

На панелі меню можна побачити 4 випадаючих списки: 1) Файл; 2) Інструменти; 3) Справка; 4) Розрахувати.

Пункт меню «Файл» - це випадаючий список, який містить в собі наступні підпункти: «Відкрити», «Зберегти», «Зберегти все».

Пункт меню «Інструмент» – містить підпункти з корисними інструментами.

Пункт меню «Розрахувати» дозволить користувача викликати системні додатки безпосередньо під час роботи.

Пункт меню «Справка» дозволяє отримати інформацію, яка може знадобитись під час використування додатку.

Меню навігації, яке представлене елементом дозволяє швидко переміщатись між полями таблиць. Воно представлене кнопками «Вперед», «Назад», «Вибрати», «Оновити». Також тут відображаеться номер поля таблиці.

Під меню програми можна побачити вкладки SELECT, INSERT, UPDATE, DELETE. Вони відповідають за маніпуляцію даними по аналогії до команд SQL.

**2.1 Особливості реалізації вводу даних з MS SQL бази даних**

Для розробки додатку обрано мову програмування C#. Найкращим середовищ розробки для цієї мови на данний час є Microsoft Visual Studio.

Для організації роботи з даними при створенні проектів (програм) система Microsoft Visual Studio пропонує різні види джерел даних, а саме:

* локальна база даних Microsoft SQL Server, що розміщується в окремому “\*.mdf” – файлі.
* локальна база даних Microsoft SQL Server. У цьому випадку має бути встановлений локальний сервер, наприклад SQLEXPRESS.
* локальна база даних Microsoft Access. У цьому випадку створюється “\*.mdb” файл бази даних;
* база даних, побудована на використанні ODBC-драйвера;
* Oracle база даних.

Можна також створити власну базу даних у вигляді структур чи класів. Потім, для цієї бази даних можна створити програмний функціонал, що буде обробляти записи бази даних, здійснювати зручний вивід, конвертувати у відомі формати тощо.

Розроблена програма також буде мати можливість маніпулювати даними, які не розміщені безпосередньо вже в готовій базі даних. Програма будет мати можливість здійснювати імпорт даних з таблиць MS Exel або MS Access.

Для роботи програми використаємо локальну базу даних, яка розміщена в окремому “\*.mdb”– файлі і призначена для роботи під управлінням системи керування реляційними базами даних Microsoft SQL Server.

Для того, щоб підключити базу даних до нашого додатку потрібно скористатись

Перед безпосереднім початком роботи з базою даних потрібно підготовути папку для файлів проекту бази даних. Після цього потрібно завантажити систему візуальної розробки додатків Microsoft Visual Studio та створити новий проект на мові C# за шаблоном Windows Forms Application. Після задавання папки для бази даних можна створювати нову базу даних або підключити вже існуючу. В нашому випадку ми підключимо нашу існуючу базу даних. База даних розміщується у двох файлах:

файл “MyDataBase.mdf”;

файл “MyDataBase.ldf”.

Підключення готової, раніше створеної, бази даних до проекту здійснюється однією з команд:

в меню Tools вибрати команду “Connect to Database”;

в утиліті Server Explorer вибрати кнопку “Connect to Database”.

У результаті відкриється вікно Add Connection, в якому потрібно вибрати такі налаштування:

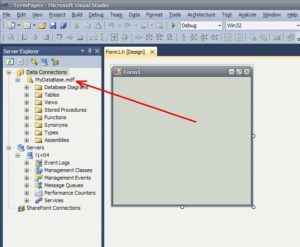
поле “Data Source” = “Microsoft SQL Server Database File”;

поле “Database File name (new of existing)” = “D:\Programs\C\_SHARP\TermPaper1\MyDataBase.mdf”. Тут з допомогою кнопки “Browse” вибирається шлях до нашої бази даних MyDataBase.mdf;

опція “Log on to the server” = “Use Windows Autentification

Після підключення (або створення) бази даних у вікні “Server Explorer” відобразиться база даних “MyDataBase.mdf”

Після підключення (або створення) бази даних у вікні “Server Explorer” відобразиться база даних “MyDataBase.mdf” (рисунок 2.2).

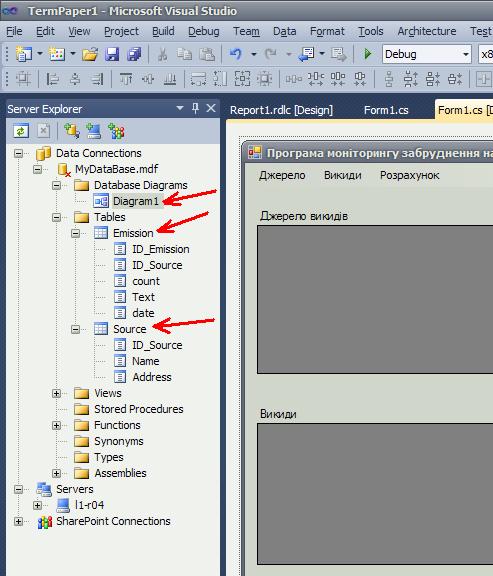
[](https://www.bestprog.net/wp-content/uploads/2017/08/02_02_00_020_02_02_.jpg)

База даних “MyDataBase.mdf” містить:

таблицю Source (рис. 2.3);

таблицю Emission (рис. 2.4);

діаграму Diagram1, що містить інформацію зв’язки між таблицями Source та Emission (рис. 2.5).



Якщо розкрити діаграму зв’язків, то відобразиться зв’язок між таблицями Source та Emission за полем ID\_Source (рис. 2.5).

Щоб відобразити діаграму зв’язків використовується команда Design Database Diagram, з контекстного меню діаграми (рис. 2.4).

Також відображення діаграми зв’язків викликається подвійним кліком мишкою на Diagram1.

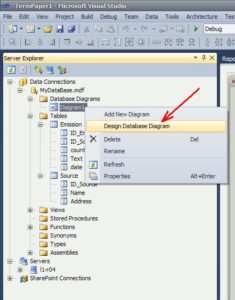


Рис. 2.4. Виклик команди відображення діаграми зв’язків між таблицями

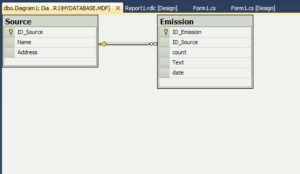


Рис. 2.5. Діаграма зв’язків між таблицями

**РОЗДІЛ 3.**

**Засоби розробки програмного додатку**

**3.1 Microsoft Visual Studio 2017**

Середовище Microsoft Visual Studio – інтегроване середовище розробки програмного забезпечення у поєднанні з іншими інструментальними засобами. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом (Windows Forms та WPF), а також веб-сайти, веб-додатки, веб-служби в керованому коді для всіх платформ, що підтримують операційні системи Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Інтегроване середовище розробки програмного забезпечення Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

**3.2 Microsoft .NET Framework**

.NET Framework – програмна платформа, випущена компанією Microsoft. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), яке використовується для різних мов програмування. Функціональні можливості CLR доступні в будь-яких мовах програмування, що використовують це середовище.

.NET є патентованою технологією корпорації Microsoft і офіційно розрахована на роботу під операційними системами сімейства Microsoft Windows, але існують незалежні проекти (перш за все це Mono і Portable.NET), що дозволяють запускати програми .NET на деяких інших операційних системах.

Основною ідеєю при розробці .NET Framework було забезпечення потреб розробника за рахунок надання йому можливості створювати додатки різних типів, які здатні виконуватися на різних типах пристроїв і в різних середовищах.

Другим принципом стала орієнтація на системи, що працюють під управлінням сімейства операційних систем Microsoft Windows [26].

Архітектура .NET Framework описана і опублікована в специфікації Common Language Infrastructure (CLI), розробленої Microsoft і затвердженої ISO і ECMA. У CLI описані типи даних .NET, формат метаданих про структуру програми, система виконання байт-коду і багато іншого.

Об'єктні класи .NET, доступні для всіх підтримуваних мов програмування, містяться в бібліотеці Framework Class Library (FCL). У FCL входять класи Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation та інші. Ядро FCL називається Base Class Library (BCL).

.NET підтримують наступні середовища розробки:

– Microsoft Visual Studio (C#, Visual Basic .NET, Managed C++, F#);

– SharpDevelop;

– MonoDevelop;

– Embarcadero RAD Studio (Delphi for .NET); ранее Borland Developer Studio (Delphi for .NET, C#);

– Zonnon;

– PascalABC.NET;

– JetBrains Rider.

Для розробленої системи необхідно мати встановлений .NET Framework версії 4.0 або вище.

**РОЗДІЛ 2**

**ПРОЕКТУВАННЯ ФОРМИ ДОДАТКУ**

Метою програми є робота з даними БД під управлінням MS SQL Server.

Користувач підключається до бази даних. Після підключення відображаються всі таблиці БД з можливістю зміни даних в них.

**2.1 Аналіз і методи вирішення задачі**

Програмний продукт повинен забезпечувати виконання таких функцій:

- можливість підключення базі даних;

- редагування даних (видалення і додавання);

- зрозумілий інтерфейс російською мовою;

- доступ до довідкової інформації про основні функції з програми;

- візуальне інформування про помилки;

- можливість користувачеві переглядати результати запитів.

**2.2 Розробка алгоритму**

При розробці алгоритму застосований метод розбиття загальної задачі на більш прості (метод декомпозиції). Основний алгоритм програми показує команди основного вікна. Схема даного алгоритму представлена нижче на малюнку 3

**2.3 Разработка пользовательского интерфейса**

Інтерфейс користувача був розроблений при використанні подієвого програмування. Видом організації взаємодії програми з користувачем є система з графічним інтерфейсом користувача (GUI), клавіатурним і маніпуляторного (введення з «миші»). Шаблон екранної форми основного вікна показаний на малюнку 4.  
  
**2.4 Написання коду програми**

Для поліпшення читаності і інформативного вихідного коду він був відкоментований, а також відформатований. Повний лістинг програми приведений в додатку А.

Програма складається з декількох форм. Перша форма це основне вікно програми, також є вікна «Про програму», «Про автора» і «Керівництво користувача», які містять довідкову інформацію. Код вікна довідки містить необхідну інформацію, закриття даних форм відбувається при натисканні на кнопку «ОК».

# Розділ 5.

# Методика роботи користувача з програмною системою

Для забезпечення безвідмовної роботи програмної системи зі збору та обробки первинних даних стану підземних вод треба дотримуватися основних вимог рекомендацій щодо її використання.

## 5.1 Системні вимоги до використання

Для використання розробленої програмної системи персональний комп’ютер має мати встановлений веб-браузер та мати доступ до мережі інтернет. Особливого значення, який браузер буде встановлено у користувача, не має, але рекомендований – Mozilla Firefox версії не менш ніж 44.

## 5.2 Опис графічного інтерфейсу системи

Після переходу за посиланням користувач бачить головну сторінку (рисунок 5.2.1) на якій розміщене меню, коротка інформація про сайт, кнопки “Register” та “Log in”. При натисканні кнопки “Log in” виконується перехід на форму авторизації для підтвердження прав на виконання певних дій у системі (рисунок 5.2.2).

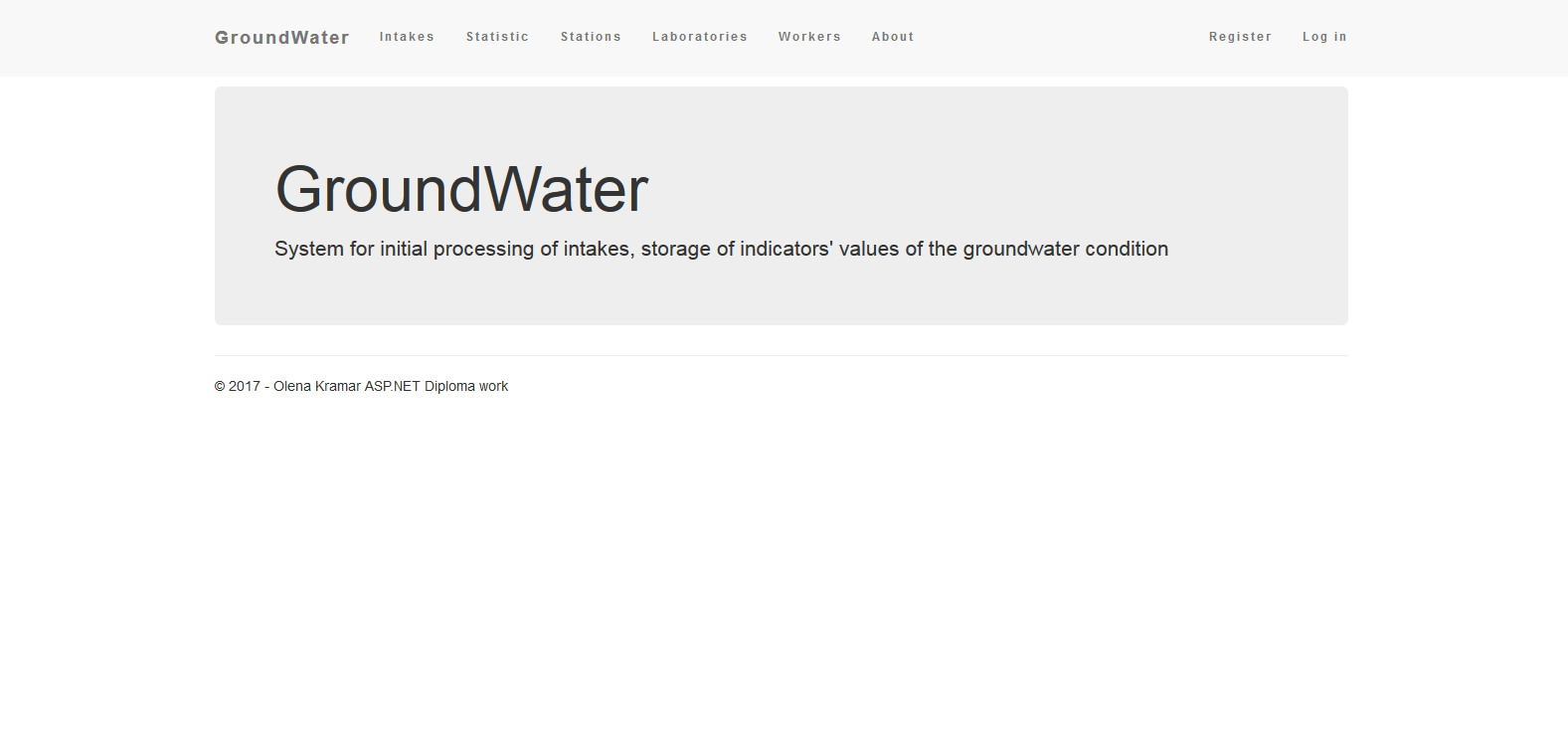


Рисунок 5.2.1 – Головна сторінка веб-додатку.

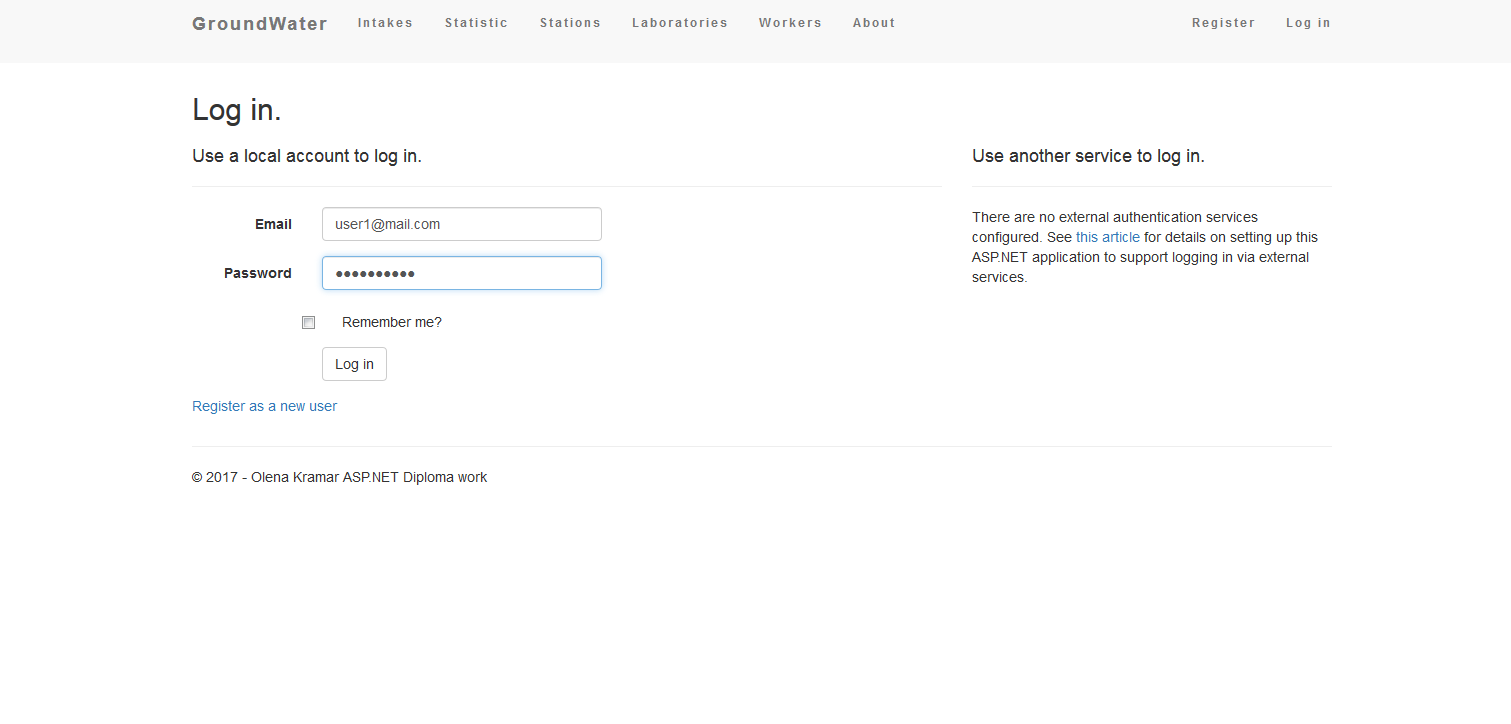


Рисунок 5.2.2 – Сторінка авторизації.

Після натискання на кнопку головного меню користувач бачить список усіх заборів води по усіх станціях та деяку інформацію про них (рисунок 5.2.3).

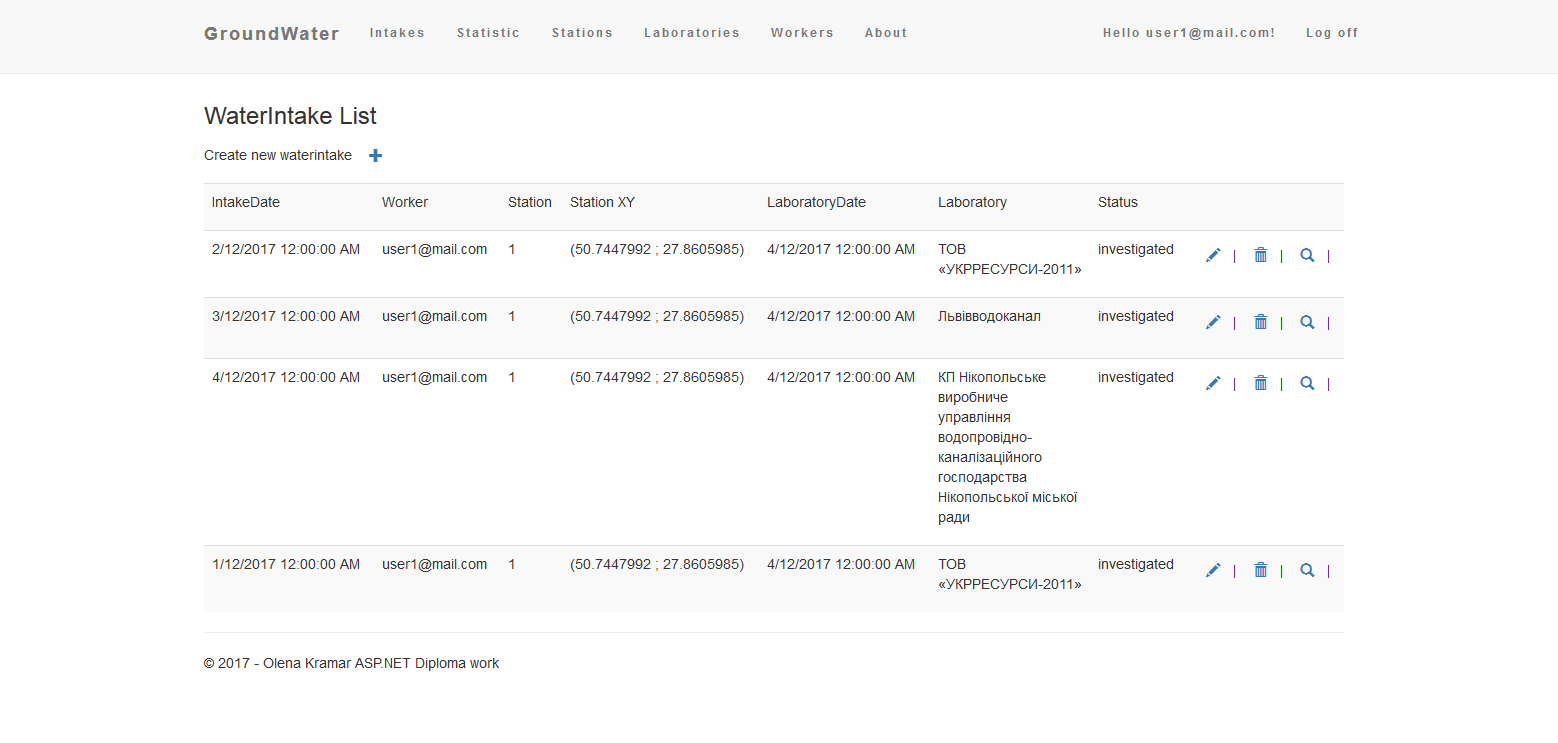


Рисунок 5.2.3 – Сторінка заборів води.

Якщо користувач хоче створити новий забір води, він має заповнити форму (рисунок 5.2.4). Щоб зменшити вірогідність введення некоректних даних або помилки користувача, система налаштована так, що:

- дата забору задається автоматично, її значення – значення дати в системі пристрою (але її можна корегувати);

- працівник, що забирав пробу зі станції, вводиться автоматично з профайлу користувача, і відповідне поле недоступне для редагування;

- поля для введення інформації про станцію та лабораторію –список з усіма існуючими записами з бази даних;

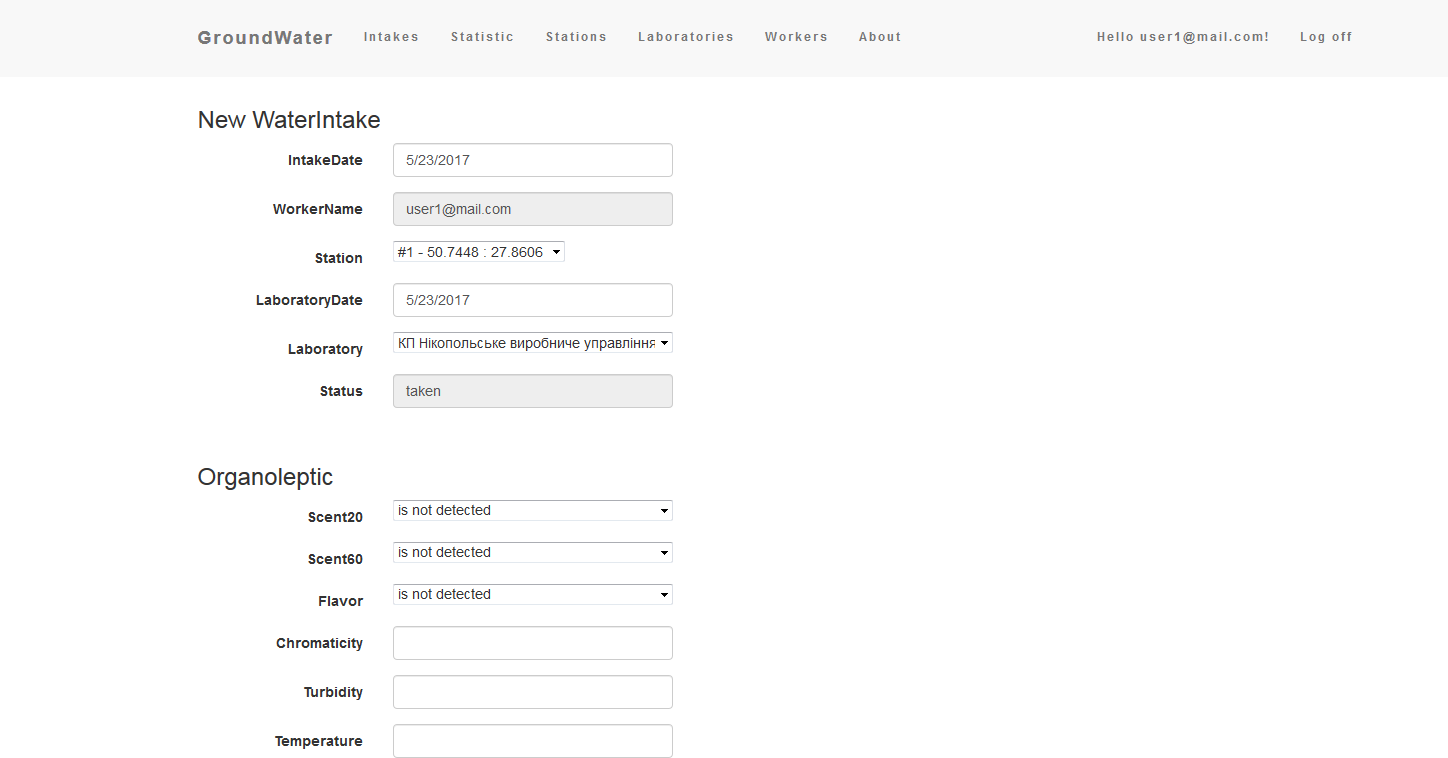
- статус зразка автоматично встановлюється на «забраний».

Рисунок 5.2.4 – Форма для додавання нового забору води.

У випадку успішної операції додавання відображується оновлений список заборів води.

Форма для редагування даних для забору води має подібний інтерфейс.

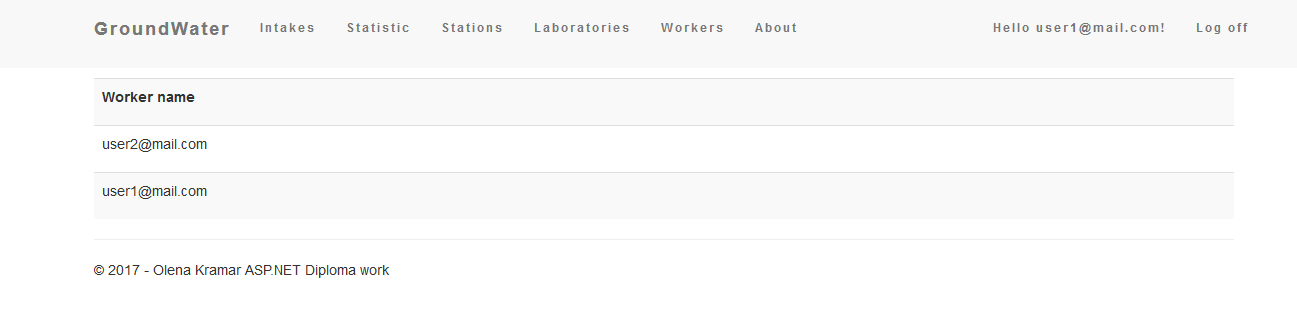
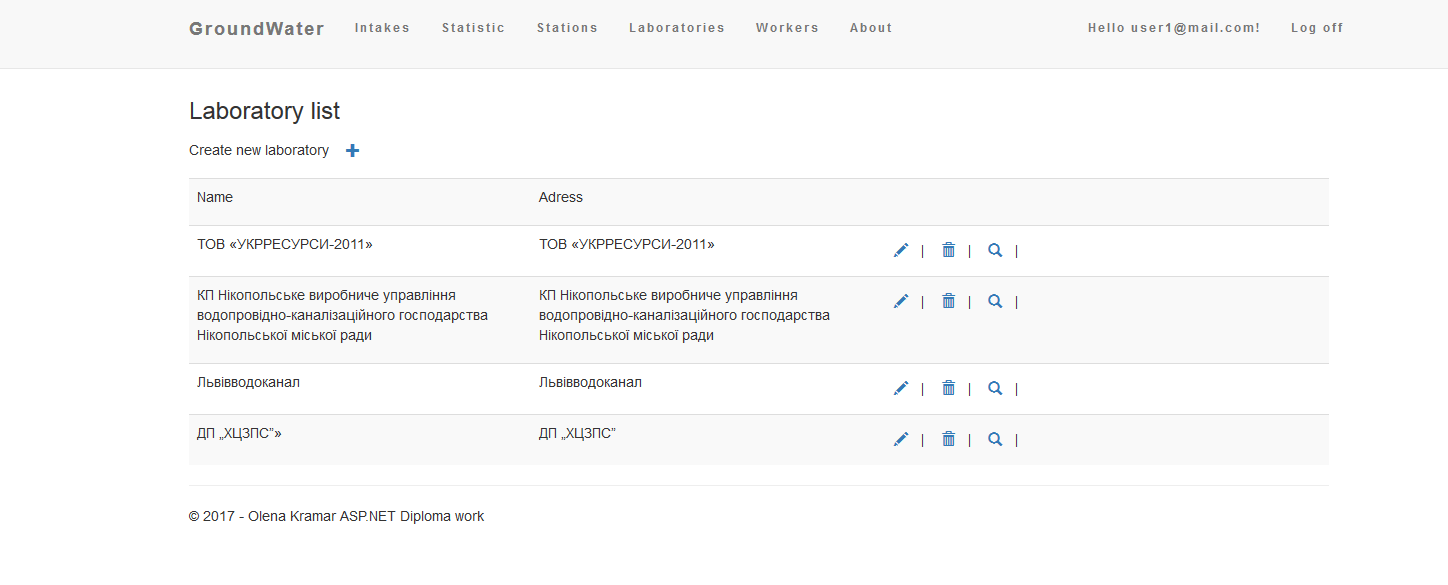
Таким чином можливо переглянути кількість і коротку інформацію про інші таблиці: працівників (рисунок 5.2.5), лабораторії (рисунок 5.2.6), станції (рисунок 5.2.7).

Рисунок 5.2.5 – Сторінка працівників.



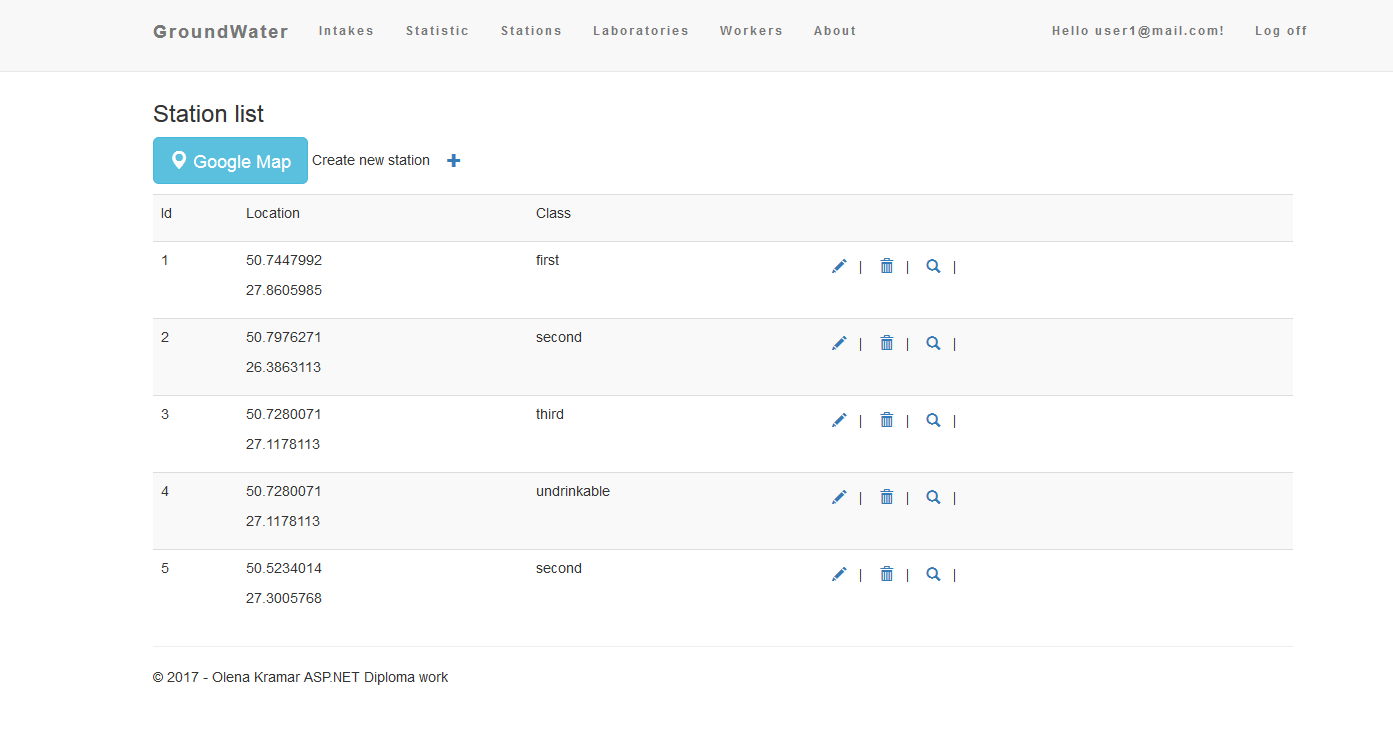
Рисунок 5.2.6 – Сторінка лабораторій для дослідження води.

Рисунок 5.2.7 – Сторінка станцій з забору води.

Також є можливість переглянути детальну інформацію про будь-який запис з таблиць у окремому вікні. Детальна інформація по запису станції забору води зображена на рисунку 5.2.8

Дана опція має схожий вигляд і для інших таблиць системи.

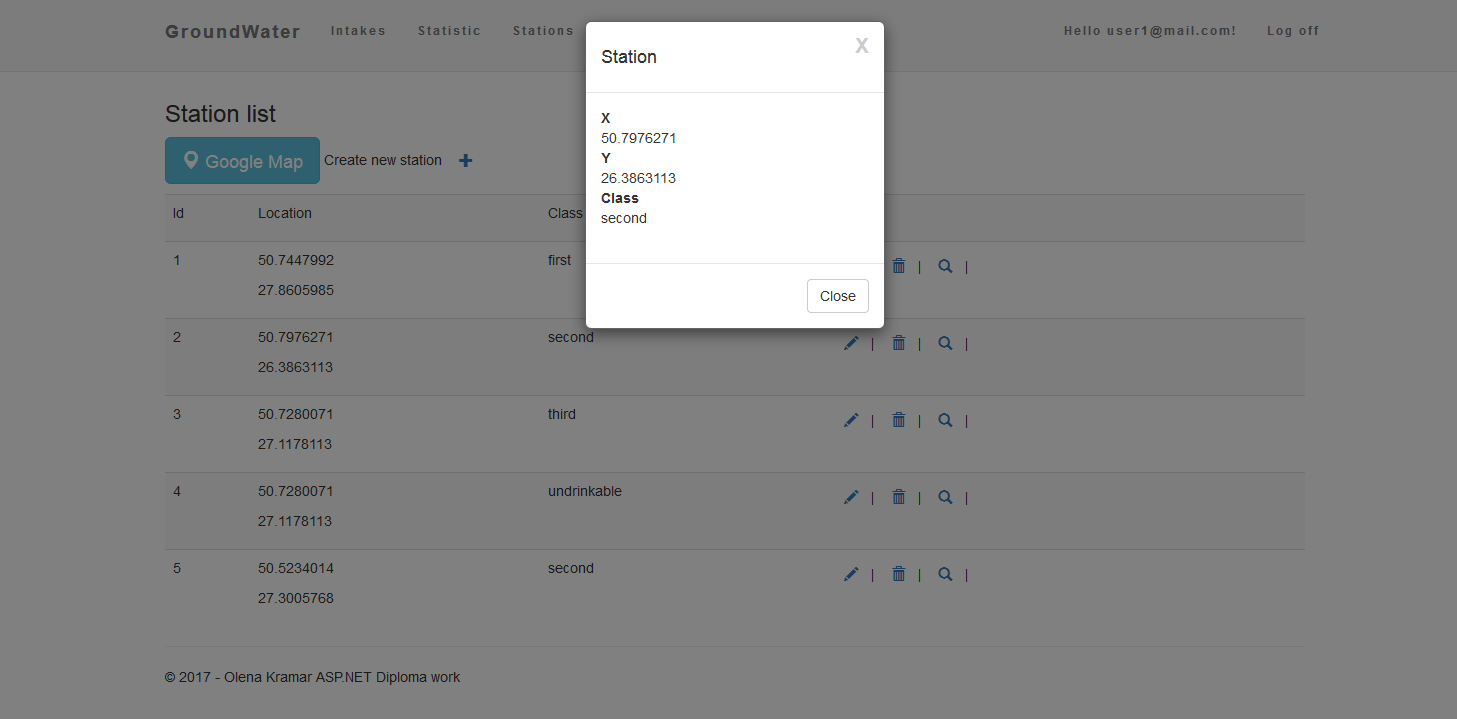


Рисунок 5.2.8–Вікно детальної інформації по запису станції забору води.

Якщо користувач не авторизується в системі, усі вище зазначені дії для нього недоступні. Такий користувач має можливість переглянути лише сторінку статистики (рисунок 5.2.9), маючи можливість дізнатись динаміку забрудненості конкретного джерела (рисунок 5.2.10) за певний період (рисунок 5.2.11) по будь-якому індикатору (рисунок 5.2.12). Результат відображається у вигляді ламаної на графіку (рисунок 5.2.13).

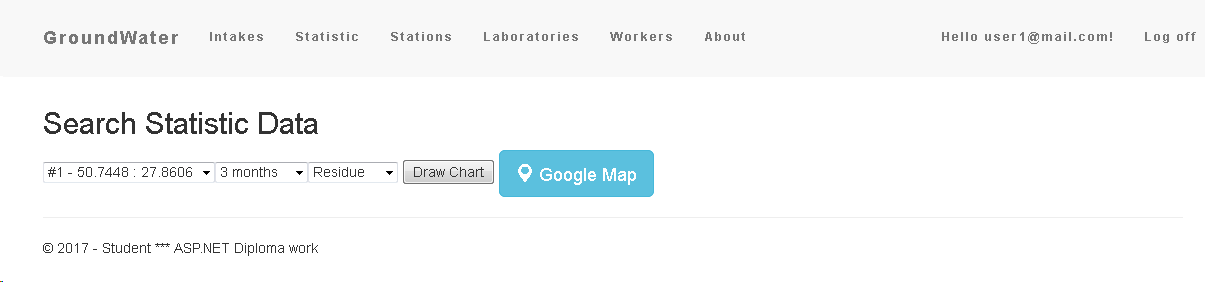


Рисунок 5.2.9 – Сторінка статистики.

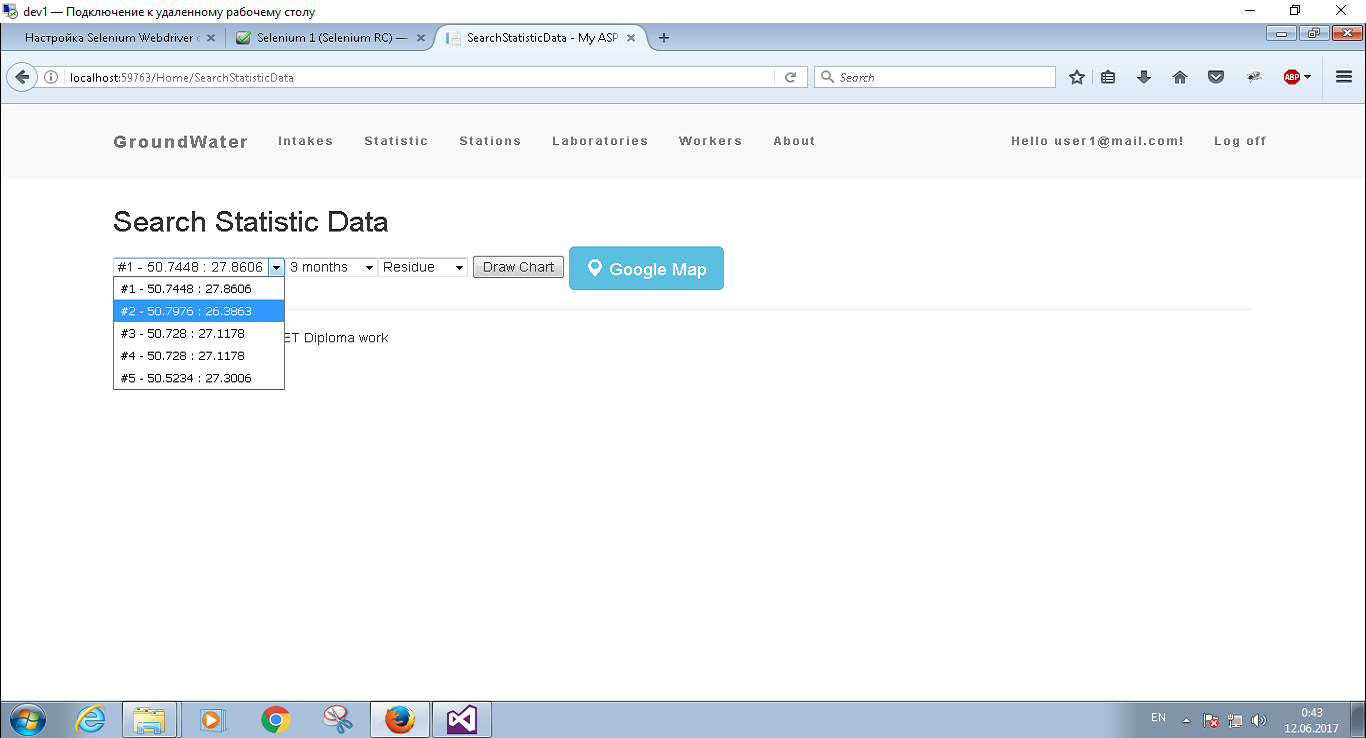


Рисунок 5.2.10 – Вибір станції забору води.

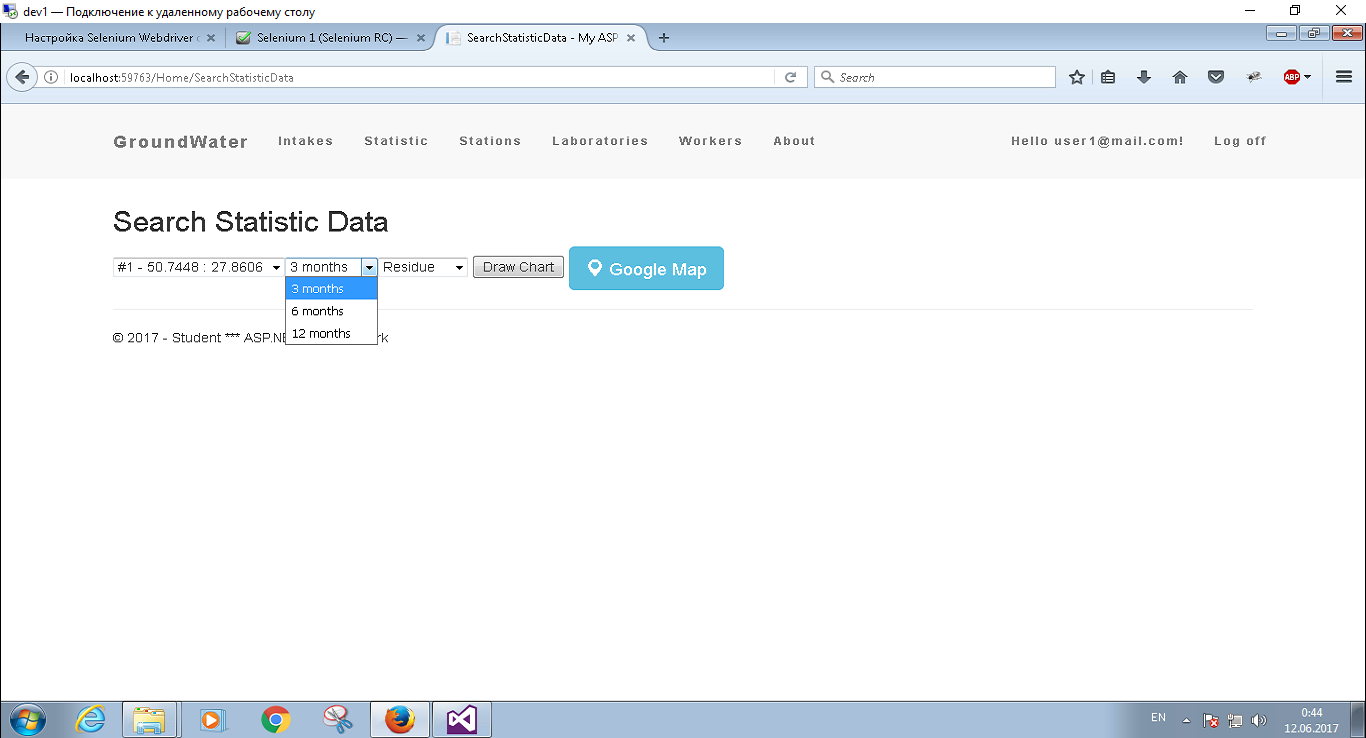


Рисунок 5.2.11 – Вибір періоду часу для відображення.

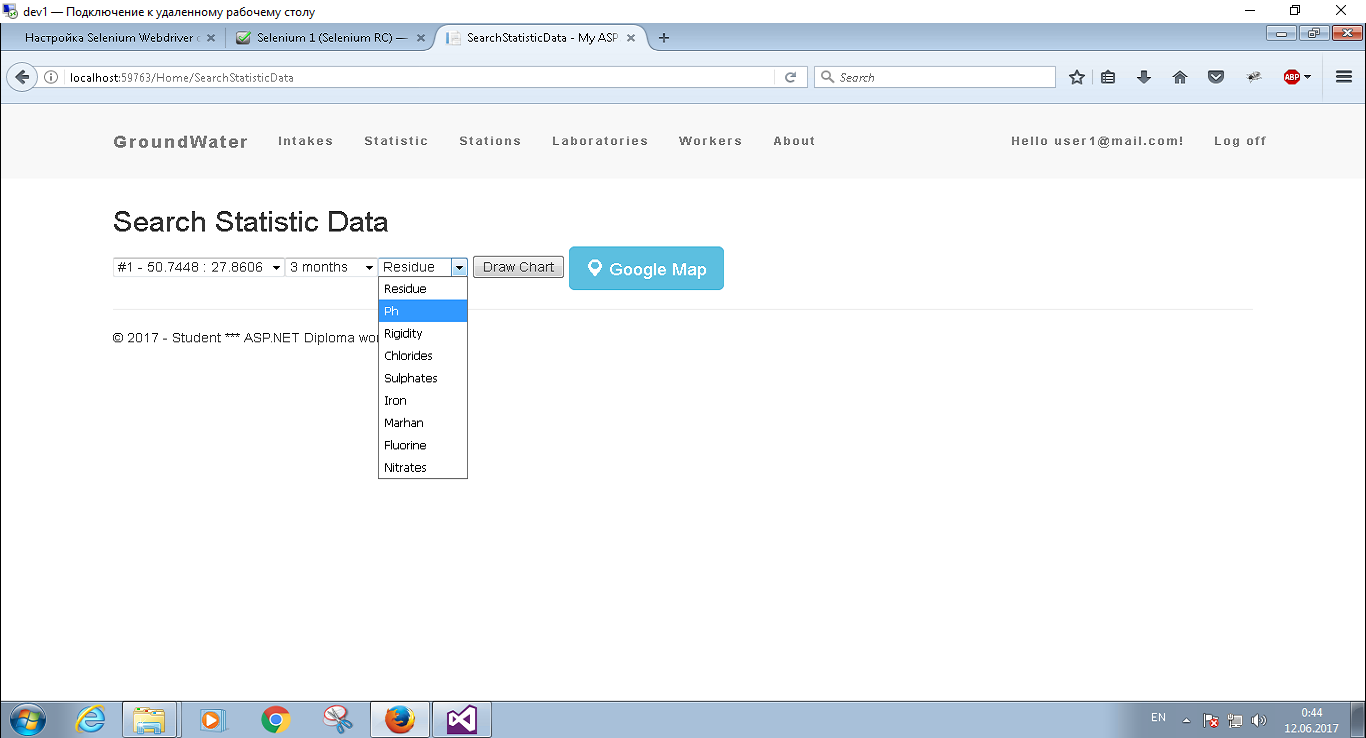


Рисунок 5.2.12 – Вибір індикатора для відображення.

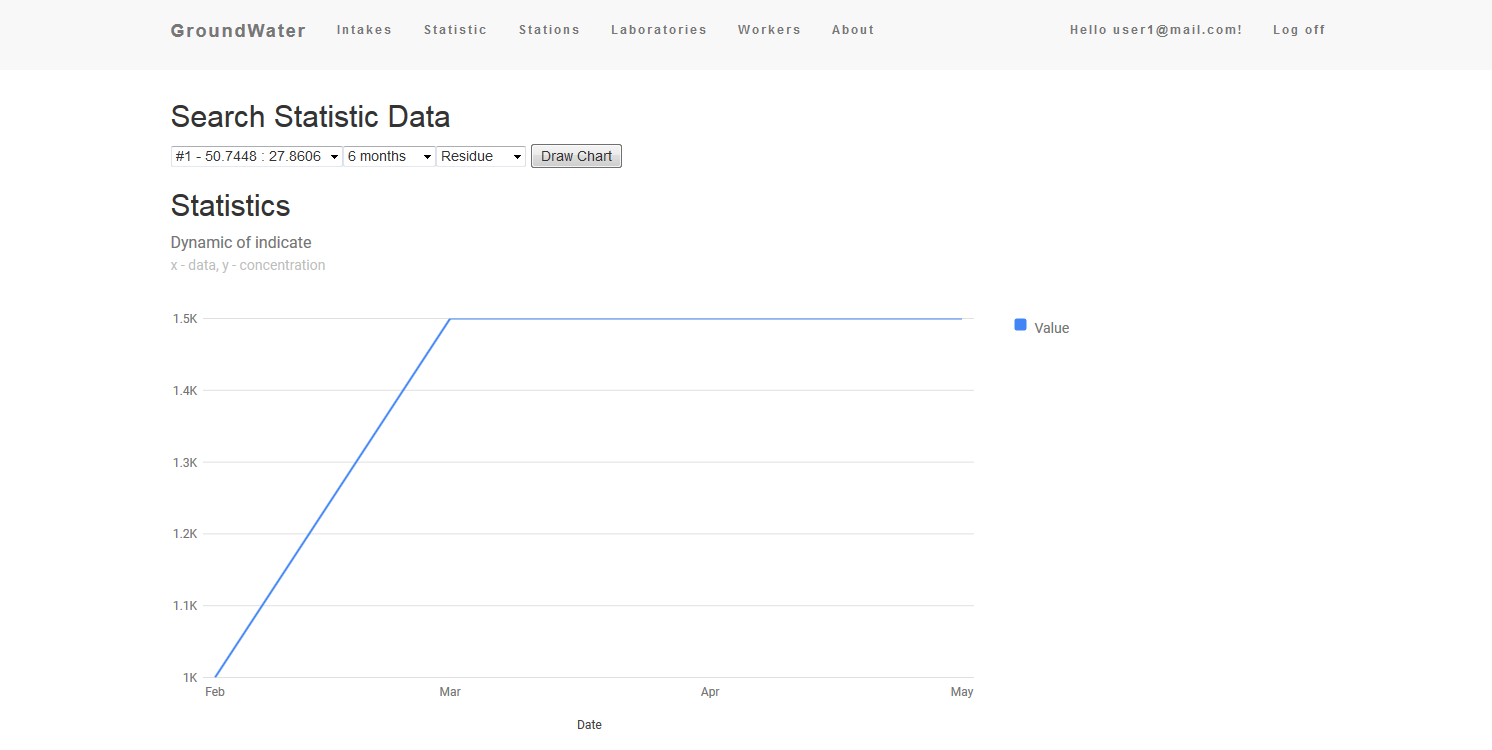


Рисунок 5.2.13 – Вигляд результатів обробки запиту.

Також користувач може переглянути карту (рисунок 5.2.14) з зображеними на ній у вигляді маркерів станцій забору води. Кожен маркер при натисканні на нього коливається, також над ним з'являється вікно з інформацією про дане джерело.

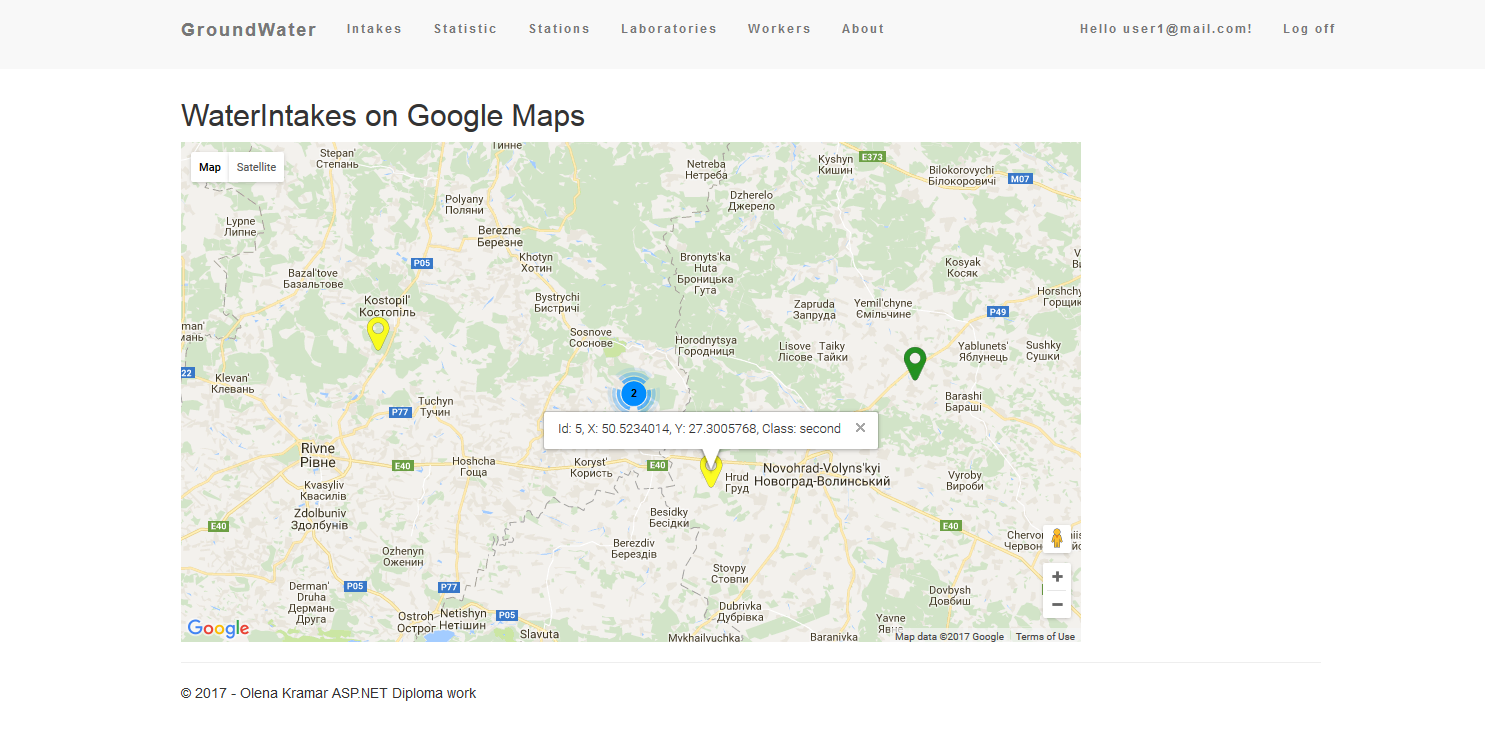
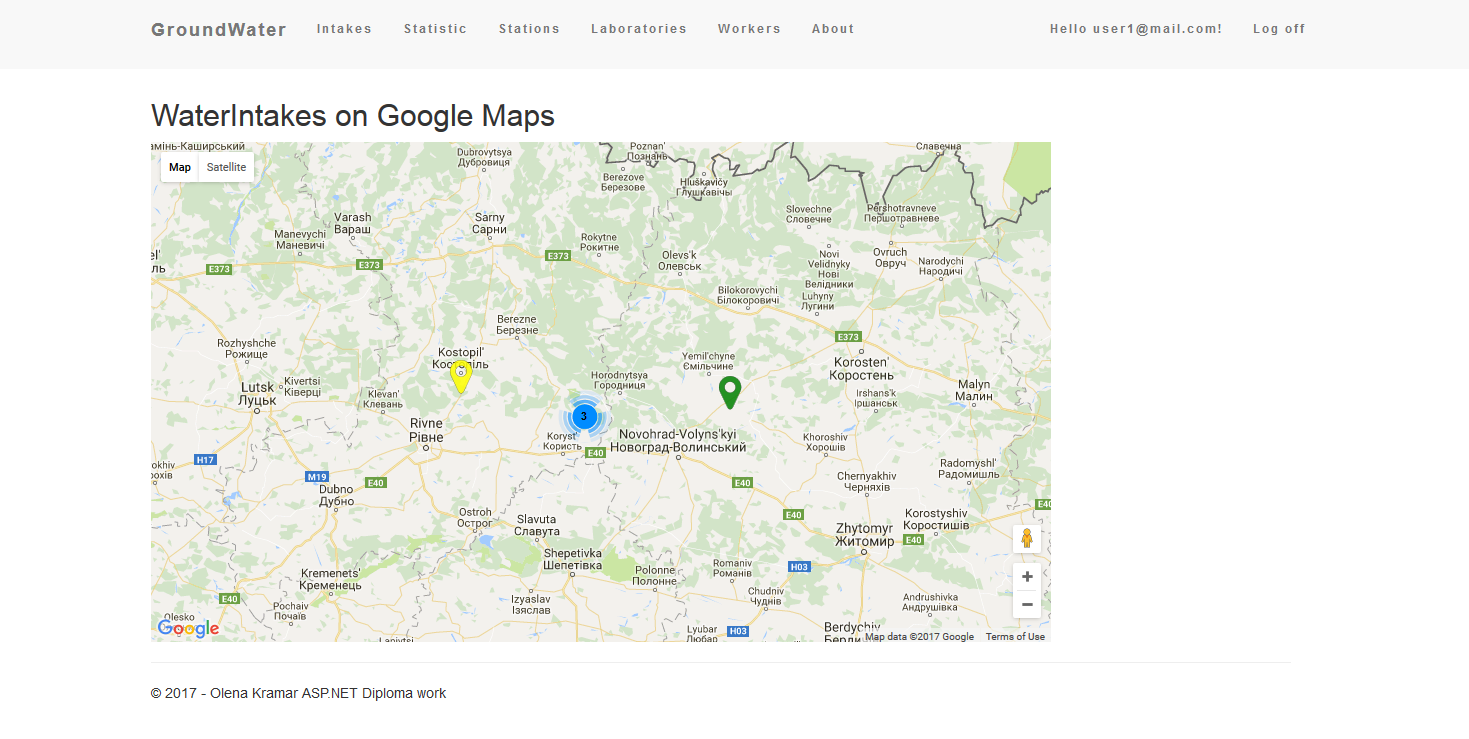


Рисунок 5.2.14 – Відображення маркерів на карті та вікна інформації.

При повторному натисканні анімація припиняється, вікно з інформацією зникає. Кожен маркер має певний колір. Він залежить від того, якого класу небезпеки це джерело. Якщо маркер зелений, то це джерело відноситься до першого класу, жовтий - до другого, червоний – до третього, чорний – джерело непридатне для використання.

В залежності від масштабу карти може з'являтися ще один вид елемента на карті – кластер. Кластер – об'єднання декількох маркерів в один елемент в даному випадку за відносною відстанню один від одного (рисунок 5.2.15).

Рисунок 5.2.15 – Кластеризація маркерів.

**ВИСНОВОК**

Проблема якості питної води, яка погіршується з року в рік через загострення екологічної обстановки, є на сьогоднішній день однією з головних. Вода є одним з найцінніших природних скарбів, адже без неї неможливе органічне життя на землі, їй належить одна з головних ролей у життєдіяльності людини. Напруженість сучасної екологічної обстановки вимагає невідкладного прийняття заходів в бік покращення якості питних вод.

Одним з першочергових кроків може стати впровадження загально державної системи збору первинних даних показників стану підземних вод в системі моніторингу водних ресурсів. Сучасна система, до якої централізовано вносяться дані, стане важливою базою для попередження, усунення екологічних проблем, дозволить контролювати викиди і виявити неоподатковані викиди. Система має інтуїтивний інтерфейс, тому не потребує особливих навичок від користувача під час роботи. Система не вимагає високих технічних характеристик пристроїв, тому може бути використана без перепон навіть на мобільних пристроях. Завдяки своїй архітектурі система дозволяє швидко і оперативно обробити велику кількість запитів від різних користувачів одночасно та не потребує особливих умов розташування – технічних станцій.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**