МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук____

Кафедра інформаційних технологій

ДИПЛОМНА РОБОТА

Рівень вищої освіти бакалавр

на тему: Розробка додатку з обробки кліматичних показників

Виконала студентка 4 курсу групи К-44
Напрям підготовки 6.050101
Компютерні науки
Яковенко Микита Володмирович
_____

Керівник асистент_____
Шуптар Наталія Йосипівна_____

Консультант к.геогр.н.,доцент___
Кузніченко Світлана Дмитрівна_

Рецензент к.ф-м.н.,доцент_____
Зінкевич Яніна Сергіївна_____

Одеса 2017

МІНІСТЕРСТВО ОСВІТИ І НАУКИУ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук
Кафедра Інформаційних технологій
Рівень вищої освіти бакалавр
Напрям підготовки 6.050101 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____
_____
"____" _____2018р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Яковенко Микита Володимирович
(прізвище, ім'я, по батькові)

1. Тема роботи «РОЗРОБКА ДОДАТКУ З ОБРОБКИ КЛІМАТИЧНИХ ПОКАЗНИКІВ»
керівник роботиасистент Шуптар Наталія Йосипівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від " _____"_____2018р
2. Строк подання студентом роботи травня 2017
3. Вихідні дані до роботи
_____
_____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз предметної області.
2. Специфікація вимог.
3. Проектні та технічні рішення.
_____
_____

5. Перелік графічного матеріалу
_____
_____
_____
_____

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання“____”_____2016 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Термін виконання етапів проекту	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Аналіз предметної області	15.03-22.03	90	
2	Специфікація вимог	22.03-15.04	90	
3	Проектні та технічні рішення	15.04-30.05	100	
4	Оформлення пояснювальної записки	30.05-10.06	100	
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)			

Студент _____ Яковенко М.В.
(підпис) (прізвище та ініціали)

Керівник проекту _____ Шуптар Н.Й.
(підпис) (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

ІС – інформаційна система
БД – база даних
СУБД – система управління базами даних
XAML - eXtensible Application Markup Language
WPF - Windows Presentation Foundation
CLR – Common Language Runtime
CLI – Common Language Intermediate
IDEF - Integrated Computer-Aided Manufacturing
DDL - Data Definition Language
DML - Data Manipulation Language
SQL - Structured Query Language

ВСТУП

В останні десятиліття зміни клімату сприяють стрімкому розвитку технологій їх моніторингу, моделювання та прогнозування. Накопичуються архіви метеорологічних даних і розроблені методики їх обробки відкривають перед дослідниками широкі можливості для вирішення кліматичних завдань різного рівня складності. Однак, різнорідність наявної метеорологічної інформації та різноманіття методів її обробки помітно ускладнюють процес аналізу і зіставлення отриманих результатів.

У зв'язку з широким розповсюдженням комп'ютерних технологій, що дозволяють здійснювати позиціонування об'єктів на земній поверхні, прикладами яких є GPS і ГЛОНАС, з'явилася унікальна можливість створювати такі комп'ютерні системи, які можна використовувати в повсякденній діяльності підприємства без значних витрат. Всім відомі такі системи відстеження автотранспортних засобів автобусних підприємств, медичних установ, а також персональні автомобільні навігатори. Всі ці системи об'єднує те, що вони усі в своїй основі мають картографічний модуль, що відображає інтерактивну карту і дозволяє будь-яким способом відстежувати заданий об'єкт.

Метою даного дипломного проекту є створення додатку з обробки кліматичних показників, який значно спростить комплексне спостереження за станом навколишнього середовища.

У загальному вигляді процес екологічного моніторингу можна представити схемою: навколишнє середовище (або конкретний об'єкт навколишнього середовища) -> вимірювання параметрів різними підсистемами моніторингу -> збір і передача інформації -> обробка і представлення даних, прогнозування. Інформація про стан навколишнього середовища, оброблена за допомогою створеного додатку, може бути використана системою управління для запобігання або усунення негативної екологічної ситуації, для оцінки несприятливих наслідків зміни стану навколишнього середовища, а також для розробки прогнозів соціально-економічного розвитку, розробки програм в області екологічного розвитку та охорони довкілля.

Даній дипломний проект містить в собі 48 сторінок, 32 рисунки, 11 по-силань та 2 листи додатків.

1 ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Характеристика об’єкта розробки

Розроблена програма представляє собою одне базове вікно, на якому здійснюється усі маніпуляції та базові настройки, функціонал програми. У верхній частині вікна можна є меню з головними налаштуваннями додатка, та реалізована можливість здійснювати управління обраними приладами. У середині вікна розташована карта місцевості, де встановлені прилади. У нижній частині екрана можна розташовано інструменти для перегляду інформації про вибрані пристрої та здійснення аналізу отриманої з датчиків інформації.

Вікно побудова "Рози вітрів" представляє із себе інструмент в якому зручно та наочно можна спостерігати за видозміною потоків вітру на обраній ділянці за певний проміжок часу.

1.2 Опис предметної області????

Ідея про необхідність систематичного збору, збереженню і переробці даних про стан навколишнього середовища остаточно сформувалася в кінці 60-х років минулого століття. Нині під поняттям "моніторинг" розуміють сукупність територіально і хронологічно організованих спостережень за компонентами біосфери.

Розроблений додаток представляє собою систему спостережень, збирання, оброблення, передавання, збереження та аналізу інформації про стан довкілля, для подальшого прогнозування його змін і розроблення науково-обгрунтованих рекомендацій для прийняття рішень про запобігання негативним змінам стану довкілля та дотримання вимог екологічної безпеки. Створена система - це відкрита інформаційна система, пріоритетами функціонування якої є спостереження за основними елементами природних екосистем для відвернення кризових змін екологічного стану довкілля і запобігання надзвичайним екологічним ситуаціям тощо[3].

Розроблена система спрямована на: підвищення рівня вивчення і знань про екологічний стан довкілля; підвищення оперативності та якості інформаційного обслуговування користувачів на всіх рівнях; підвищення якості обґрунтування природоохоронних заходів та ефективності їх здійснення.
Основними завданнями додатку з обробки кліматичних показників є: довгострокові систематичні спостереження за станом довкілля; аналіз екологічного стану довкілля та прогнозування його змін; інформаційне обслуговування користувачів екологічною інформацією [3].

1.4 Постановка задачі

Мета даної дипломної роботи – розробити інтернет-магазин подарунків ручної роботи, за допомогою якого користувач матиме можливість здійснити купівлю товару та ознайомитись зі всією необхідною інформацією про ком-панію, а також про методи оплати та доставки.
Для реалізації поставленої мети були виділені основні задачі розробки програмного продукту:
1) обрати програмні засоби, за допомогою яких буде реалізовуватись інтернет-магазин;
2) зробити аналіз існуючих аналогів та зібрати необхідні матеріали, які дозволили би створити всі необхідні умови для зручної роботи користувачів з даним програмним продуктом;
3) використовуючи методологію моделювання UML спроектувати роботу системи;
4) створити надійну функціональну оболонку програмного продукту, яка дозволяла би працювати системі без помилок, а також з максимальною швидкістю.

2ВИБІР АРХИТЕКТУРИ, ПРОГРАМНИХ ЗАСОБІВ

2.1 Вибір мови реалізації

C# —об'єктно-орієнтованамова програмуванняз безпечноюсистемою типізаціїдля платформи .NET. РозробленаАндерсом Гейлсбергом,Скотом ВілтамутомтаПітером Гольдепід егідоюMicrosoft Research(при фірмі Microsoft) [4].

СинтаксисC# близький доC++iJava.Мовамає строгу статичну типізацію, підтримуєполіморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості,винятки, коментарі у форматіXML. Перейнявши багато що від своїх попередників — мовC++,Delphi,МодулаiSmalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприкладмножинне спадкуванняклясів (на відміну від C++).

C# розроблялась як мова програмування прикладного рівня дляCLRi тому вона залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C#; подібної взаємодії слід чекати і надалі. CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо.

Специфікація C# визначає мінімальний набір бібліотек типів і класів, на який має розраховуватикомпілятор. На практиці, C# найчастіше використовується з якоюсь реалізацієюCommon Language Infrastructure(CLI), яка стандартизована як ECMA-335 Common Language Infrastructure (CLI).

2.2Вибір технологій побудови графічного інтерфейсу

Технологія WPF (Windows Presentation Foundation) є частиною системи платформи .NET і являє собою підсистему для побудови графічних інтерфейсів. Якщо при створенні традиційних додатків на основі WinForms за створення елементів управління і графіки відповідали такі частини ОС Windows, як User32 і GDI +, то додатки WPF засновані на DirectX. У цьому полягає ключова особливість рендеринга графіки в WPF: використовуючи WPF, значна частина роботи по відображенні графіки, як найпростіших кнопочок, так і складних 3D-моделей, лягати на графічний процесор відео карти, що також дозволяє скористатися апаратним прискоренням графіки[7].

Однією з важливих особливостей є використання мови декларативної розмітки інтерфейсу XAML, заснованої на XML: можливе створення насиченого графічного інтерфейсу, використовуючи або декларативне оголошення інтерфейсу, або код на керованих мовах C # і VB.NET, або поєднувати і те, і інше.

Переваги WPF:

– Використання традиційних мов .NET-платформи - C # і VB.NET для створення логіки додатка
– Можливість декларативного визначення графічного інтерфейсу за допомогою спеціальної мови розмітки XAML, заснованої на xml і представляє альтернативу програмному створенню графіки та елементів управління, а також можливість комбінувати XAML і C # / VB.NET
– Незалежність від дозволу екрану: оскільки в WPF всі елементи вимірюються в незалежних від пристрою одиницях, додатки на WPF легко масштабуються під різні екрани з різним дозволом.
– Нові можливості, яких складно було досягти в WinForms, наприклад, створення тривимірних моделей, прив'язка даних, використання таких елементів, як стилі, шаблони, теми і ін.
– Гарна взаємодія з WinForms, завдяки чому, наприклад, в додатках WPF можна використовувати традиційні елементи управління з WinForms.
– Різноманітні можливості по створенню різних додатків: це і мультимедіа, і двомірна і тривимірна графіка, і багатий набір вбудованих елементів управління, а також можливість самостійного створення нових елементів, анімації, прив'язка даних, стилі, шаблони, теми і багато іншого

– Апаратне прискорення графіки - незалежно від того, чи працюєте користувач з 2D або 3D графікою або текстом, всі компоненти програми транслюються в об'єкти, зрозумілі Direct3D, і потім візуалізуються за допомогою процесора на відеокарті, що підвищує продуктивність, робить графіку більш плавною.
– Створення додатків під безліч ОС сімейства Windows - від Windows XP до Windows 10.

2.3 Середовище проектування базами даних СУБД MS SQL SERVER

2.3.1 Зберігання даних

В даний час одним з найбільш широко поширених форматів для зберігання картографічної інформації є файли формату SHP (ESRI Shapefile). Однак вони не дуже зручні у використанні, тому що в стандарті не визначені механізми пошуку і вибірки об'єктів за певними критеріями. Хоча і самі розробники і сторонні фірми роблять спроби побудови спеціальних індексних файлів для таких цілей. Одним з таких безкоштовно розповсюджуються компонентів є SharpMap. Він дозволяє здійснювати всі необхідні дії при роботі з геоінформаційними даними, однак як показали тести об'єм споживаної пам'яті і продуктивність залишають бажати кращого[8].
Геоінформаційна система передбачає можливості зміни масштабу карти, а також переміщення і пошуку картографічних об'єктів, що викликає ряд технічних проблем, наприклад, завантаження, ще не завантажених областей, вивантаження неактуальною інформації і т.п. Попередні тести показали неприйнятну продуктивність при використанні бібліотеки SharpMap, тому було вирішено зробити імпорт картографічної інформації в СУБД. В якості СУБД на первинному етапі було вирішено використовувати СУБД - Microsoft SQL Server.

2.3.2 Призначення, загальна характеристика та об'єкти СУБД MS SQL Server

Система управління базами даних (СУБД) - сукупність програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують управління створенням і використанням баз даних.

– управління даними у зовнішній пам'яті (на дисках);
– керування даними в оперативній пам'яті з використанням дискового кешу;
– журналізація змін, резервне копіювання і відновлення бази даних після збоїв;
– підтримка мов БД (мова визначення даних, мова маніпулювання даними);

У загальних рисах, базу даних можна розглядати з двох точок зору - користувача і системи бази даних. Користувачі бачать базу даних як набір логічно пов'язаних даних, а для системи баз даних це просто послідовність байтів, які зазвичай зберігаються на диску. Хоча це два повністю різні погляди, між ними є щось спільне: система баз даних повинна надавати не тільки інтерфейс, що дозволяє користувачам створювати бази даних і витягувати або модифікувати дані, але також системні компоненти для управління збереженими даними. Тому система баз даних повинна надавати такі можливості[3]:

– фізичну незалежність даних;
– логічну незалежність даних;
– оптимізацію запитів;
– цілісність даних;
– управління паралелізмом;
– резервне копіювання і відновлення;
– безпеку баз даних.
– різноманітні інтерфейси;

Microsoft SQL Server - система керування базами даних (РСУБД), розроблена корпорацією Microsoft. Основний використовуваний мову запитів - Transact-SQL, створений спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO по структурованого мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД в цьому сегменті ринку.

SQL – мова реляційної бази даних

Мова реляційної бази даних в системі SQL Server називається Transact-SQL. Це різновид самої значимої на сьогоднішній день мови бази даних - мови SQL

(Structured Query Language - мова структурованих запитів). Походження мови SQL тісно пов'язане з проектом, званим System R, розробленого і реалізованого компанією IBM ще на початку 80-х років минулого століття. За допомогою цього проекту було продемонстровано, що, використовуючи теоретичні основи роботи Едгара Ф. Кодда, можливе створення системи реляційних баз даних[1].
На відміну від традиційних мов програмування, таких як C #, C ++ і Java, мова SQL є безліч-орієнтованим (set-oriented). Розробники мови також називають її запис-орієнтованим (record-oriented). Це означає, що в мові SQL можна запитувати дані з декількох рядків однієї або декількох таблиць, використовуючи всього лише одну інструкцію. Це одне з найбільш важливих переваг мови SQL, що дозволяє використовувати цю мову на логічно більш високому рівні, ніж традиційні мови програмування.
Іншою важливою властивістю мови SQL є її непроцедурність. Будь-яка програма, написана на процедурній мові (C #, C ++, Java), крок за кроком описує, як виконувати певне завдання. На противагу цьому, мову SQL, як і будь-яка інша непроцедурна мова, описує, що хоче користувач. Таким чином, відповідальність за знаходження відповідного способу для задоволення запиту користувача лежить на системі[9].
Мова SQL містить дві складові: мова опису даних DDL (Data Definition Language) і мову обробки даних DML (Data Manipulation Language). Інструкції мови DDL також застосовуються для опису схем таблиць баз даних. Мова DDL містить три загальні інструкції SQL: CREATE, ALTER і DROP. Ці інструкції використовуються для створення, зміни та видалення, відповідно, об'єктів баз даних, таких як бази даних, таблиці, стовпці та індекси.
На відміну від мови DDL, мова DML охоплює всі операції з маніпулювання даними. Для маніпулювання базами даних завжди застосовуються чотири загальні операції: вилучення, вставка, видалення і модифікування даних (SELECT, INSERT, DELETE, UPDATE).

2.3.3 Entity Framework - технологія роботи з базою даних

Для побудови баз даних був використаний Entity Framework. Він є продовженням технології Microsoft ActiveX Data і надає можливість роботи з базами даних через об'єктно-орієнтована код C #. Цей підхід надає ряд істотних

переваг: вам не потрібно турбуватися про код доступу до даних, вам не потрібно знати деталей роботи СУБД SQL Server і синтаксису мови запитів T-SQL, замість цього ви працюєте з таблицями бази даних як з класами C #, з полями цих таблиць - як з властивостями класів, а синтаксис SQL-запитів, який в ADO.NET раніше потрібно було вставляти в код C # у вигляді команд, замінений на більш зручний підхід з LINQ. Entity Framework бере на себе обов'язки по перетворенню коду C # в SQL-інструкції. При роботі з Entity Framework вам надаються величезні можливості по створенню моделі бази даних за допомогою інтегрованого середовища розробки (IDE) Visual Studio[6].

Модель EDM

Entity Framework акцентує свою увагу на моделюванні, в якому ви побачите багато знайомих речей - тут використовуються діаграми ER (entity-relationship, "сутність-відношення"), підхід з використанням логічного і фізичного проектування шарів і багато іншого. Ядром Entity Framework є модель EDM (Entity Data Model), суть якої полягає в зберіганні сутностей (entity) в вигляді суворо типізованих класів, а не у вигляді об'єктів схеми бази даних (показано на малюнку нижче). Модель EDM (рис. 2.1) дозволяє забезпечити зв'язок між сутнісними класами в коді і таблицями бази даних.

Шари

Архітектура Entity Framework в абстрактному сенсі заснована на шарах (layers): робочий, віддалений і сполучний.

Класи з кодом сутностей містяться в робочому шарі, в якому працюють програмісти. Залежно від того, який підхід використовується (Code-First або DB-First), робочий шар може бути змодельований або за допомогою графічного дизайнера Visual Studio, або за допомогою коду. Після цього у програмістів з'являється широкий інструментарій для роботи з Entity Framework. Синтаксис робочого шару описується за допомогою мови Conceptual Schema Definition Language (CSDL)[4].

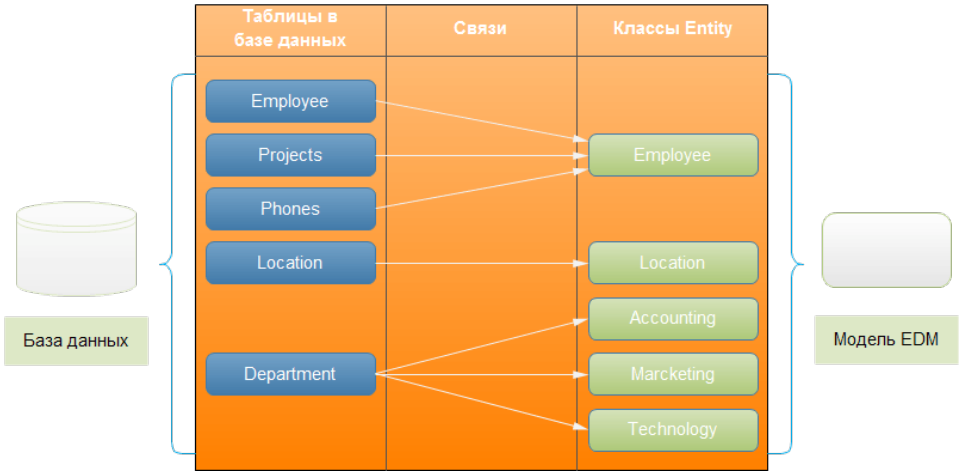


Рисунок 2.1 – Модель EDM

Віддалений шар визначає таблиці, стовпці, рядки, відносини між таблицями бази даних. Синтаксис віддаленого шару описується за допомогою мови Store Schema Definition Language (SSDL).

Сполучний шар визначає відповідність між робочим і віддаленим шарами, він пов'язує властивості сутнісного класу в робочому шарі за допомогою стовпців таблиці бази даних в віддаленому шарі. Керувати цим шаром (тобто деталями прив'язки) можна з вікна Mapping Details знаходиться в інструментах дизайнера Visual Studio або за допомогою анотацій Fluent API, якщо ви працюєте з підходом Code-First. Мова Mapping Specification Language (MSL) визначає синтаксис сполучного шару.

Важливо відзначити, що мови CSDL, SSDL і MSL мають синтаксис XML, але при цьому використовують різну семантику.

Файли Entity Framework

Всі файли, які використовуються в Entity Framework засновані на синтаксисі XML. Використання XML робить файли простими й універсальними для інших додатків. Також XML-файли читабельні для людини - ви можете в будь-який момент відкрити і переглянути вміст цих файлів. Проте, кожен елемент Entity Framework використовує різні файли XML з різним розширенням [8].

Після того як створено новий додаток, який спирається на Entity Framework і додано сутнісні класи бази даних, можна побачити результуючі файли в основній папці проекту (рис. 2.2). У середовищі Visual Studio 2012 міститься єдиний файл Entity Data Model XML (.EDMX), хоча в більш старих версіях

Visual Studio можливо буде створено декілька файлів (один для кожної сутності).

Файл EDMX містить в собі кілька секцій, написаних на мовах CSDL, SSDL і MSL, що представляють різні верстви в архітектурі Entity Framework.

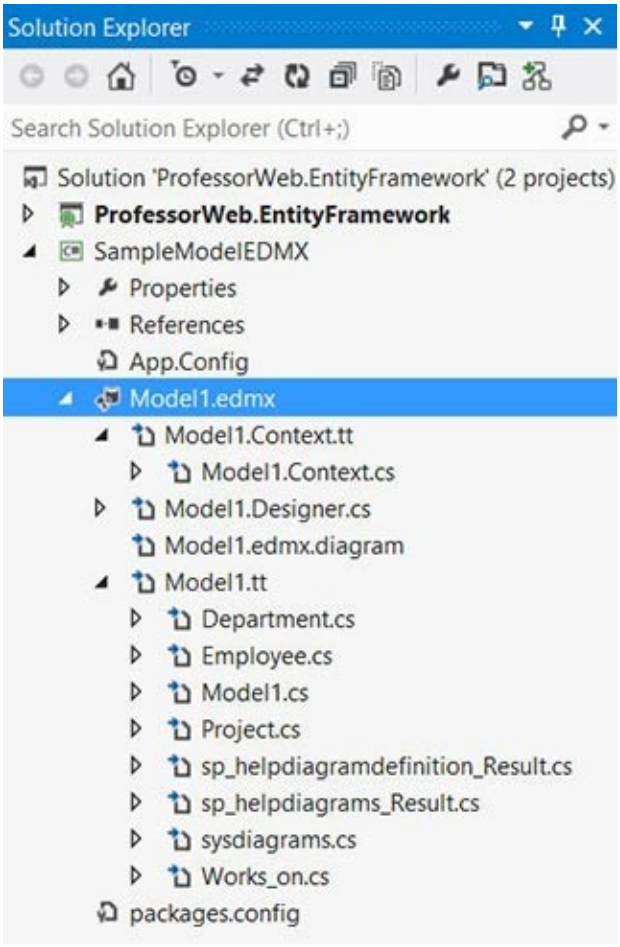


Рисунок 2.2 – Файли Entity Framework

Database-First

Підходить для проектувальників баз даних - спочатку створюється база даних за допомогою різних інструментів (наприклад, SQL Server Management Studio), а потім генерується EDMX-модель бази даних (надає зручний графічний інтерфейс для взаємодії з базою даних у вигляді діаграм і об'єктну модель у вигляді класів C #). В даному випадку потрібно працювати з SQL Server і добре знати синтаксис T-SQL, але при цьому не потрібно розбиратися в C # [1].

Model-First

Підходить для архітекторів - спочатку створюється графічна модель EDMX в Visual Studio (в фоновому режимі створюються класи C # моделі), а потім генерується на основі діаграми EDMX базу даних. При цьому підході не потрібно знати ні деталей T-SQL ні синтаксису C #.

Code-First

Підходить для програмістів - при даному підході модель EDMX взагалі не використовується і вручну налаштовуються класи C # об'єктної моделі (даний підхід підтримує як генерацію сутнісних класів з існуючої бази даних, так і створення бази даних зі створеної вручну моделі об'єктів C #). Очевидно, що це підходить для програмістів, добре знайомих з синтаксисом C #.

У проекті при роботі з Entity Framework був обраний підхід Code-First.

2.3.4 Створення бази даних Code-First

Основу функціональності Entity Framework складають класи, що знаходяться в просторі імен System.Data.Entity. Серед усього набору класів цього простору імен слід виділити наступні[8]:

- DbContext: визначає контекст даних, який використовується для взаємодії з базою даних.
- DbModelBuilder: зіставляє класи на мові C # з сутностями в базі даних.
- DbSet / DbSet <TEntity>: представляє набір сутностей, що зберігаються в базі даних.

У будь-якому додатку, що працює з БД через Entity Framework, потрібен контекст (клас похідний від DbContext) і набір даних DbSet, через який є можливість взаємодіяти з таблицями з БД. У нашій розробці таким контекстом є клас dbContext.

Для установки підключення зазвичай використовується файл конфігурації програми. У проектах для десктопних додатків файл конфігурації називається App.config, в проектах веб-додатків - web.config. Після додавання Entity Framework файл виглядає наступним чином:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<configSections>
```

```
<section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
EntityFramework, Version=6.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false" />
</configSections>
<startup>
<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
</startup>
<entityFramework>
<defaultConnectionFactory
type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
EntityFramework">
<parameters>
<parameter value="mssqllocaldb" />
</parameters>
</defaultConnectionFactory>
<providers>
<provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices,
EntityFramework.SqlServer" />
</providers>
</entityFramework>
<connectionStrings>
<add name="myMapDB" connectionString="data source=(LocalDB)\v12.0;Initial
Catalog=MapDB;Integrated Security=True;"
providerName="System.Data.SqlClient"/>
</connectionStrings>
</configuration>
```

3 ПРОЕКТУВАННЯ….

3.1Побудова модель предметної області БД«Розробка додатку з обробки кліматичних показників»

У відповідності з сучасними методологіями модель предметної області найчастіше являє собою сукупність діаграм, виконаних в якій-небудь нотації і структурованих специфікацій, характеристики ,що описують елементи моделі**.**

3.1.1 ER-модель та її призначення

Існує безліч підходів до побудови моделей предметних областей: графові моделі, семантичні мережі, модель «сутність-зв'язок» і т. д.
Найбільш популярною з них виявилася модель «сутність-зв'язок», або ER-модель (від англ. Entity-Relationship, тобто сутність-зв'язок).
На використанні різновидів ER-моделі засновано більшість сучасних підходів до проектування баз даних (головним чином, реляційних).
Різні варіанти діаграм «сутність-зв'язок» використовуються в якості інструменту семантичного моделювання, яке застосовується в реальному проектуванні структури бази даних. Семантичне моделювання являє собою моделювання структури даних, спираючись на зміст цих даних.
Перший варіант моделі «сутність-зв'язок» був запропонований в 1976 році Пітером Пін-Шен Ченом (Chen). Надалі багатьма авторами були розроблені свої варіанти подібних моделей (нотація Мартіна, нотації IDEF1X, нотації Баркера та ін). Крім того, різні програмні засоби, що реалізують одну й ту ж позначення, можуть відрізнятися своїми можливостями[4].
ER-моделювання предметної області базується на використанні графічних діаграм, що включають невелике число різнорідних компонентів.
По суті, всі варіанти діаграм «сутність-зв'язок», що виходять з однієї ідеї – малюнок завжди наочніше текстового опису. І в зв'язку зі своєю наочністю подання концептуальних схем баз даних ER-діаграми отримали широке поширення.

3.1.2 Основні поняття ER-моделі

Всі ER-діаграми використовують графічне зображення сутностей предметної області, їх властивостей (атрибутів), і взаємозв'язків між сутностями. Отже, сутності, зв'язки між ними і їх властивості (атрибути) є основними елементами опису предметної області[3].

Сутність – будь-який конкретний або абстрактний об'єкт у розглянутої предметної області. Сутності – це базові типи інформації, які зберігаються в БД. До сутностей можуть відноситься: прилади, клієнти і т. д. Кожна сутність моделі зображується у вигляді прямокутника з найменуванням (рис. 3.1).

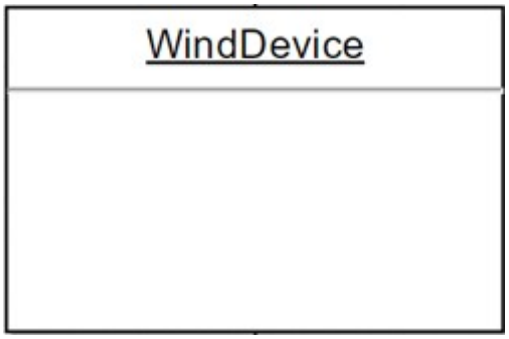


Рисунок 3.1 – Приклад зображення сутності в ER-моделювання

Необхідно розрізняти такі поняття, як тип сутності й екземпляр сутності. Поняття тип сутності відноситься до набору однорідних особистостей, предметів, подій або ідей, виступаючих як єдине ціле. Екземпляр сутності відноситься, наприклад, до конкретної особистості в наборі. Типом сутності може бути якийсь прилад, а екземпляром – анемометр, опадомір і т. д.

Атрибут – це властивість сутності предметної області. Атрибутом сутності є будь-яка деталь, що слугує для уточнення, ідентифікації, класифікації, числової характеристики чи вираження стану сутності.

Атрибути використовуються для визначення того, яка інформація повинна бути зібрана про сутності. Наприклад, для сутності студент можуть бути використані наступні атрибути: ПІБ, дата і місце народження, № залікової книжки, форма навчання і т. д.

Атрибути зображуються в межах прямокутника, що визначає сутність. Найменування атрибута повинно бути унікальним для конкретного типу сутності (рис. 3.2).

Тут також існує розходження між типом і екземпляром. Тип атрибута Місце народження, наприклад, має багато екземплярів чи значень: Харків, Київ, Одеса, Куп'янськ і т. д., однак кожному екземпляру сутності привласнюється тільки одне значення атрибута[9].

Зв'язок – асоціації (відносини) між сутностями предметної області. Зв'язки являють собою з'єднання між частинами БД.

Зв'язку дозволяють по одній сутності знаходити інші сутності, зв’язані з нею.

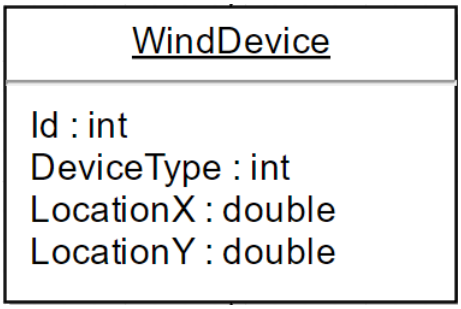


Рисунок 3.2 – Приклад зображення атрибутів сутності
в ER-моделювання

Кожен зв'язок може мати один з наступних типів (рис. 3.3):

– один-до-одного (1:1);
– один-до-багатьох (1:M);
– багато-до-багатьох (M:M).

Зв'язок один-до-одного (1:1) означає, що один екземпляр першої суті пов'язаний з одним примірником другий сутності. Зв'язок один-до-одного частіше всього свідчить про те, що насправді ми маємо лише одну сутність, неправильно розділену на дві.

Зв'язок один-до-багатьох (1:M) означає, що один екземпляр першої суті пов'язаний з декількома екземплярами другої сутності. Це найбільш часто використовуваний тип зв'язку. Наприклад, сутність ОПАДОМІР та сутність ХАРАКТЕРИСТИКИ_ОПАДОМІР мають зв'язок 1:M, так як опадомір може отримати декілька характеристик.

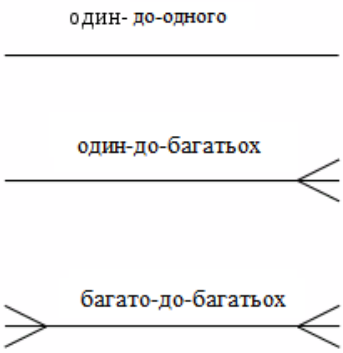


Рисунок 3.3 –Типи зв’язків в ER-моделюванні

Між зв'язок один-до-багатьох і багато-до-одного, в принципі, немає ніякої різниці, тому що між двома сутностями можливі зв'язки в обох напрямках і все залежить від того, з якими сутностями пов'язані дані.

Зв'язок багато-до-багатьох (M:M) означає, що кожен примірник пер-виття суті може бути пов'язаний з декількома екземплярами другої сутності, і кожен другий примірник суті може бути пов'язаний з кількома примірниками першої сутності. Тип зв'язку багато-до-багатьох є тимчасовим типом зв'язку, допустимим на ранніх етапах розробки моделі. Надалі цей тип зв'язку повинен бути замінений двома зв'язками типу один-до-багатьох шляхом створення проміжної сутності[10].

Кожен зв'язок має два кінця і одне або два найменування. Найменування зазвичай виражається в невизначеній формі дієслова: «мати», «належати» і т. п. Кожне з найменувань ставиться до свого кінця зв'язку (рис 3.4).

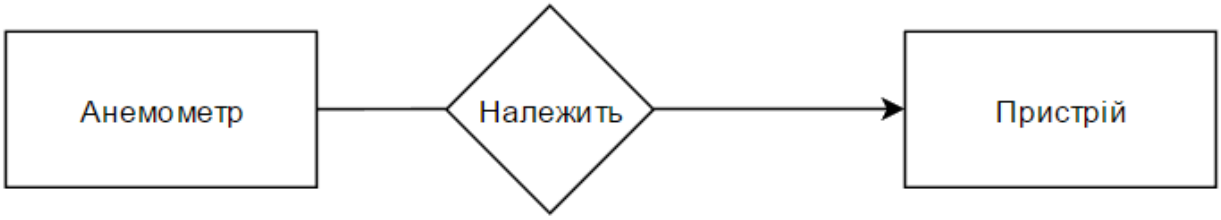


Рисунок 3.4 –Приклад моделі сутність-зв’язок

3.2 Формування картографічної продукції

Для графічного відображення метеорологічної інформації в системі був розроблений комплекс програмних засобів створення шарів ГІС та їх відображення на bing maps користувача. Комплекс програмних засобів створення шарів ГІС реалізований, як додаток з доступом до віддалених сервісів Microsoft таких як Bing Maps, виконується на стороні сервера і відповідає за різні способи генерації відображення інформації. У свою чергу на стороні клієнта відбувається прив'язка робочих об'єктів до точок геолокації.

Для відображення фізичних карт в додатку була встановлена бібліотека Bing Maps WPF Control, яка потребує реєстрації та отримання індивідуального token-a - ключа, який надає доступ до використання API на стороні сервера.
При формуванні запитів були використані стандарти класи і методи бібліотеки Microsoft.Maps.MapControl.WPF, розроблені Microsoft.
Для того, щоб отримати ключ Bing Maps Key, необхідно скористатися порталом http://www.bingmapsportal.com/ і увійтив систему за своїм обліковим записом Microsoft (Microsoft Account). Після створення ключа нижній частині сторінки "Create or view keys" порталу http://www.bingmapsportal.com/ можна буде подивитися деталі створеного ключа (рис. 3.5).

Рисунок 3.5 – Ключ Bing Maps

3.3 Предметна область і проблемне середовище додатків з обробки кліматичних показників

3.3.1 Призначення додатка з обробки кліматичних показників

У зв'язку з широким розповсюдженням комп'ютерних технологій, що дозволяють здійснювати позиціонування об'єктів на земній поверхні, прикладами яких є GPS і ГЛОНАС, з'явилася унікальна можливість створювати

такі комп'ютерні системи, які можна використовувати в повсякденній діяльності підприємства без значних витрат.

3.3.2 Дослідження предметної області

В даний час основними джерелами метеорологічний інформації, які використовуються при дослідженнях зміни клімату, є архіви даних вимірювань і моделювання метеорологічних параметрів. Різні процедури збору, зберігання і обробки даних, що застосовуються в наукових установах по всьому світу, часто призводять до взаємної несумісності програмного забезпечення і відповідних форматів файлів. Слід також зазначити, що архіви даних по навколишньому середовищу, зокрема в метеорології, характеризуються значним обсягом, що обмежує їх доступність. В результаті комплексне практичне застосування отриманих результатів досліджень, включаючи їх уніфіковане порівняння, стає доволі проблемним.

3.3.3 Основні вимоги до проекту БД досліджуваної предметної області

Основними вимогами до створеного додатку з обробки кліматичних показників є наступні:

- можливість генерування рози вітрів з отриманих кліматичних даних;
- вибір часового діапазону, за який буде проходити аналіз даних;
- введення персональних даних по кожному з доданих пристроїв;
- прив’язка пристроїв до геолокації;
- створення архіву з обробки кліматичних показників;
- перегляд інформації по кожному пристрою;

3.3.4 Сутності предметної області додатку з обробки кліматичних показників

Ґрунтуючись на аналізі предметної області і вимогах до додатку можна визначити такі основні сутності: РН-МЕТР (PrecipitateDevices), ОСАДКОМЕР (SoilDevices), АНЕМОМЕТР (WindDevices), РН-МЕТР ИНФОРМАЦИЯ (PrecipitateCharacteristics), ОСАДКОМЕР ИНФОРМАЦИЯ (SoilCharacteristics), АНЕМОМЕТР ИНФОРМАЦИЯ (WindCharacteristics), УСТРОЙСТВА (Devices).

Основні предметно-значущі атрибути сутностей можуть бути наступними. Для сутності УСТРОЙСТВА можна виділити наступні атрибути (рис. 3.6):

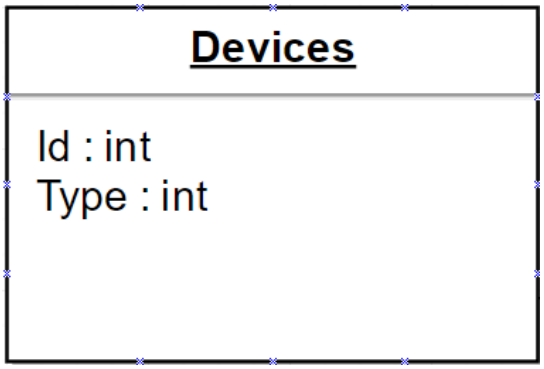


Рисунок 3.6 – Атрибути УСТРОЙСТВА

Для сутності РН-МЕТР можна виділити наступні атрибути (рис. 3.7):

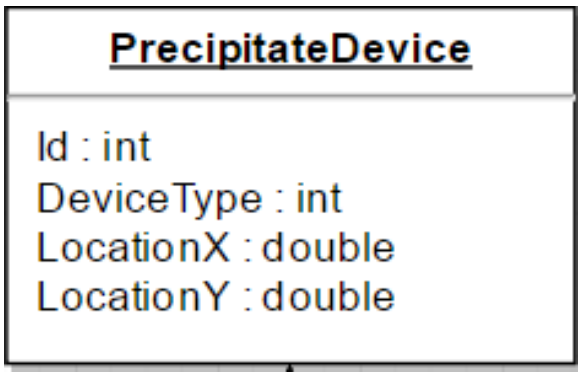


Рисунок 3.7 – Атрибути РН-МЕТР

Для сутності РН-МЕТР ИНФОРМАЦИЯможна виділити наступні атрибути (рис. 3.8):

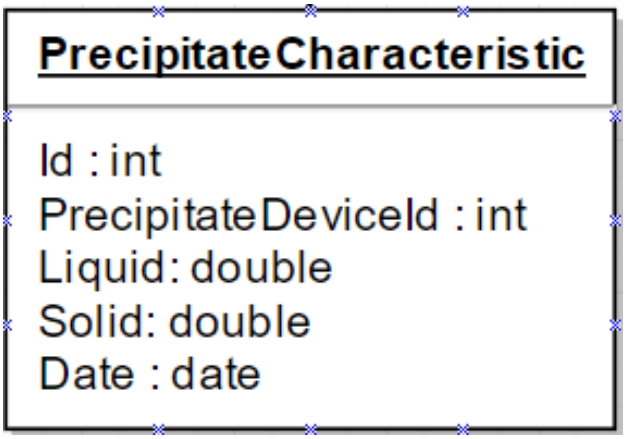


Рисунок 3.8 – Атрибути РН-МЕТР ИНФОРМАЦИЯ

Для сутності ОСАДКОМЕР можна виділити наступні атрибути (рис. 3.9):

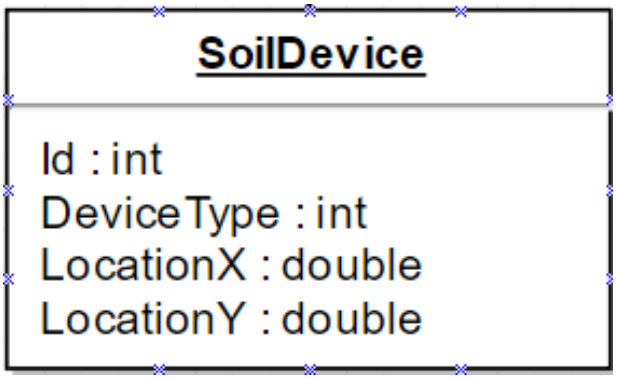


Рисунок 3.9 – Атрибути ОСАДКОМЕР

Для сутності ОСАДКОМЕР ИНФОРМАЦИЯ можна виділити наступні атрибути (рис. 3.10):

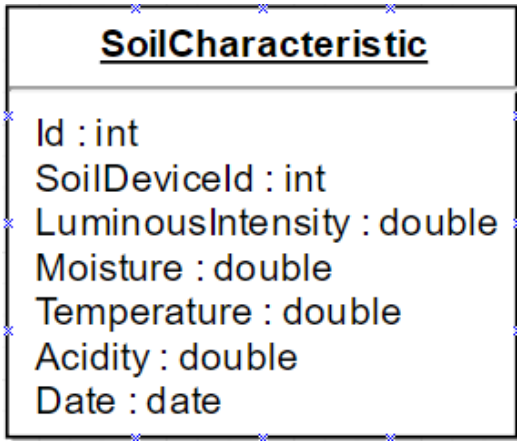


Рисунок 3.10 – Атрибути ОСАДКОМЕР ИНФОРМАЦИЯ

Для сутності АНЕМОМЕТРможна виділити наступні атрибути (рис. 3.11):

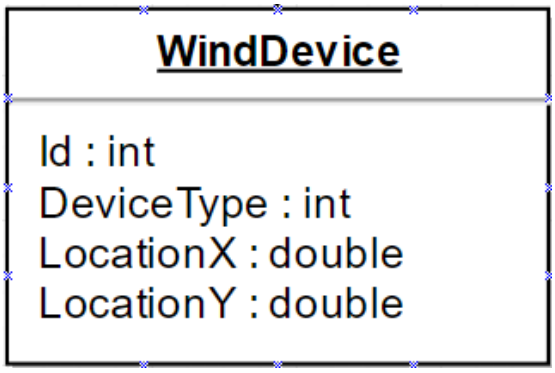


Рисунок 3.11 – Атрибути АНЕМОМЕТР

Для сутності АНЕМОМЕТР ИНФОРМАЦИЯможна виділити наступні атрибути (рис. 3.12):

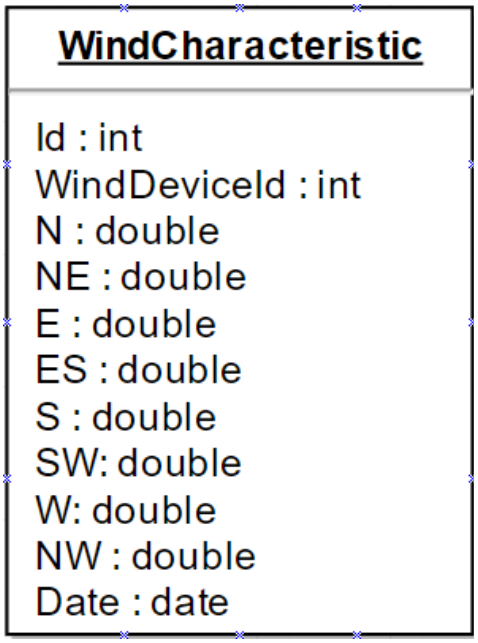


Рисунок 3.12 – Атрибути АНЕМОМЕТР ИНФОРМАЦИЯ

3.3.5 Зв'язування сутностей і побудова ER-моделі предметної області деканату

Якби призначенням бази даних було тільки зберігання окремих, не зв'язаних між собою даних, то її структура могла б бути дуже простою. Однак одне з основних вимог до організації бази даних – це забезпечення можливості

відшукання одних сутностей за значеннями інших, для чого необхідно встановити між ними певні зв'язки.

Сутності зв'язуються між собою за допомогою ключів[10].

Ключ сутності – це не надмірний набір атрибутів, значення яких в сукупності є унікальними для кожного екземпляра сутності. Ненадмірність полягає в тому, що видалення будь-якого атрибута з ключа порушується його унікальність. Сутність може мати кілька різних ключів.

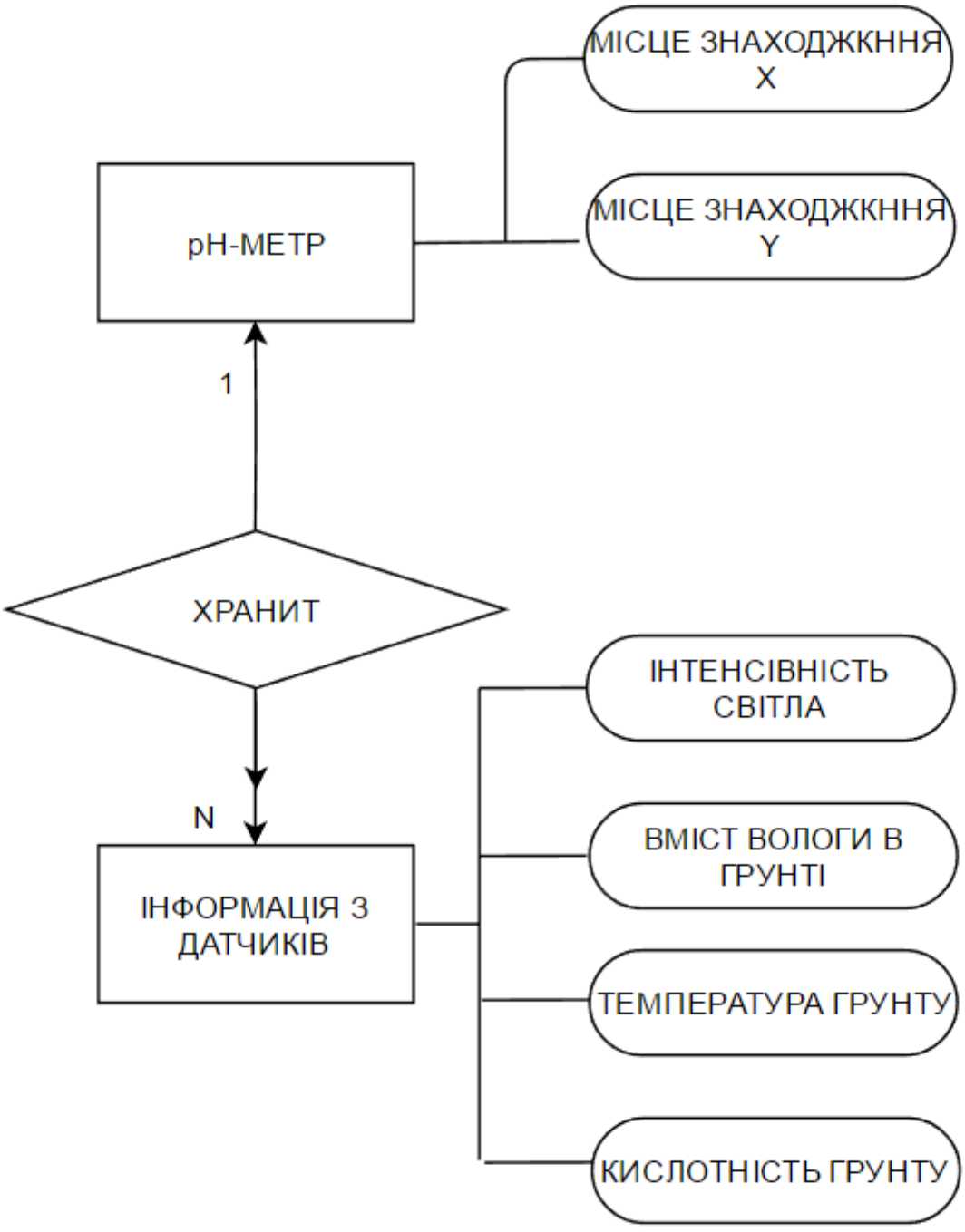


Рисунок 3.13 – Діаграма сутності РН-МЕТР та РН-МЕТР ИНФОРМАЦИЯ

Ключем для сутності РН-МЕТР є його номер, так як він унікальний для кожного пристрою.

Сутність РН-МЕТР та РН-МЕТР ИНФОРМАЦИЯ визначено як 1:М, тому що різна інформація зібрана з датчиків може зберігатися в одному пристрої (наприклад, була зібрана інформація за 21.04.2017р. та 23.04.2017р.).Аналогічній зв'язок як в РН-МЕТР та РН-МЕТР ИНФОРМАЦИЯ зберігається і для ОПАДОМІР та ОПАДОМІР ИНФОРМАЦИЯ. і АНЕМОМЕТР та АНЕМОМЕТР ІНФОРМАЦИЯ.

Побудовану мовою «сутність-зв'язок» модель предметної області легко відобразити в реляційній базі даних. Так кожна сутність буде визначена як окрема таблиця, а кожен атрибут сутності – як окрему властивість (поле), що має свої значення.

3.3.6 Представлення моделі предметної області в СУБД MS SQL SERVER

Microsoft SQL Server - система керування базами даних (РСУБД), розроблена корпорацією Microsoft. Основний використовуваний мову запитів - Transact-SQL, створений спільно Microsofti Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO по структурованого мови запитів (SQL) з розширеннями.[7]

Виходячи з побудованої ER-діаграми кожна сутність визначаєтьсяяк окрема таблиця. Таким чином, створено 7 таблиць:

1. Пристрій
2. pH-метр
3. Опадомір
4. Анемометр
5. pH-метр інформація
6. Опадомір інформація
7. Анемометр інформація

Щоб уникнути проблем зі зв'язуванням полів,таблиці будуються за правилом: спочатку головні потім підлеглі.

Головною таблицею, тобто об'єктом БД є Пристрій. Таблиця pH-метр, Опадомір та Анемометр – підлеглі щодо таблиці Пристрій, а pH-метр інформація,

Опадомір інформація та Анемометр інформація – до pH-метр, Опадомір, Анемометр відповідно.

Атрибути сутностей стали відповідно назвами полів таблиць. Таблиці в режимі конструктора наведено в додатку В.(а где додаток А??)

В даних таблицях є назви полів, типи даних, що характеризують поля і опис кожного поля. Інформація, введена в полі Опис, відображається в рядку стану таблиці при виборі поля.

Тип даних визначає, яку інформацію можна ввести в полі. У випадаючому списку передбачені наступні типи даних[3]:

1. Текстовий – використовується для полів, що містять комбінації символів і цифр (не більше 255 символів). За замовчуванням полів присвоюється цей тип даних, так як він отримав найбільш широке поширення.
2. Поле Memo – призначене для введення букв, цифр і знаків пунктуації (довгі тексти і коментарі – не більше 65535 символів). Поле цього типу не може бути ключовим.
3. Числовий – містить тільки цифрову інформацію (за винятком грошових величин), яку надалі можна використовувати для обчислень.
4. Грошовий – призначений для введення грошових величин. Стандартний шаблон для полів цього типу передбачає використання двох десяткових знаків, тобто облік копійок. Максимальне число десяткових знаків не перевищує чотирьох. Він використовується для запобігання помилок при округленні.
5. Дата/час – містить інформацію про дату і час. Дата й час зберігаються у вигляді числа, ціла частина якого представляє дату, дробова – час. Дата і час можуть виводитися в різних форматах.
6. Лічильник має формат довгого цілого (Long Integer). При додаванні нового запису значення цього поля автоматично присвоюються. Це поле можна використовувати в якості ключового.
7. Логічний – застосовується для полів, що містять значення Так або Ні. Наприклад, у такому полі можна вказати має даний співробітник дітей чи ні. В логічні поля можна записати число 0, яке інтерпретується як Брехня або 1 – мається на увазі Істина. Логічне

поле не може бути ключовим, але по ньому можна індексувати таблицю.

8. Поле об'єкта OLE – містить об'єкти з інших програм (растрові і векторні малюнки, аудіо та відео файли, електронні таблиці тощо). Це поле не може бути ключовим або індексним.

9. Гіперпосилання – призначений для зберігання адреси веб-сторінки, розташованої в Інтернеті, Інтранеті, локальної мережі або на автономному комп'ютері. Після клацання мишею на цьому полі автоматично запускається браузер. Гипрпосилання дозволяють виконувати переходи між об'єктами Microsoft Access без допомоги програмування. Для створення посилання, що відкриває об'єкт Access, введіть ім'я об'єкта в полі гіперпосилання.

10. Майстер підстановок – завантажує майстра підстановок і виводить комбіноване вікно, що дозволяє вибрати зі списку потрібне значення, наприклад, можна вибрати поле з іншої бази даних [4, 8].

11. Вибравши необхідні типи даних, і описавши поля можна перейти до складання схеми даних.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1 Загальна частина роботи

При запуску програми запускається головне вікно програми(рис. 4.1), яке містить головне меню та базові елементи управління, за допомогою яких можна почати роботу з додатком.

У головному меню програми є такі пункти:

– «Диаграмма»
– «Мод»
– «Прибор»
– «Обновить статистику»

Також розроблено елемент управління, який відповідає за графічне відображення вибраної користувачем місцевості.

У нижній частині вікна знаходиться панель інструментів, яка позволяє в наглядній та зручній формі керувати та аналізувати дані, які були внесені в програму.

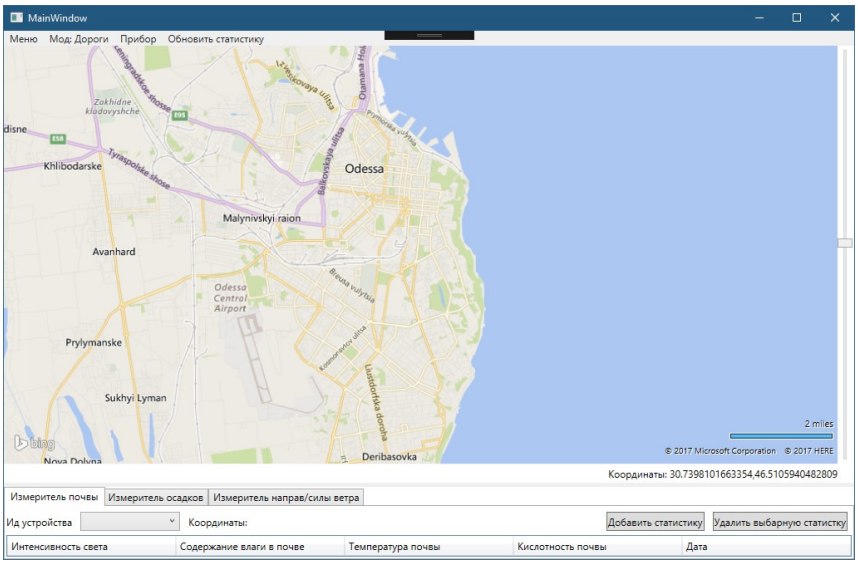


Рисунок 4.1 – Головне вікно програми

Розглянемо елементи менюбільш детально.

«Диаграмма» - в даному пункті ми можемо вибрати функцію «Роза ветров» (рис. 4.2), яка позволяє створювати діаграми.Роза вітрів - векторна діаграма, що характеризує в метеорології і кліматології режим вітру в даному місці за багаторічними спостереженнями.

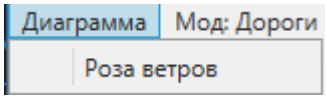


Рисунок4.2 – Меню «Диаграмма»

«Мод» - даний додаток має декілька режимів відображення місцевості (рис. 4.3). Розглянемо кожний із них:

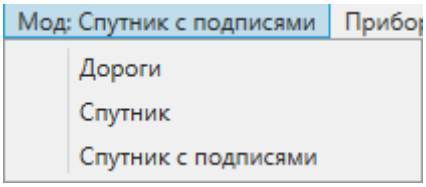


Рисунок 4.3 – Меню «Мод»

– «Дороги» (рис. 4.4). При виборі даного пункту, карти відображаються як дорожній режим у якому інформація про місцевість зведена до мінімуму і відображаються лишедороги, будівлі, парки та поля.

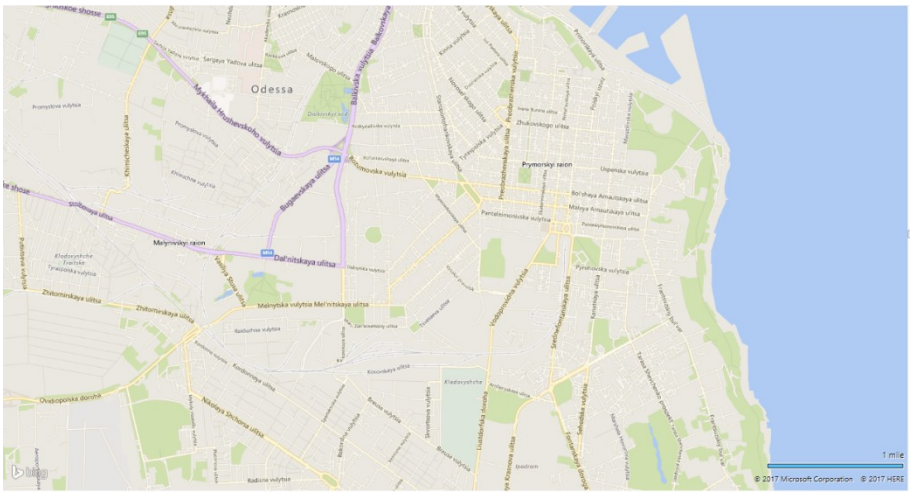


Рисунок 4.4 – «Мод: Дороги»

– «Спутник» (рис. 4.5). При виборі даного пункту, карти відображаються як знімок місцевості з супутника. На ній більш детально можна розгледити місцевість тазробити певні аналізи.

Рисунок 4.5 – «Мод: Спутник»

– «Спунтник с подписями» (рис. 4.6). При виборі даного пункту, карти відображаються як в режимі «Спутнік», проте ще додаються назви вулиць та географічних обєктів.

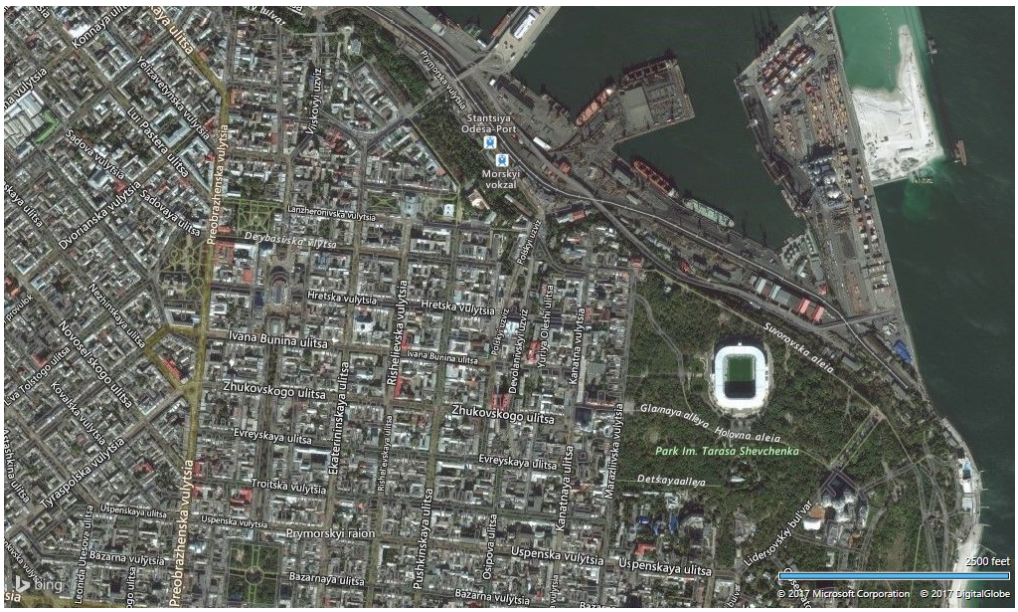


Рисунок 4.6 – «Мод: Спутник с подписями»

«Прибор» - дане меню відповідає за контроль над приборами. Тут є можливість додавати, редагувати та видаляти різні пристрої (рис. 4.7).

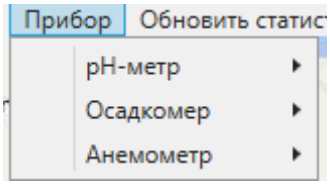


Рисунок 4.7 – «Прибор»

В програмі доступні такі прибори:

– «pH-метр»:використовується для вимірювання вологості на рівні коренів, також вимірює освітленість рослини і рівень pH. Саме останній показник дуже важливий для сприятливого росту гортензій, вересів, лохини і багатьох інших рослин, яким важливо забезпечити і підтримувати високу кислотність грунту. Ідеальний і необхідний інструмент для садівників, городників та квітникарів. Прилад дуже простий в експлуатації, а свідчення точні. Знання цих показників дозволить виростити хороший урожай і дозволить рослинам нормально розвиватися.

– «Осадкомер»:людям, які займаються цілими плантаціями, розвитком сільськогосподарських культур не обійтися без такого важливого приладу, як опадомір. Пристрій дозволяє виміряти рівень рідких, твердих опадів (дощ, град, сніг). Вперше про такі аксесуарах заговорили в 70-х рр. минулого сторіччя. За допомогою даного приладу можна з'ясувати і дати прогноз врожайності.

– «Анемометр»:прилад для вимірювання швидкості руху газів, повітря в системах, наприклад, вентиляції. У метеорології застосовується для вимірювання швидкості вітру.

При натисканні на кожен із них з’являється нове меню, де можна додати або видалити пристрій (рис. 4.8).

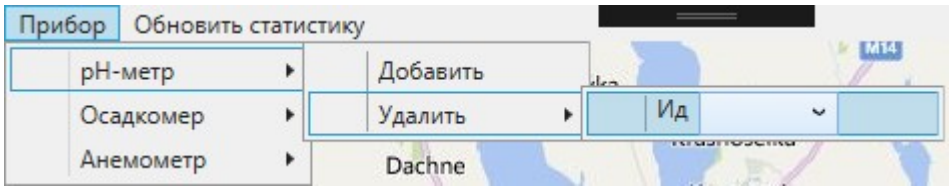


Рисунок 4.8 – «Прибор. Докладніше»

Після натискання на меню «Добавить» можна розмістити приладиу будь якому місці на карті.

Якщо треба видалити якийсь пристрій, то необхідно натиснути на меню «Удалиь» після чого указати «ID» необхідного нам прибору і буде проведено видалення (рис. 4.9).

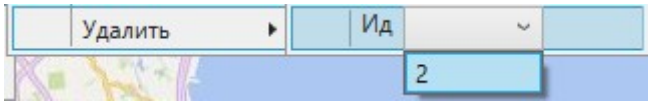


Рисунок 4.9 – «Прибор» Видалення

«Обновить статистику»

При виборі данного пункту меню відбуваєтьсяоновлення усієї зміненої інформації у додатку.

Більшу частину робочої областізаймає карта. Вона відповідає за відображення приладів та роботи з ними. Прилади мають універсальний вигляд,

що дозволяє швидко орієнтуватися на карті і знаходити необхідні. Пристрій являє собою маркер на карті з означенням властивостей – колір, назва, «ID» (рис. 4.10).

Рисунок 4.10 – Маркери приборів на карті

У нижньому правому куті виводяться координати місцевості та є можливість зміни масштабу. Ці елементи управління дозволяють більш детально відредагувати місцезнаходження пристроїв на карті (рис. 4.11).

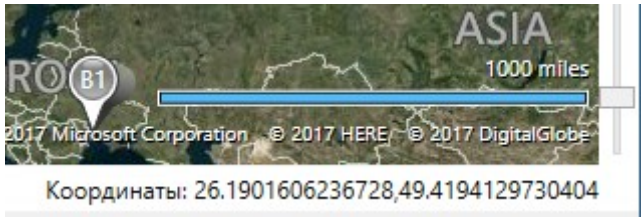


Рисунок 4.11 – Маркери приборів на карті

У нижній частині вікна можна побачити панель управління даними з трьома вкладками. Розглянемо їх більш детально.

– «pH-метр»
– «Осадкомер»
– «Анемометр»

«pH-метр»

В цих вкладказможна вести звіт та аналіз усіх отриманих даних з приладів. Тут відображається прилад, його місце знаходження, а також інформація прийнята с його датчиків (рис. 4.12).

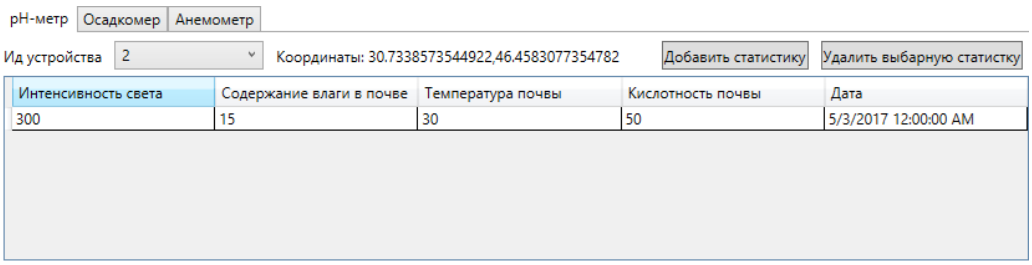


Рисунок 4.12 – «pH-метр»

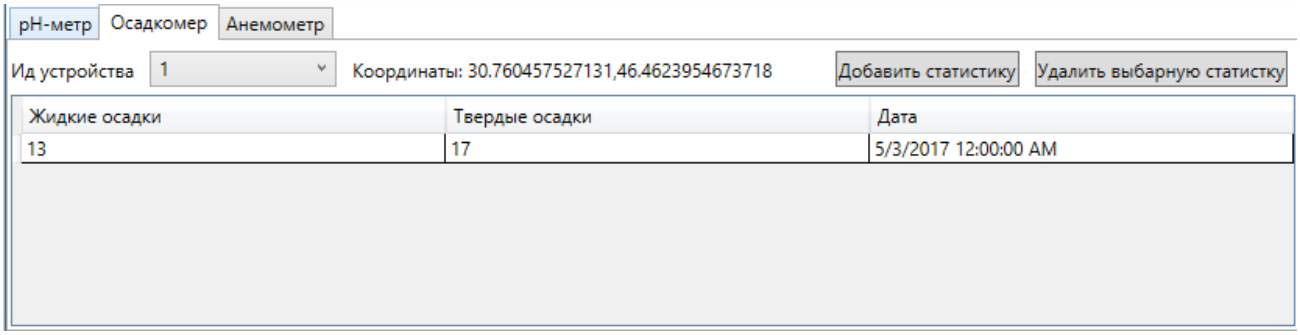


Рисунок 4.13 – «Осадкомер»

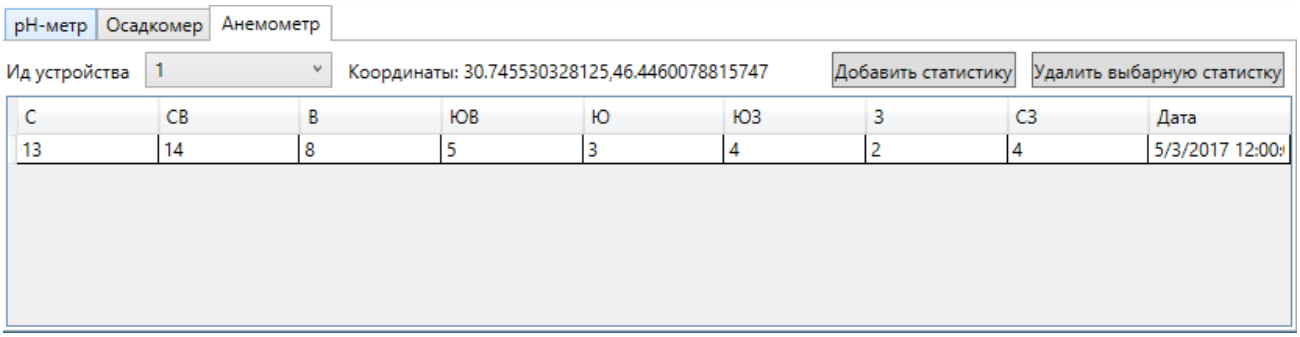


Рисунок 4.14 – «Анемометр»

4.2 Проектування рози вітрів

При виборі у головному вікні програми пункту «Диаграмм» та виборі підменю «Роза ветров» з’являється нове вікно, в якому можна будувати відповідні звіти (рис. 4.15).

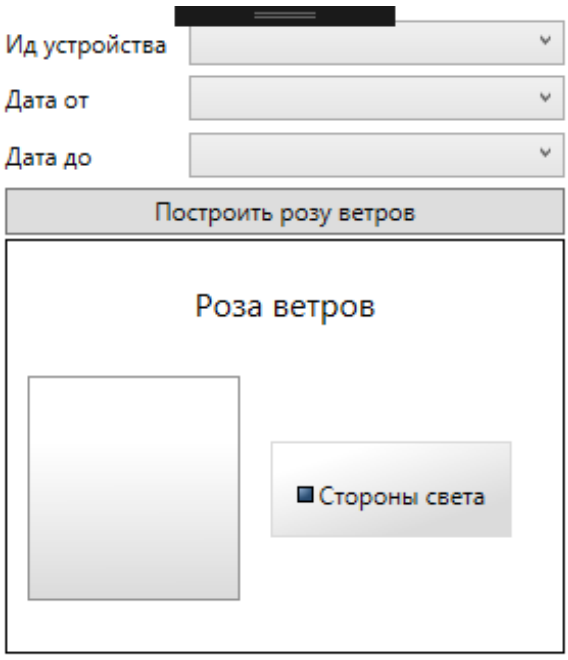


Рисунок 4.15 – «Роза ветров»

В цьому вікні є можливість вибору пристрою, який будевикористовуватися для побудови рози вітрів. Для цього необхідно обрати потрібний прилад і вибрати дату початку і кінця аналізу даних (рис. 4.16) та натиснути кнопку «Построить розу ветров».

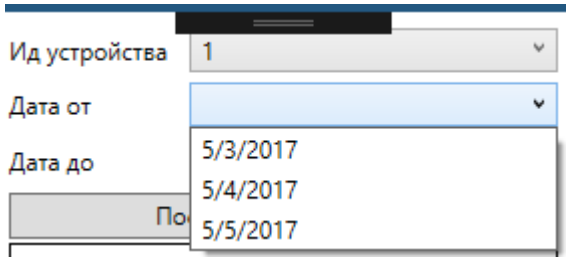


Рисунок 4.16 – Випадаючий список вибору дати.

Після генерації звіту ми побачимо на екрані нашу розу вітрів представлену у вигляді стовпчастої діаграми (рис. 4.17).

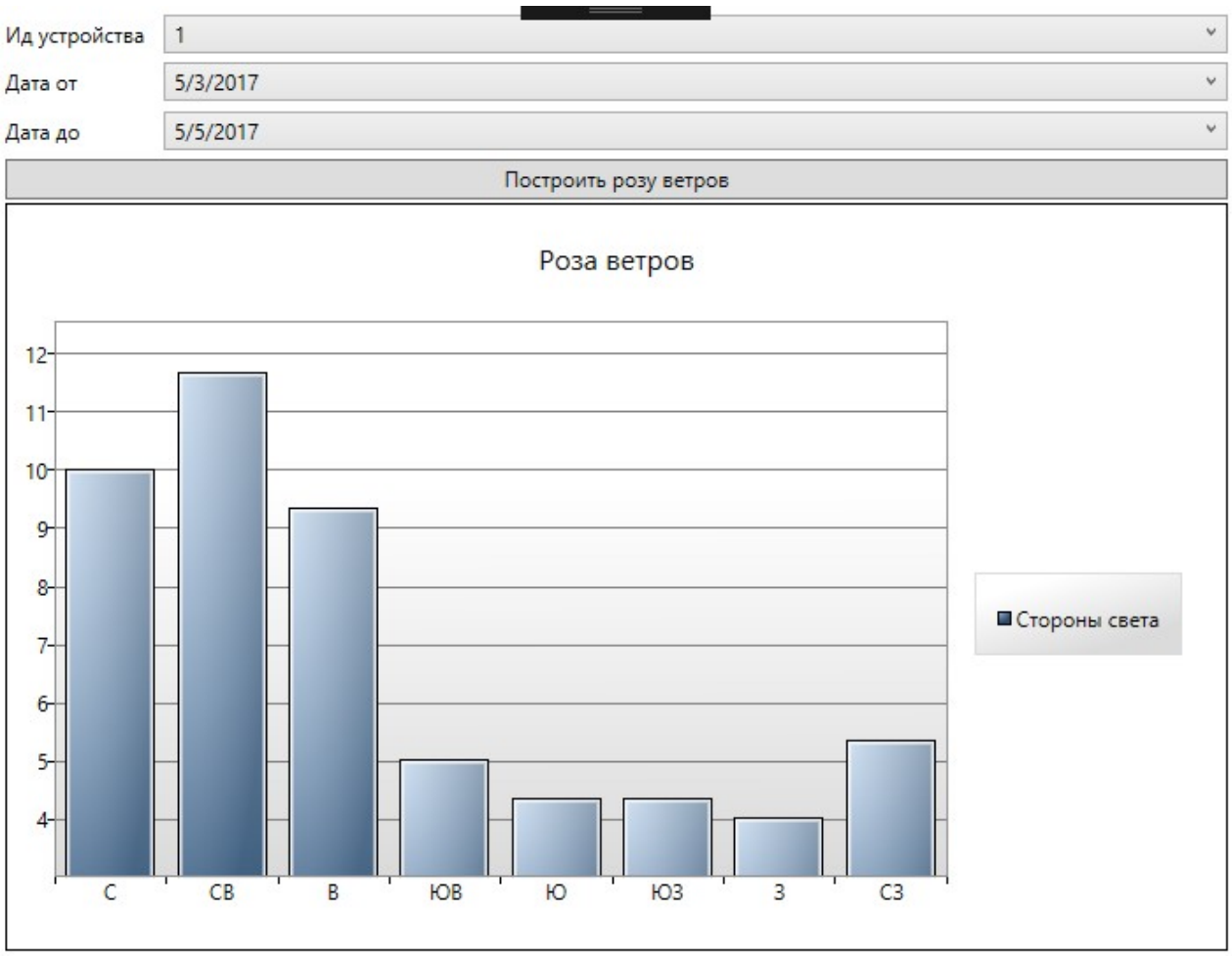


Рисунок 4.17 – «Роза ветров»

ВИСНОВКИ

Сучасний етап розвитку суспільства передбачає впровадження в усі сфери життєдіяльності новітніх інформаційних технологій, використання значних обсягів інформації. Актуальною стає необхідність розробкипрограмних продуктів, що дозволяють вироблення найбільш ефективних методів відбору інформації, її обробки і поширення.

Розроблений додаток дає можливість оперативного рішення типових задач, що виникають при дослідженні динаміки кліматичних характеристик, що значно спростить і полегшить роботу дослідників з великими наборами просторово-розподілених даних, а також забезпечить до них доступ. Надалі для проведення комплексної математичної і статистичної обробки даних, а також для візуалізації результатів планується використання програмного забезпечення, яке володіє значно більш багатими функціональними можливостями.

В дипломній роботі були спроектовані форми та запити, які виконують наступні функції:

– можливість генерування рози вітрів з отриманих кліматичних даних;
– вибір часового діапазону, за який буде проходити аналіз даних;
– введення персональних даних по кожному з доданих пристроїв;
– прив’язка пристроїв до геолокації;
– створення архіву з обробки кліматичних показників;
– перегляд інформації по кожному пристрою;

В подальшому планується……

ПЕРЕЛІК ПОСИЛАНЬ

1. Герасимов, О. В. Коллективная разработка функциональной модели информационной системы [Электронный ресурс] / О. В. Герасимов. – Режим доступа: http://www.ict.edu.ru/vconf/files/7316.doc, свободный. – Название с экрана.

2. Дженнингс, Роджер. Использование Microsoft Access 2000. Специальное издание [Текст]: учеб. пос. / Роджер Дженнингс. – М.: Издательский дом «Вильямс», 2000. – 1147 с.

3. Єрьоміна, Н. В. Проектування баз даних [Текст]: навч. посібник / Н. В. Єрьоміна. – К.: КНЕУ, 1998. – 208 с.

4. Кузин, А. В. Базы данных [Текст]: учеб. пособ. для студ. высш. учеб. заведений / А. В. Кузин, С. В. Левонисова. – 2-е изд. – М.: Издательский центр «Академия», 2008. – 320 с.

5. Ольховая, М. А. Подсистема документного и информационного оборота деканата [Электронный ресурс] / М. А. Ольховская, А. В. Новиков и др. – Режим доступа: http://db.biysk.secna.ru/conference/conference.conference.doc_ download?id_thesis_dl=470, свободный. – Название с экрана.

7. Рыбанов, А. А. Инструментальные средства автоматизированного проектирования баз данных [Электронный ресурс] / А. А. Рыбанов. –Режим доступа: http://window.edu.ru/window_catalog/redir?id= 47119&file=rybanov_ bd.pdf, свободный. – Название с экрана.

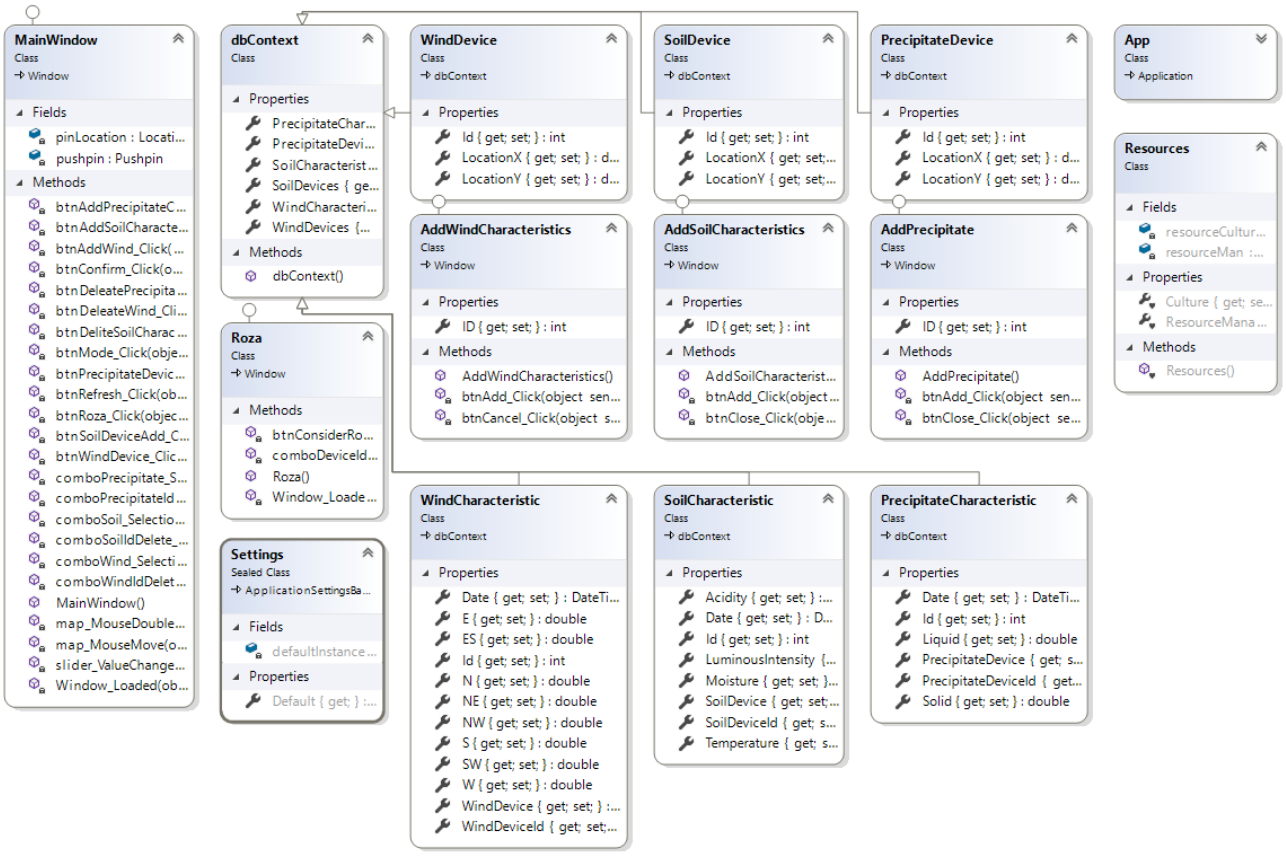
8. Создание таблицы в режиме конструктора [Электронный ресурс]. – Режим доступа: http://www.officepack.ru/access/11/, свободный. – Название с экрана.

9. Ткаченко, В. А. Системы управления базами данных и экспертные системы [Электронный ресурс] / В. А. Ткаченко. – Режим доступа: http://www.lessons-tva.info/edu/e-inf2/m2t4.html, свободный. – Название с экрана.

10. Элементы модели «сущность-связь» [Электронный ресурс] / Cit Forum. – Режим доступа: http://www.citforum.ru/database/dblearn/ dblearn08.shtml, свободный. – Название с экрана.

Д О Д А Т К И

Додаток А
Діаграма класів



Додаток В
ER - МОДЕЛЬ