МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ВЕБОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ МОНІТОРИНГУ ПАРАМЕТРІВ АТМОСФЕРНОГО ПОВІТРЯ»

Рівень «бакалавр»

Галузь знань 12 «Інформаційні технології»,

спеціальність 122 «Комп'ютерні науки»

Виконав здобувач 4 курсу

Турчин Михайло Михайлович

Керівник к.т.н., доцент

Чикунов Павло Олександрович

Рецензент к.т.н., доцент

Трофименко Олена Григорівна

Одеса – 2024

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Факультет кібербезпеки та інформаційних технологій

Кафедра інформаційних технологій

Галузь знань 12 Інформаційні технології

Спеціальність 122 «Комп'ютерні науки»

Назва освітньої програми «Комп’ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій

доцент Наталія Логінова _____

«____ »_____ 2024 року

З А В Д А Н Н Я

на кваліфікаційну (бакалаврську) роботу

здобувачу Турчину Михайлу Михайловичу

1. Тема роботи: «Веборієнтована інформаційна система для моніторингу параметрів атмосферного повітря».
Керівник роботи: к.т.н, доцент Чикунов Павло Олександрович
затверджені наказом НУ «ОЮА» від «02» квітня 2024 року № 809-06
2. Строк подання здобувачем роботи «20» травня 2024 року
3. Дата видачі завдання «15» вересня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та наявних аналогів	10.10.2023	
2	Формулювання цілей та завдань дослідження	26.12.2023	
3	Проєктування інформаційної системи	10.02.2023	
4	Програмна реалізація та тестування інформаційної системи	10.04.2024	
5	Оформлення звітної частини	17.05.2024	

Здобувач _____ Михайло Турчин _____
(підпис) (ім’я та прізвище)

Керівник роботи _____ Павло Чикунов _____
(підпис) (ім’я та прізвище)

АНОТАЦІЯ

Турчин М. М. «Веборієнтована інформаційна система для моніторингу параметрів атмосферного повітря». – Кваліфікаційна робота бакалавра за спеціальністю 122 «Комп'ютерні науки», освітньою програмою «Комп’ютерні науки».

Кваліфікаційна робота викладена на 40 сторінках основного тексту і 55 сторінках загального тексту, містить 3 розділи, 29 рисунків, 3 таблиці, 2 додатка, 23 використаних джерел.

Метою виконання роботи є проектування, програмна реалізація та модульне тестування веборієнтованої інформаційної системи для моніторингу параметрів атмосферного повітря, здатної отримувати регулярну, своєчасну та достовірну інформацію, проводити оцінку зібраних даних, надавати необхідну інформацію у зручному форматі кінцевому користувачеві.

Об'єктами роботи є методи та засоби моніторингу параметрів повітря, способи реєстрації даних за допомогою метеорологічного приладу та візуалізації отриманих даних у вигляді метеорологічних міток на онлайн-карті місцевості.

Предметом дослідження є техніки розробки веборієнтованої інформаційної системи для моніторингу параметрів атмосферного повітря, зокрема мультиплатформними засобами середовища Visual Studio з використанням відкритих геосервісів.

У кваліфікаційній роботі виконано реалізацію особистого кабінету користувача, завантаження наборів даних зі зовнішніх ресурсів, відображення карти місцевості, що відображає мітки з значеннями показників, формування віджетів, що описують актуальний стан та динаміку змін навколишнього середовища.

Ключові слова: інформаційна система, моніторинг, якість повітря, функціональні вимоги, Visual Studio, Xamarin.Forms, мова C#, мова XAML, GoogleMaps API, MSN Weather, мультиплатформний застосунок.

ABSTRACT

Turchin M. M. "Weboriented information system for monitoring atmospheric air parameters". – Bachelor's work in Computer Science, specialization 122 "Computer Science", educational program "Computer Science".

The thesis consists of 40 pages of main text and 55 pages of total text, comprising 3 chapters, 29 figures, 3 tables, 2 appendices, and references to 23 sources.

The aim of this work is the design, development, and testing of a web-oriented information system for monitoring atmospheric air parameters, capable of obtaining regular, timely, and reliable information, conducting assessment of collected data, and providing necessary information in a convenient format to end users.

The objects of this work are methods and means of air parameter monitoring, data registration methods using meteorological instruments, and visualization of obtained data in the form of meteorological tags on an online map of the locality.

The subject of research is the techniques for developing web-oriented information systems for monitoring atmospheric air parameters, particularly utilizing the multi-platform environment of Visual Studio with the use of open geo-services.

The bachelor's thesis includes the implementation of a user personal account, data set downloading from external resources, displaying a map of the locality with tags indicating parameter values, and the creation of widgets describing the current state and dynamics of environmental changes.

Keywords: information system, monitoring, air quality, functional requirements, Visual Studio, Xamarin.Forms, C#, XAML, GoogleMaps API, MSN Weather, multiplatform application.

ЗМІСТ

ВСТУП

Система моніторингу екологічної обстановки – це комплекс технологій, методів і засобів, що використовуються для постійного контролю за станом довкілля, включаючи якість повітря, води, ґрунтів, рослинності та інші параметри навколишнього середовища. До основних компонент системи моніторингу можна віднести датчики та вимірювальні прилади, системи збору та передачі даних, програмні модулі аналізу та обробки даних, системи візуалізації та звітності, системи сповіщення та аварійного реагування. Важливість таких систем полягає в їх здатності відстежувати рівні небезпечних забруднюючих речовин, таких як оксиди вуглецю, азоту, діоксид сірки, озон та пил, що дозволяє вчасно реагувати на екологічні загрози.

Актуальність дослідження обумовлена потребою у веборієнтованій інформаційній системі (ІС) для моніторингу параметрів атмосферного повітря, яка дозволяє регулярно збирати інформацію з різних джерел, проводити оцінку даних і надавати їх у зручному форматі користувачеві. Це сприятиме інформуванню громадськості та відповідних органів про стан повітря, що є критично важливим для прийняття ефективних заходів щодо покращення якості повітря та зниження ризиків для здоров'я населення.

Метою виконання роботи є проектування, програмна реалізація та модульне тестування веборієнтованої інформаційної системи для моніторингу параметрів атмосферного повітря, здатної отримувати регулярну, своєчасну та достовірну інформацію, проводити оцінку зібраних даних, надавати необхідну інформацію у зручному форматі кінцевому користувачеві.

Об'єктами роботи є методи та засоби моніторингу параметрів повітря, способи реєстрації даних за допомогою метеорологічного приладу та візуалізації отриманих даних у вигляді метеорологічних міток на онлайн-карті місцевості.

Предметом дослідження є техніки розробки веборієнтованої інформаційної системи для моніторингу параметрів атмосферного повітря, зокрема

мультиплатформними засобами середовища Visual Studio з використанням відкритих геосервісів.

Проведення оцінки параметрів навколишнього середовища потребує виявлення та відстеження конкретного характеру забруднення. Отримання подібної інформації дозволяє отримати ступінь впливу на здоров'я населення та основу для ретельно продуманих рішень. Зазвичай збір даних проводиться у стаціонарних метеостанціях та пересувних постах. Контроль якості повітря проводиться у хіміко-аналітичній лабораторії. Обробка, зберігання та аналіз даних відбувається в інформаційно-аналітичних центрах.

Методи досліджень: методи системного аналізу для дослідження предметної області, методи моделювання програмного забезпечення, методи програмної реалізації застосунків, методи модульного тестування.

Практичною новизною роботи є розробка веборієнтованої інформаційної системи у вигляді мультиплатформного застосунку, за допомогою якого можна інформувати користувача про якість атмосферного повітря вибраного міста України з формуванням метеорологічних міток на онлайн-карті, з можливістю візуалізації динаміки змін навколишнього середовища.

Публікації та апробація кваліфікаційної роботи:

1. публікація тези доповіді «Інформаційна система моніторингу параметрів атмосферного повітря» на Всеукраїнської науково-практичної конференції здобувачів вищої освіти «Суспільство та держава в умовах воєнного стану: виклики та перспективи» (м. Одеса, 27 квітня 24 р.);

2. публікація тези доповіді «Розробка інформаційної система для моніторингу параметрів атмосферного повітря» на Всеукраїнської науково-технічної конференції «Новітні технології в освіті, науці та виробництві» (м. Луцьк, 25-26 квітня 2024 р.).

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області та наявних аналогів

Моніторинг довкілля – це сукупність регулярного контролю, спостережень, що описується за незмінною програмою для оцінювання стану довкілля [4]. Моніторинг має бути систематичним і постійним процесом, що дозволяє збирати дані регулярно і тривалий час. Це дозволяє виявляти тренди та зміни в середовищі з плином часу. Моніторинг повинен бути динамічним і включати механізми для швидкого реагування на зміни в середовищі, такі як аварійні ситуації або несподівані екологічні події.

Будь-яка система моніторингу екологічної обстановки обов'язково має накопичувати інформацію зі знятих показань із різних джерел, які мають достатню точність, для надання коректних результатів. Дані, отримані в ході моніторингу, мають бути об'єктивними і незалежними від будь-яких сторонніх впливів. Це забезпечує достовірність і достатню обґрунтованість інформації про стан довкілля.

Більшість комплексів моніторингу акцентують свою увагу найбільш поширених і небезпечних забруднюючих довкілля речовинах, тому що вони потребують постійного спостереження.

Для систем контролю якості повітря найбільш затребуваним є відстеження таких речовин та мікрочастинок: оксид вуглецю, азот, діоксид сірки, озон, пил, азот, а також індекс якості повітря.

Використання стандартизованих методів збору даних, аналізу та оцінки дозволяє забезпечити порівнянність результатів між різними дослідженнями і регіонами.

Проведення всебічної оцінки навколишнього середовища потребує виявлення та відстеження конкретного характеру забруднення. Отримання подібної інформації дозволяє отримати ступінь впливу на здоров'я населення та основу для ретельно продуманих рішень. Збір даних проводиться у спеціалізованих постах: стаціонарний пост (метеостанція) та пересувний пост

(спеціалізований автомобіль чи інший засіб пересування). Контроль якості повітря проводиться у хіміко-аналітичній лабораторії. Обробка, зберігання та аналіз наданої інформації з постів відбувається в інформаційно-аналітичному центрі – це програмно- апаратний комплекс.

Накопичення, зберігання і доступ до даних моніторингу повинно бути організовано в ефективну систему, що дозволяє вчасно аналізувати та використовувати цю інформацію для прийняття рішень у сфері охорони довкілля.

Результати моніторингу мають бути доступні для широкої громадськості та зацікавленим сторонам, щоб забезпечити відкритість і прозорість процесу оцінки стану довкілля.

На даний момент існує низка геоінформаційних проектів, здатних виконувати моніторинг довкілля та надавати актуальні метеорологічні дані, а також формувати рекомендації для користувачів.

Закон України «Про охорону навколишнього природного середовища» встановлює створення системи державного моніторингу навколишнього природного середовища та проведення спостережень за його станом і рівнем забруднення.

Наразі деякі міста та облдержадміністрації власноруч впроваджують регіональні системи моніторингу довкілля. Деякі з цих міст співпрацюють з Гідрометслужбою для розширення та стандартизації мережі спостережень. У Київській області, місті Києві та декількох містах Дніпропетровської області (Дніпро, Павлоград, Нікополь, Кривий Ріг) вже запроваджено комунальні мережі спостережень і автоматичні системи моніторингу. Проте наразі ці мережі не координовані, обладнання є різнорідним, методи збору даних та показники не узгоджені з Міндовкіллям, що ускладнює їх використання на національному рівні.

Сучасні автоматичні пункти спостережень мають бути оснащені аналізаторами для визначення рівня твердих часток пилу, діоксиду сірки, оксидів азоту, оксиду вуглецю, приземного озону, летких органічних сполук та датчиками для вимірювання швидкості вітру, атмосферного тиску, температури та вологості повітря.

Підходи до моніторингу включають широкий спектр організаційних, кадрових, технічних, матеріальних та інших логістичних рішень, які потребують значних фінансових витрат. Ці підходи можна умовно розділити на дві групи.

1) Методи наземного та водного моніторингу, що включають автоматичні та напівавтоматичні датчики для постійного спостереження за станом атмосферного повітря та водних ресурсів, а також методи, які передбачають виїзд та проведення досліджень на місцях спостережень. Ця група також включає фото і відеозафіксування за допомогою безпілотних літальних апаратів (БПЛА) та організацію стаціонарних фото і відеоспостережень.

2) Використання методів дистанційного зондування земної або водної поверхні, що передбачає отримання космічних знімків необхідної роздільної здатності з подальшим їх аналізом та інтерпретацією. Також до цієї групи входить фото і відеофіксація за допомогою БПЛА.

Єдина екологічна платформа «ЕкоСистема» [6] створена Міністерством захисту довкілля та природних ресурсів України, та є автоматизованою інформаційно-аналітичною системою, спрямованою на забезпечення доступу до екологічної інформації. Вона була запущена в тестовому режимі в травні 2021 року, а в жовтні 2021 року уряд прийняв постанову щодо цієї платформи. Ця система призначена для розміщення екологічної інформації та надання екологічних адміністративних послуг. На сьогоднішній день, «ЕкоСистема» містить 10 онлайн-послуг (14 в процесі розробки), 6 активних реєстрів відкритих екологічних даних (10 в розробці) та 3 інтегровані інформаційні ресурси. Крім того, відділ Міндовкілля запустив вебресурс «ЕкоЗагроза», який включає вебсайт і мобільний застосунок, де знаходиться інформація про стан довкілля, а також дає змогу громадянам повідомляти про екологічні порушення.

В Європейському Союзі діє система Copernicus [7], що є глобальною програмою для моніторингу стану планети. Основна мета цієї програми полягає в наданні інформації про якість повітря та склад атмосфери як у межах, так і поза межами Європи, використовуючи дані з супутникових і наземних спостережень в поєднанні з моделями прогнозування. Copernicus надає користувачам доступ до

ції інформації через різні сервіси, включаючи Службу моніторингу атмосфери CAMS (The Copernicus Atmosphere Monitoring Service). CAMS забезпечує постійні дані та інформацію щодо складу атмосфери. Послуги CAMS фокусуються на п'яти основних напрямках: якість повітря та склад атмосфери, озоновий шар і ультрафіолетове випромінювання, хвилі та поверхневі течії, сонячна радіація і вплив на клімат.

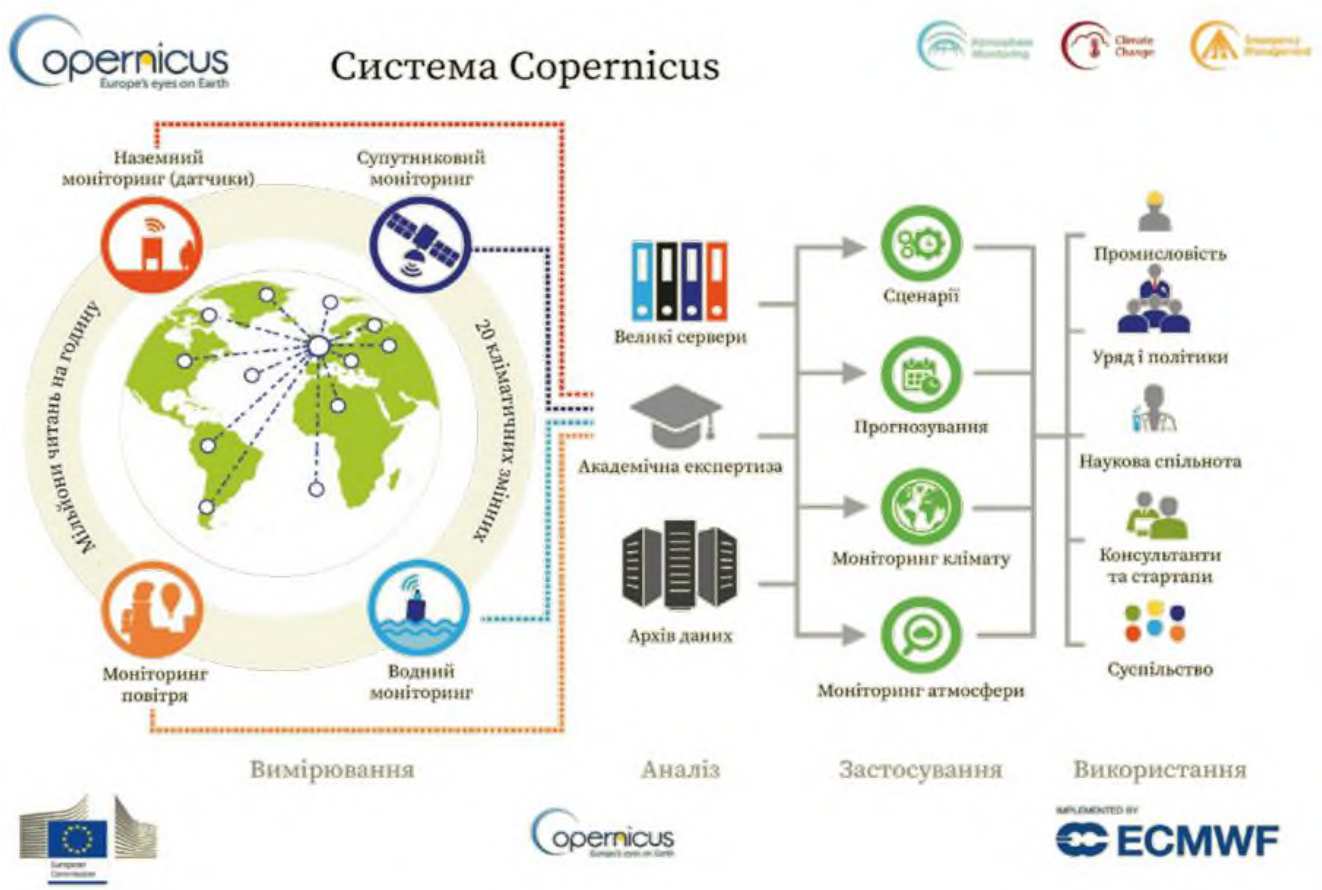

Рисунок 1.1 – Організація роботи інформаційної системи Copernicus

Європейський загальний погодинний індекс якості повітря [8] визначається як максимальне значення серед п'яти окремих індексів забруднюючих речовин, розрахованих одночасно.

В федеральному уряді США створено Федеральну агенцію з охорони навколишнього середовища (U.S. Environmental Protection Agency, EPA), яке відповідає за розробку та впровадження стандартів якості повітря, а також за контроль і оцінку рівнів забруднення повітря [9]. Вони здійснюють нагляд за

програмами моніторингу повітря на національному рівні та забезпечують доступ до інформації для громадськості через Інтернет та інші канали.

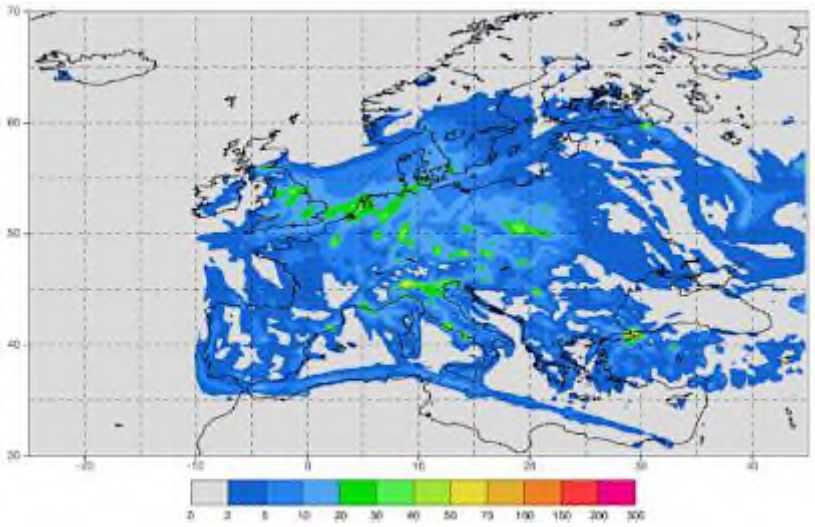

Рисунок 1.2 – Інтерактивна карта системи Copernicus

Велика кількість моніторингових станцій розташована по США для вимірювання різних параметрів якості повітря, таких як рівні токсичних речовин, пилу, оксидів азоту, сірководню та інших забруднювачів (рис. 1.3). Ці станції збирають дані, які потім аналізуються і використовуються для оцінки загального стану повітря та розробки заходів щодо його поліпшення.

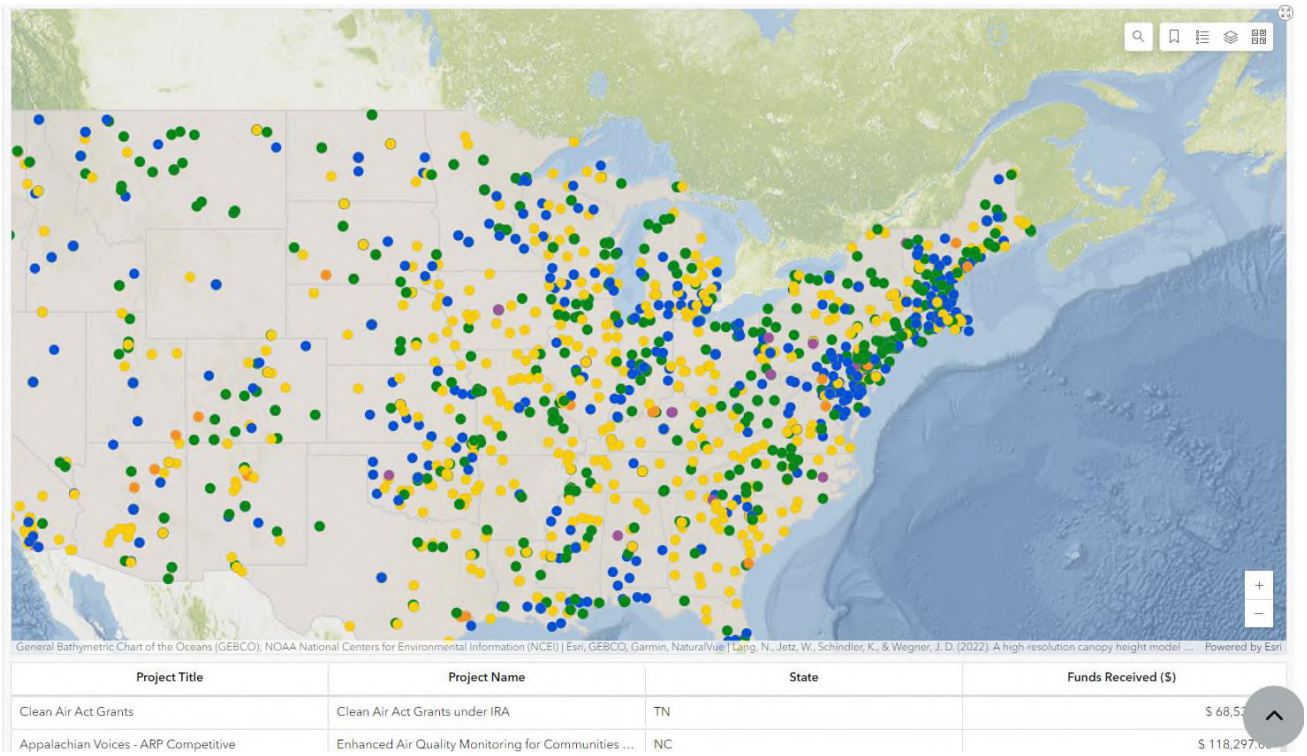

Рисунок 1.3 – Інтерактивна карта національної інфраструктури агенції EPA

1.2 Формулювання цілей та завдань дослідження

Аналіз предметної галузі та наявних аналогів дозволив сформулювати такі цілі та завдання дослідження:

1. виконати аналіз предметної галузі та наявних ІС моніторингу параметрів атмосферного повітря;
2. виконати огляд технологій та інструментів розробки веборієнтованих інформаційних систем для моніторингу параметрів атмосферного повітря;
3. визначити функціональні вимоги до веборієнтованої ІС;
4. виконати проектування веборієнтованої ІС, здатної отримувати регулярну, своєчасну та достовірну інформацію, проводити оцінку зібраних даних та надавати необхідну інформацію у зручному форматі користувачеві;
5. виконати програмну реалізацію веборієнтованої ІС у вигляді мультиплатформного застосунку;
6. перевірити коректну роботи окремих модулів ІС за допомогою модульних тестів на етапі кодування програми;
7. визначити практичну значущість розробки.

1.3 Опис очікуваних результатів

В результаті аналізу предметної області, огляду наявних аналогів інформаційної систем та формулювання головних цілей дослідження можна надати опис очікуваних результатів.

Необхідно реалізувати веборієнтовану інформаційну систему для моніторингу параметрів атмосферного повітря у вигляді мультиплатформного застосунку, за допомогою якого користувачі мають отримати такий сервіс:

1. виконувати реєстрацію та авторизацію в ІС з особистим кабінетом користувача з можливістю редагування особистої інформації;
2. надавати функціонал адміністрування ІС;

3. переглядати онлайн-карту вибраної місцевості, що відображає мітки з останніми зафіксованими значеннями показників якості повітря;

4. переглядати віджети, що описують актуальний стан та динаміку змін навколишнього середовища у вигляді графіків;

5. завантаження вибраних метеорологічних даних з зовнішніх ресурсів.

1.4 Висновки за розділом

В результаті аналізу предметної галузі:

– розглянуто складові процесу моніторингу стану довкілля;

– розглянуто наявні системи моніторингу повітря України, ЄС та США;

– сформульовано головні цілі та завдання дослідження;

– надано опис очікуваних результатів дослідження.

2 ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Огляд використовуваних технологій та інструментів

Реалізація проекту повинна виконуватися за допомогою сучасних інструментів, здатних ефективно та повною мірою вирішити те чи інше завдання. У табл. 2.1 показано обґрунтування вибору мови програмування для розробки бізнес-логіки ІС. В результаті порівняння вибрана сучасна мова програмування C# та середовище розробки Visual Studio Community 2022.

Таблиця 2.1 – Порівняння мов програмування

Метрика	**Вага метрики**	***C#***	**PHP**	**Java**	**Python**
Наявність досвіду роботи з мовою програмування	0,4	*9*	2	4	3
Швидкість розробки	0,3	*9*	4	7	5
Відмовостійкість	0,2	*7*	6	7	7
Продуктивність	0,1	*8*	7	9	7
Результат		*8,50*	3,90	6,00	4,80

У табл. 2.2 наведено показники, що вплинули на вибір платформи для розробки інтерфейсу ІС.

Таблиця 2.2 – Порівняння платформ розробки інтерфейсу ІС

Метрика	**Вага метрики**	**AngularJS**	**Xamarin. Forms**	**React**
Наявність досвіду роботи з бібліотекою	0,4	5	*9*	4
Швидкість розробки	0,3	5	*9*	6
Продуктивність	0,2	6	*8*	6
Низький поріг входу	0,05	7	*6*	5
Розмір бібліотеки	0,05	3	*6*	5
Результат		4,00	*6,90*	3,90

Xamarin – це відкрита платформа, яка дозволяє розробляти продуктивні застосунки для iOS, Android та Windows, з використанням спільного коду [18]. Завдяки Xamarin, значна частина коду може бути перевикористана на різних платформах, що дозволяє розробникам писати бізнес-логіку однією мовою (XAML та C#), а отримувати характеристики, специфічні для кожної платформи. Програми Xamarin використовують бібліотеку BCL, яка надає доступ до тисяч бібліотек з додатковими функціями.

Xamarin.Forms – це компонента платформи Xamarin, яка дозволяє розробникам створювати макети інтерфейсів для спільного використання в затосунках iOS, Android і Windows. Xamarin.Forms дозволяє розробникам створювати інтерфейси користувача за допомогою мови розмітки XAML (рис. 2.1) та писати бізнес-логіку мовою C# (рис. 2.2). Ці інтерфейси користувача автоматично адаптуються до відповідних платформ. Застосунки Xamarin.Forms можуть бути написані для iOS, Android та універсальної платформи Windows (UWP).

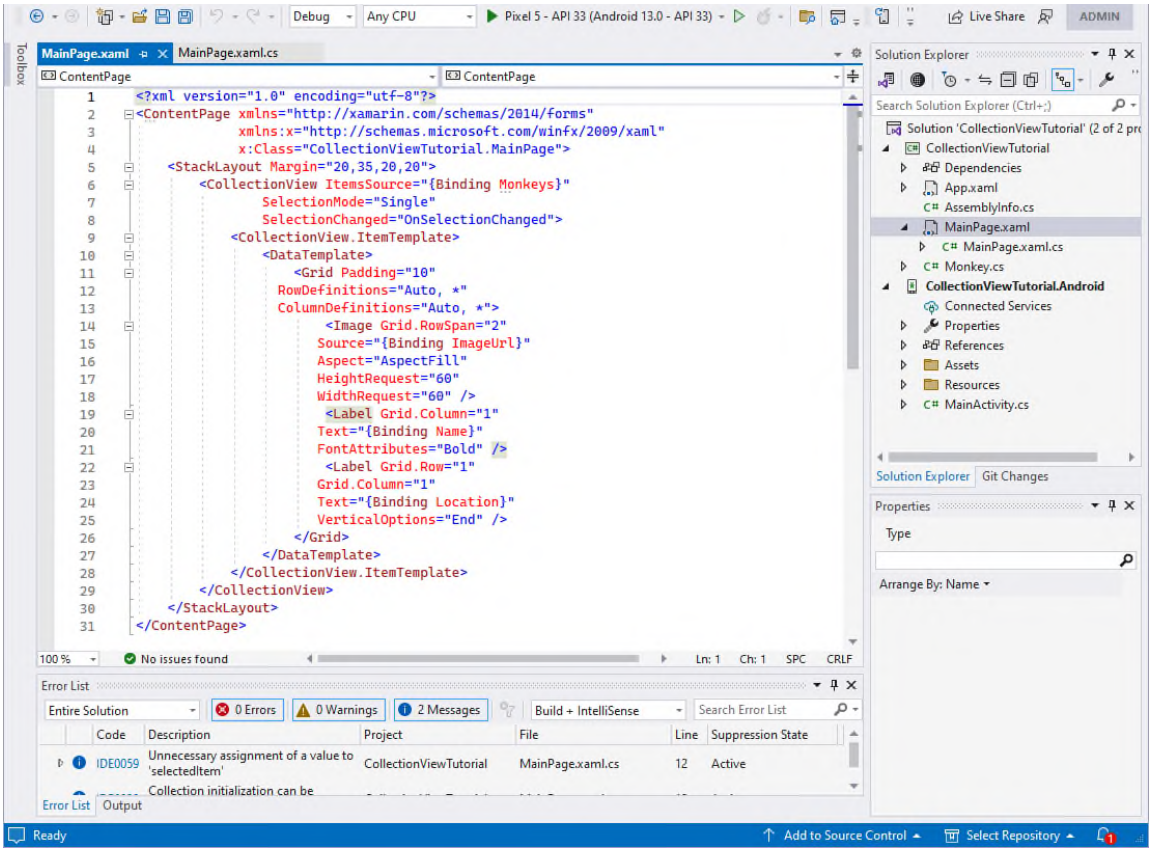

Рисунок 2.1 – Процес розробки інтерфейсу користувача у Xamarin

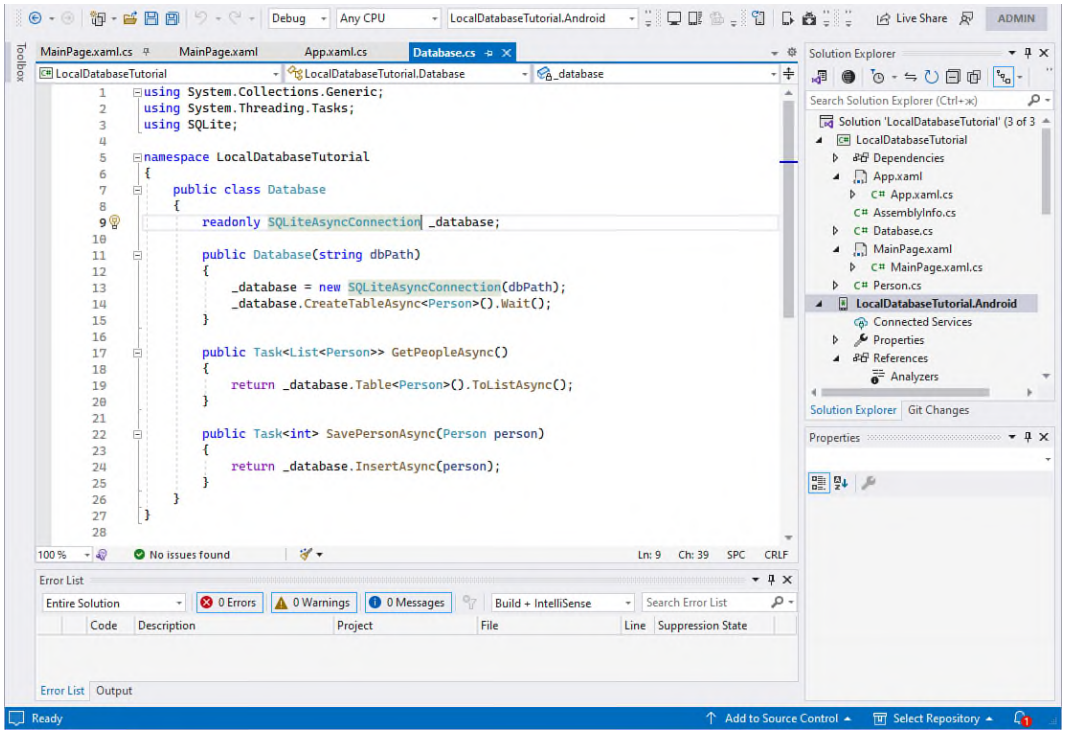

Рисунок 2.2 – Процес розробки бізнес-логіки застосунку Xamarin

NuGet-пакет SkiaSharp – це двовимірна графічна система для .NET на базі графічного механізму Skia з відкритим кодом, який широко використовується в продуктах Google. Розробник може використовувати SkiaSharp для створення двовимірної векторної графіки, растрових зображень і тексту.

CsvHelper – це бібліотека для роботи з CSV-файлами в .NET [15]. Вона спрощує процес читання з і запису в CSV-файли, надаючи чистий і зрозумілий API. CsvHelper дозволяє налаштовувати різні аспекти розбору та запису CSV.

Для реалізації інформаційної системи вибрана платформа Xamarin.Forms [14] разом з NuGet-пакетами SQLite.NET для організації БД [11] та SkiaSharp для побудови графіків [13], CsvHelper для перетворення наборів даних у CSV-форматі у XML-формат, GoogleMapsApi [16] для обгортки вебслужб «Карти Google» для .NET.

У табл. 2.3 наведено показники вибору бібліотеки для реалізації геосервісу. Розглянуто відкритий проєкт OpenStreetMap, який спрямований на збір, збереження та розповсюдження загальнодоступних геопросторових даних, створення інструментів для роботи з ними силами спільноти волонтерів.

OpenStreetMap створений спільнотою розробників мап, які вносять і підтримують дані про дороги і багато чого іншого по всьому світу [12]. Також розглянуто геосервіс MSN Weather [21], який дозволяє отримати найновіший прогноз погоди, включаючи температуру, вологість, опади для вибраного місця розташування.

Таблиця 2.3 – Порівняння бібліотек для клієнтської частини програми

Метрика	**Вага метрики**	***Google Maps***	**OpenStreet-Map**	**MSN Weather**
Наявність досвіду роботи з геосервісом	0,3	*6*	7	6
Швидкість розробки	0,35	*8*	6	6
Наявність онлайн-документації	0,15	*8*	6	5
Низький поріг входу	0,15	*7*	7	7
Продуктивність	0,05	*7*	6	7
Результат		*7,20*	6,45	6,05

Для відображення карти на вебсторінці вибрано геосервіс Google Maps [10]. Google Maps надає розробникам API, який дозволяє розробникам інтегрувати різноманітні функції картографії та місцевості у свої застосунки. API дозволяє вбудовувати карту Google у вебсторінки або мобільні застосунки, налаштовувати карти зі своїм вмістом та стилями та отримувати доступ до різних служб, таких як геокодування, маршрутизація та пошук місць (рис. 2.3). Доступ до API зазвичай здійснюється за допомогою ключів API, які розробники отримують, зареєструвавши свої застосунки на платформі Google Cloud.

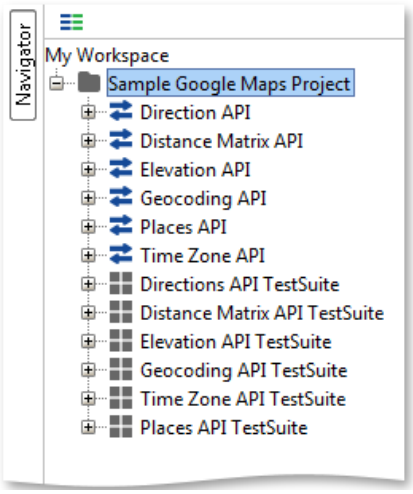

Рисунок 2.3 – Приклад використання Google Maps API

Для реалізації СКБД для вебсервісу вибрано бібліотеку SQLite.NET [11], яка встановлюється у середовище Visual Studio як NuGet-пакет (рис. 2.4). SQLite.NET – це потужна бібліотека для роботи з базами даних у .NET-проєктах на мові C#. Забезпечує легкий доступ до даних, швидкі операції зчитування та запису, а також можливість створення та керування базами даних прямо з коду. Ця бібліотека є популярним вибором для розробників завдяки своїй надійності, широкій підтримці різних платформ, включаючи мобільні пристрої.

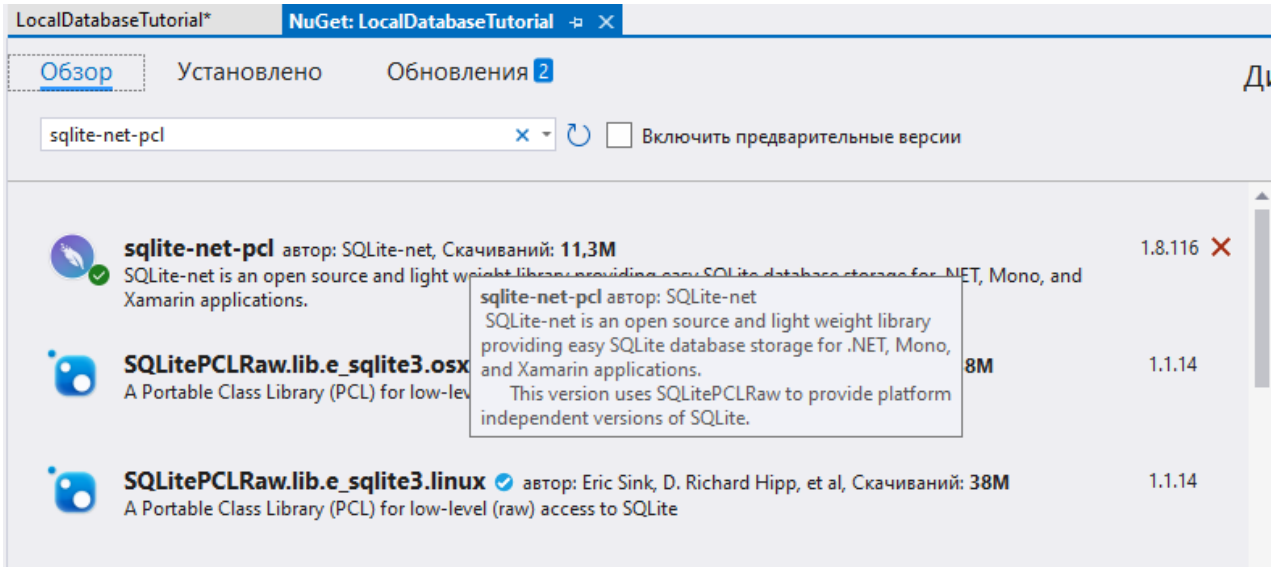

Рисунок 2.4 – Встановлення пакету SQLite.NET у IDE Visual Studio

2.2 Постановка формальних вимог до інформаційної системи

Аналіз предметної області, наявних аналогів веборієнтованої інформаційної системи моніторингу параметрів атмосферного повітря [5-9] дозволив виконати постановку функціональних вимог до інформаційної системи [17, 20, 22]:

– можливість доопрацювання, підтримки протягом усього життєвого циклу ІС (виправлення помилок, збільшення функціоналу);

– повинен працювати у всіх популярних веббраузерах;

– надійність: стабільна робота ІС та підтримка підказок користувачеві;

– відмовостійкість: під час виникнення помилки система повинна гарантувати працездатний стан;

– швидкодія: робота у ІС, починаючи від переходу між сторінками, закінчуючи розрахунками та виведенням інформації, повинна виконуватися за прийнятний час;

– ергономічність: експлуатація ІС повинна бути інтуїтивно зрозуміла і зручна, для всіх кінцевих користувачів, не враховуючи їх кваліфікацію.

Дослідження предметної області дозволило виділити три ролі користувачів інформаційної системи: адміністратор, авторизований користувач та неавторизований користувач (гість).

Неавторизований користувач повинен мати можливість:

– зареєструватися у системі;

– авторизуватися;

– переглядати карту, графіки та статистику, що відображають актуальний стан та динаміку у часовому інтервалі навколишнього середовища;

– завантажити дані вибраного екологічного показника у заданому діапазоні;

– переглядати інформацію про проект, контактну інформацію та новини.

Авторизований користувач повинен мати можливість:

– переглядати карту, графіки та статистику, що відображають актуальний стан та динаміку у часовому інтервалі навколишнього середовища;

– завантажити дані вибраного екологічного показника у заданому діапазоні;

– завантажувати датасет із показаннями датчика;

– переглядати історію власних завантажень;

– переглядати інформацію про проект, контактну інформацію та новини;

– доступ до особистого кабінету з можливістю редагувати особисту інформацію;

– вийти з системи.

Адміністратор повинен мати можливість:

– затверджувати/відкидати користувальницькі завантаження та вносити зміни до найменування станції та міста, в яких проводилися вимірювання;

– переглядати карту, графіки та статистику, що відображають актуальний стан та динаміку у часовому інтервалі навколишнього середовища;

– завантажити дані вибраного екологічного показника у заданому діапазоні;

– додавати/редагувати та видаляти новини;

– завантажувати датасет із показаннями датчика;

– переглядати історію своїх завантажень та інших користувачів;

– отримувати доступ до особистого кабінету з можливістю редагувати особисту інформацію;

– переглядати інформацію про проект, контактну інформацію та новини;

– редагувати/вилучати/створювати новини;

– вийти з системи.

Функціонал всіх перерахованих користувачів показаний у вигляді карти інформаційної системи, на якій зображені можливі переміщення по вебсторінках (рис. 2.5).

Одним із основних моментів у процесі розробки та контролю роботи програмного продукту є система логування. Протокол ІС працює для всіх користувачів з можливістю авторизації у системі та веденням журналу дій для кожного авторизованого користувача.

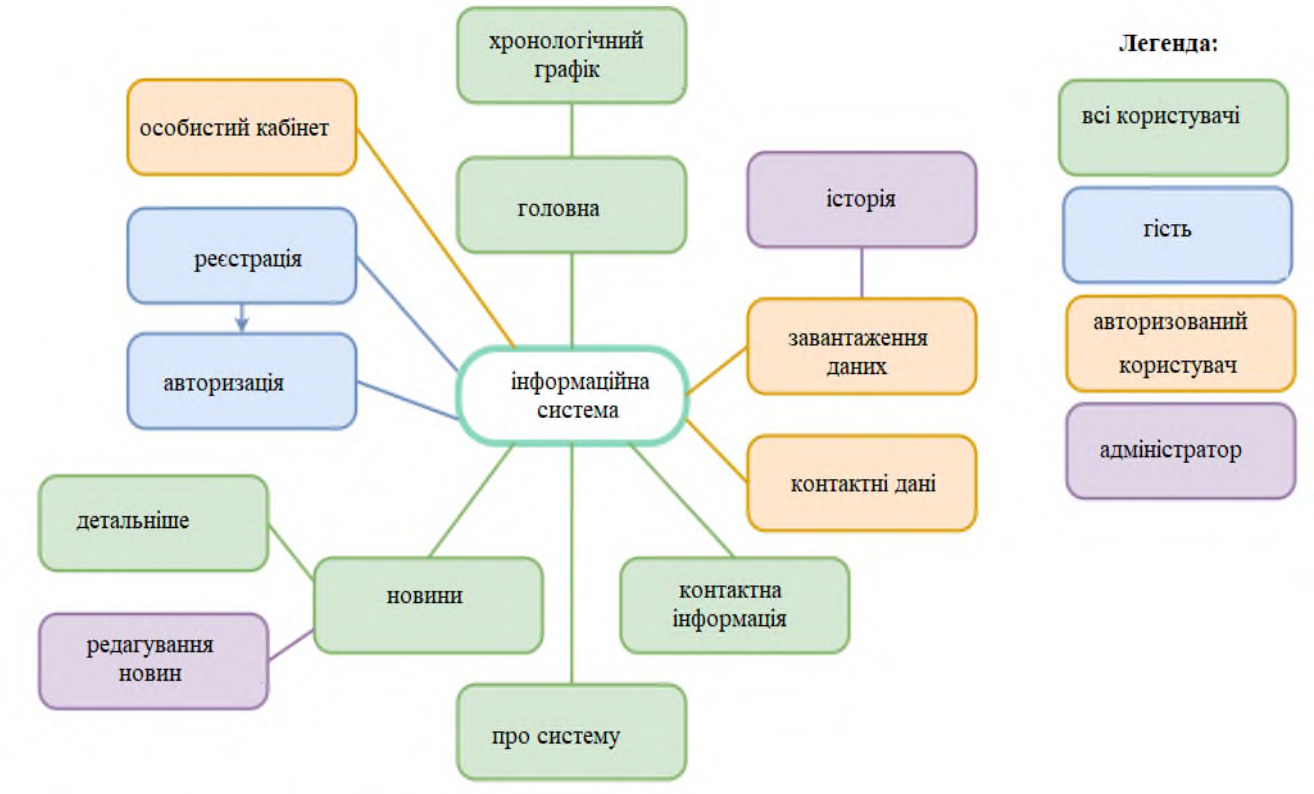

Рисунок 2.5 – Функціонал рівнів користувачів інформаційної системи

2.2 Проектування бази даних та доступу до зовнішніх ресурсів

На рис. 2.6 показано процес загрузки зовнішнього набору даних з метеорологічними показниками в інформаційну систему. Активатором процесу завантаження виступає авторизований користувач чи адміністратор. Після закінчення процесу завантаження необхідно згенерувати повідомлення про статус завантаження (вдало/невдало).

Для візуалізації змісту завантаженого набору показників розроблена сторінка з виведенням історії. Кількість метеорологічних показників в одному наборі може бути достатньо великою, тому передбачена фільтрація для зручного сприйняття інформації.

У момент натискання на вебсторінку «Історія» застосовується стандартний фільтр, що відображає всі наявні показники. При застосуванні користувачем фільтра формується HTTP-запит, у якому передаються налаштування фільтра. Цей запит передається через вебсервер та сервер бізнес-логіки до сервера баз даних, де відбувається отримання запитаних записів або повідомлення про помилку у разі непередбаченої ситуації.

Збережені дані показань із вимірювального приладу потрапляють у БД, однак, при завантаженні авторизованим користувачем вони не враховуватимуться у розрахунках та відображенні, доки записи не підтверджені адміністратором. Процедура підтвердження адміністратором потрібна, тому що це забезпечує додаткову перевірку завантажених даних.

На рис. 2.7 зображена логічна модель бази даних вебсервісу. Структура логічної моделі БД складається з таких таблиць:

– Role зберігає інформацію про ролі вебсервісу;

– User зберігає інформацію про користувачів;

– UserLoadDataSet зберігає інформацію про завантаження наборів даних;

– ParameterType зберігає найменування показників спостережень;

– Action зберігає конкретні рекомендації.

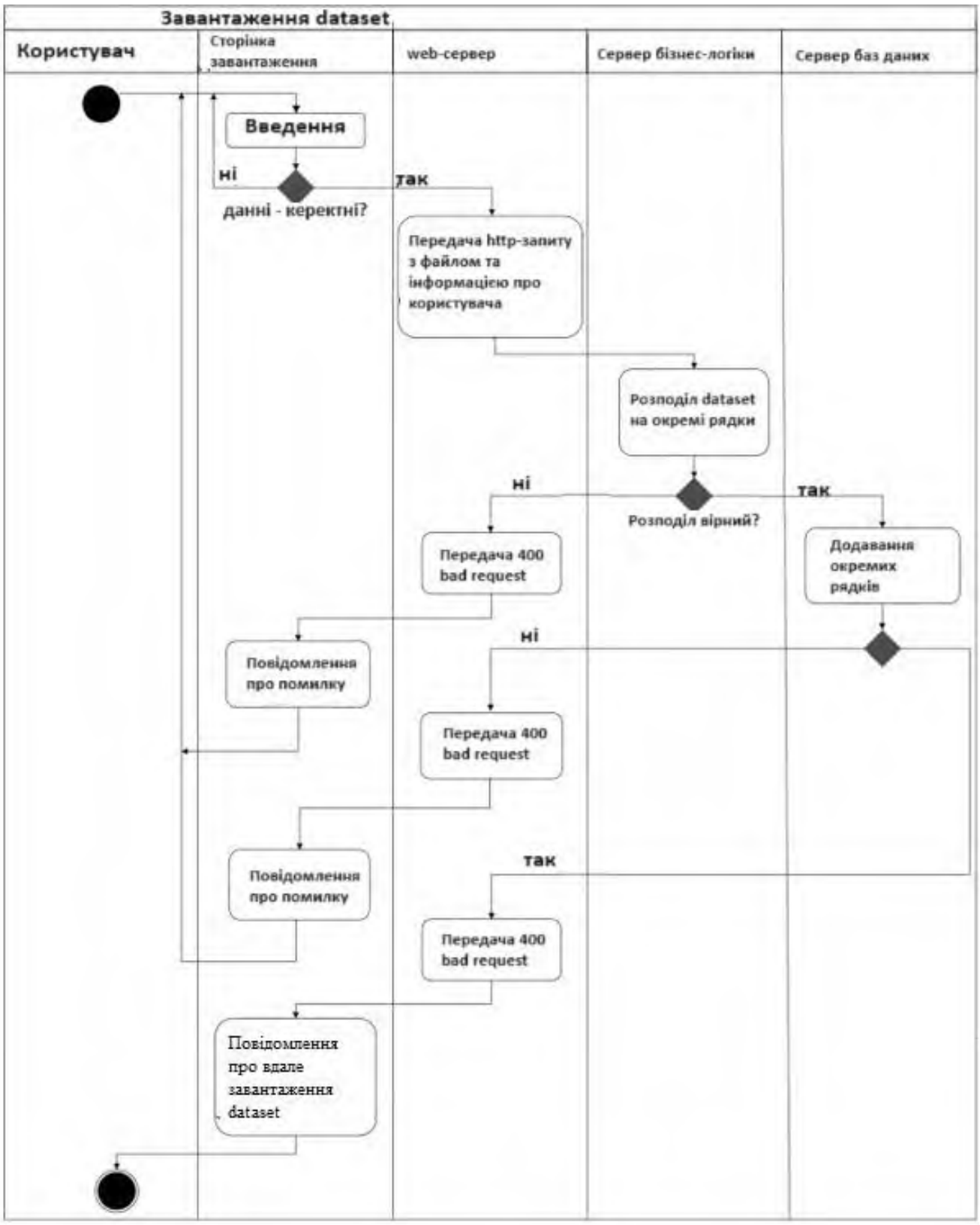

Рисунок 2.6 – Діаграма сценарію завантаження зовнішнього набору даних з метеорологічними показниками

Такі таблиці зарезервовані для майбутніх ревізій ІС:

– Measurement зберігає значення зі спеціалізованого пристрою;

– Station зберігає інформацію про населені пункти та станції на яких проводилися виміри;

– SensorData зберігає опис датчиків;

– SensorType зберігає назви датчиків;

– Parameter зберігає інформацію про інтервальні значення показників і відповідні рекомендовані дії.

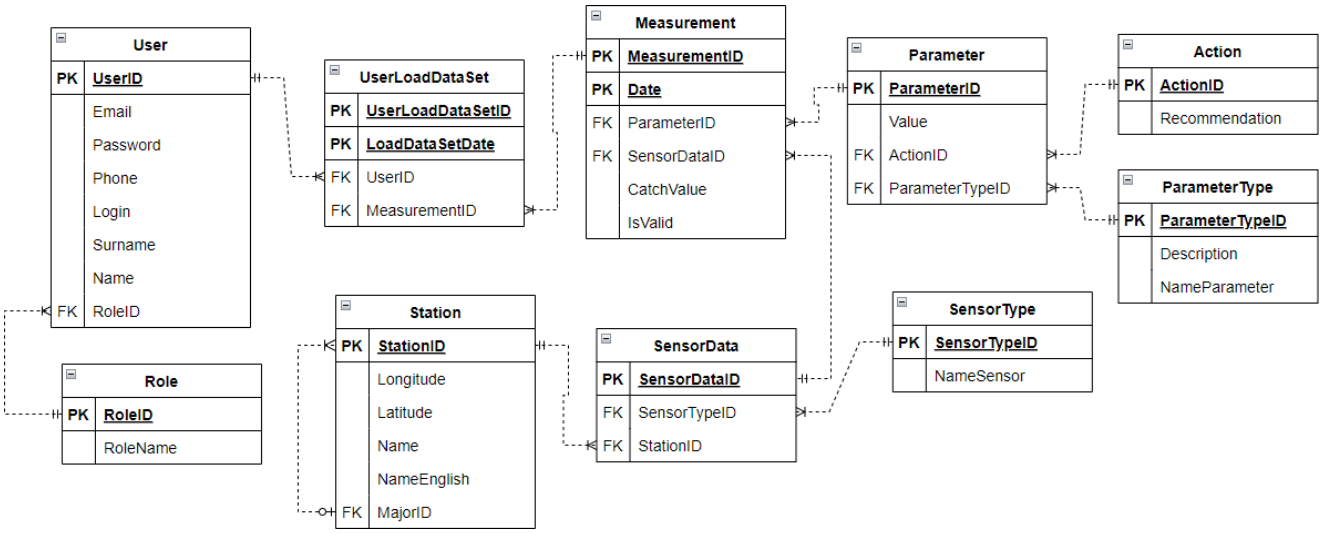

Рисунок 2.7 – Логічна модель бази даних

Для постачання до ІС інформації про екологічну обстановку необхідні зовнішні набори даних метеорологічних показників: AQI, температура, вологість повітря, атмосферний тиск, здійснювалася з доступного цифрового датчика та мікроконтролера. Вміст подібного набору у CSV-форматі, отриманого з зовнішнього ресурсу, наведено на рис. 2.8.

2.3 Розробка ієрархії класів ІС

Для відображення внутрішньої структури застосунку Xamarin.Forms розроблено UML-діаграми, у яких наведено основні класи, інтерфейси та їхні зв'язки.

На рис. 2.9 представлений опис ієрархії класів бізнес-логіки ІС. Класи AdminController і HomeController, ErrorController приймають всі запити з клієнтської частини і формують відповіді на запити.

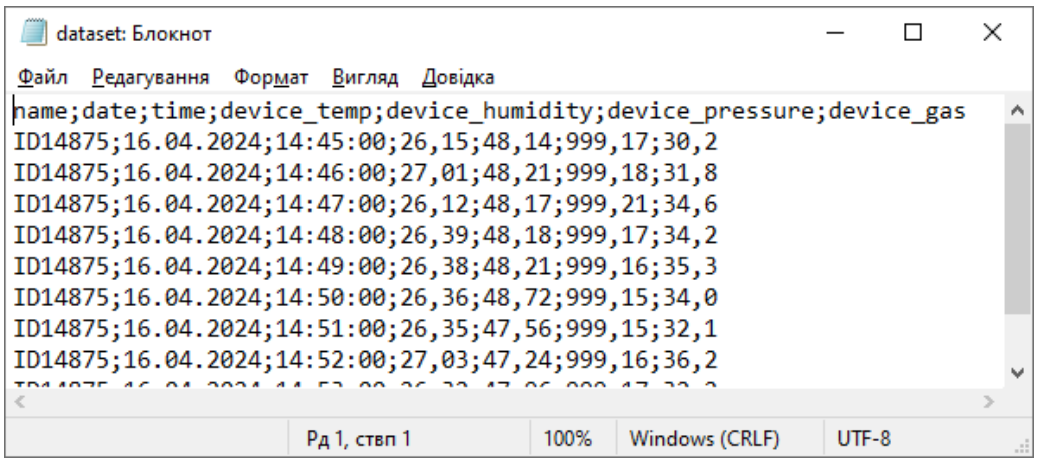

Рисунок 2.8 – Вміст зовнішнього набору даних

На рис. 2.10 представлений опис ієрархії класів. Класи UserService, EcologyService, ViewRenderService, ExcelService відповідають за бізнес-логіку та реалізують основний функціонал ІС.

Клас ViewRenderService використовується виключно для конвертації подання метеорологічних показників у рядок для поміщення в запит через CSV-файл.

Клас AirSystemDBContext – це контекстний клас, в якому зіставлені сутності та стосунки, які були визначені в моделі БД.

2.4 Проєктування інтерфейсу інформаційної системи

У сучасних застосунках та інформаційних системах важливу роль відіграє інтерфейс користувача [19-20].

Отже, для інтерфейсу необхідно встановити такі загальні вимоги всіх сторінок:

– інтерфейс має бути наближений до мінімалістичного стилю;

– необхідно використати одну колірну схему;

– потрібно налаштувати адаптивний дизайн для найбільш потрібних дозволів екрану;

– повинен підтримуватися зручне введення даних, інформативний висновок даних і підказки при допущенні помилок.

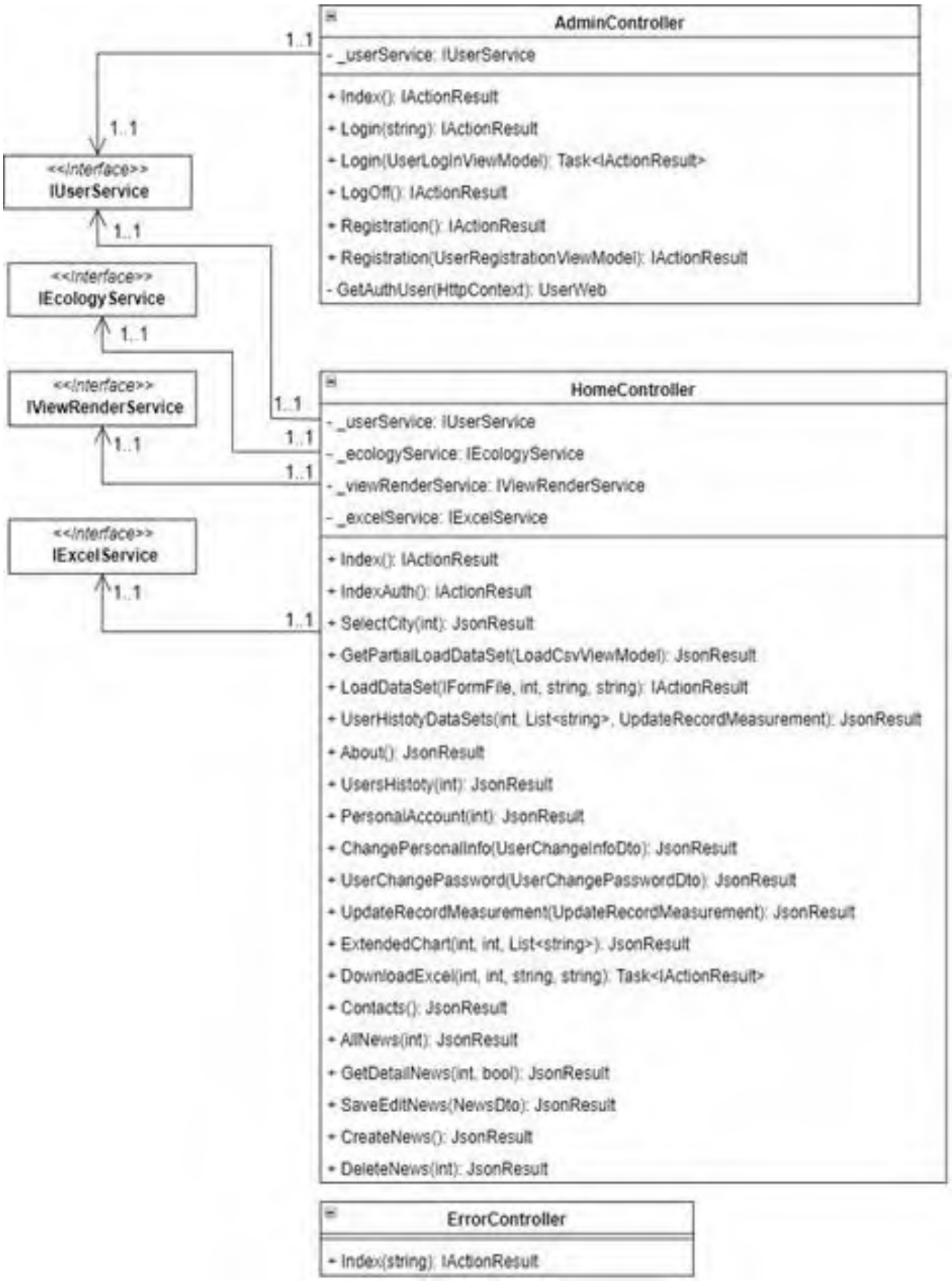

Рисунок 2.9 – UML-діаграма ієрархії класів бізнес-логіки

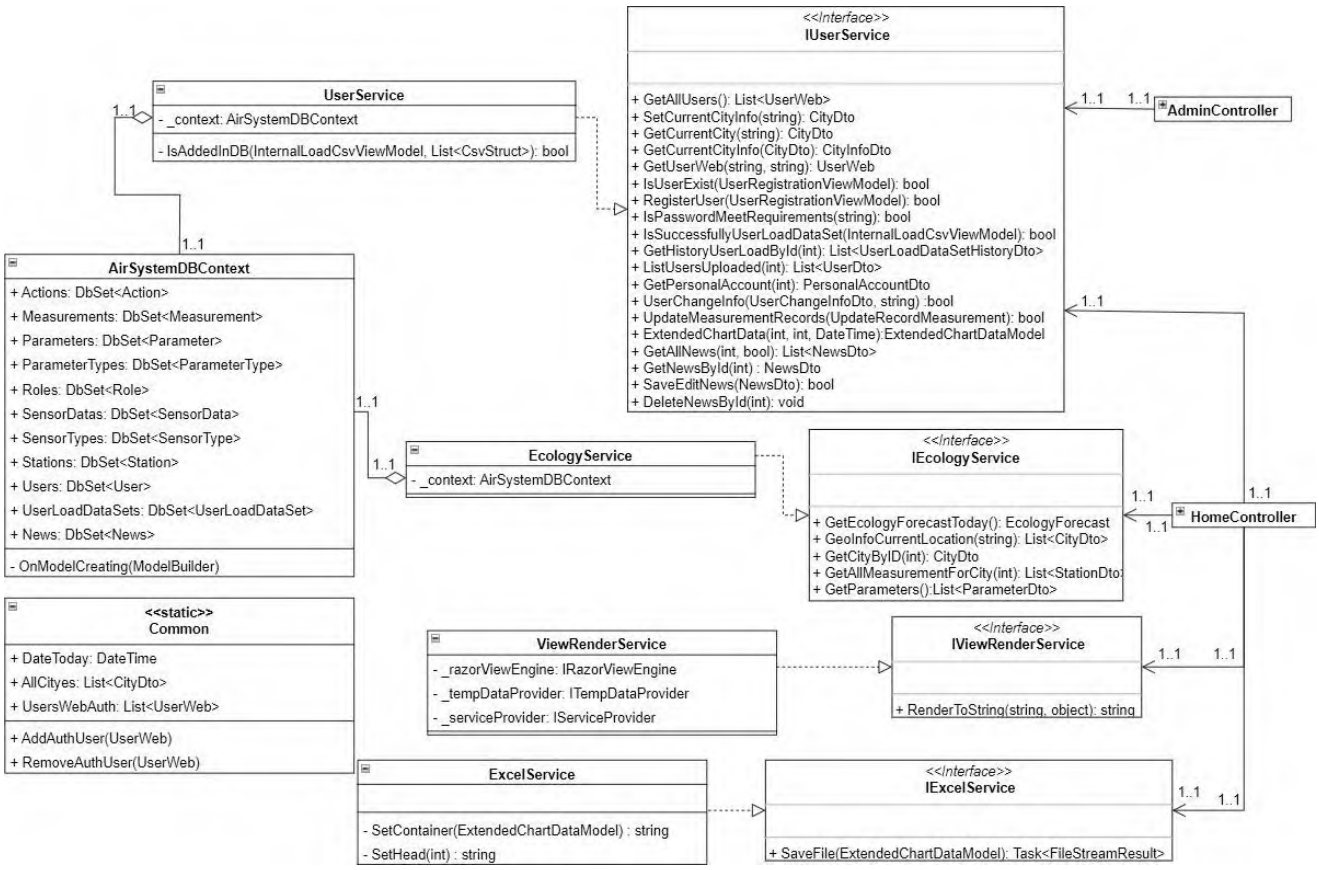

Рисунок 2.10 – UML-діаграма ієрархії класів клієнтської частини

Xamarin.Forms дозволяє розробникам створювати інтерфейси користувача в XAML за допомогою коду програмної частини в C#. Ці інтерфейси користувача на кожній платформі готуються до перегляду як власні елементи управління. У Xamarin елементи, показані на екрані пристрою, називаються Visual Element, У таких пристроях, як мобільний телефон, вони видимі або відомі як візуальний елемент.

2.5 Розробка UML-діаграм інформаційної системи

Розроблено діаграму послідовності (sequence diagram) внесення адміністратором змін у історію завантажень інформаційної системи (рис. 2.11).

Розроблена діаграма варіантів використання (use case) ІС, яка дозволяє уявити типи ролей акторів та їх взаємодію із системою (рис. 2.12).

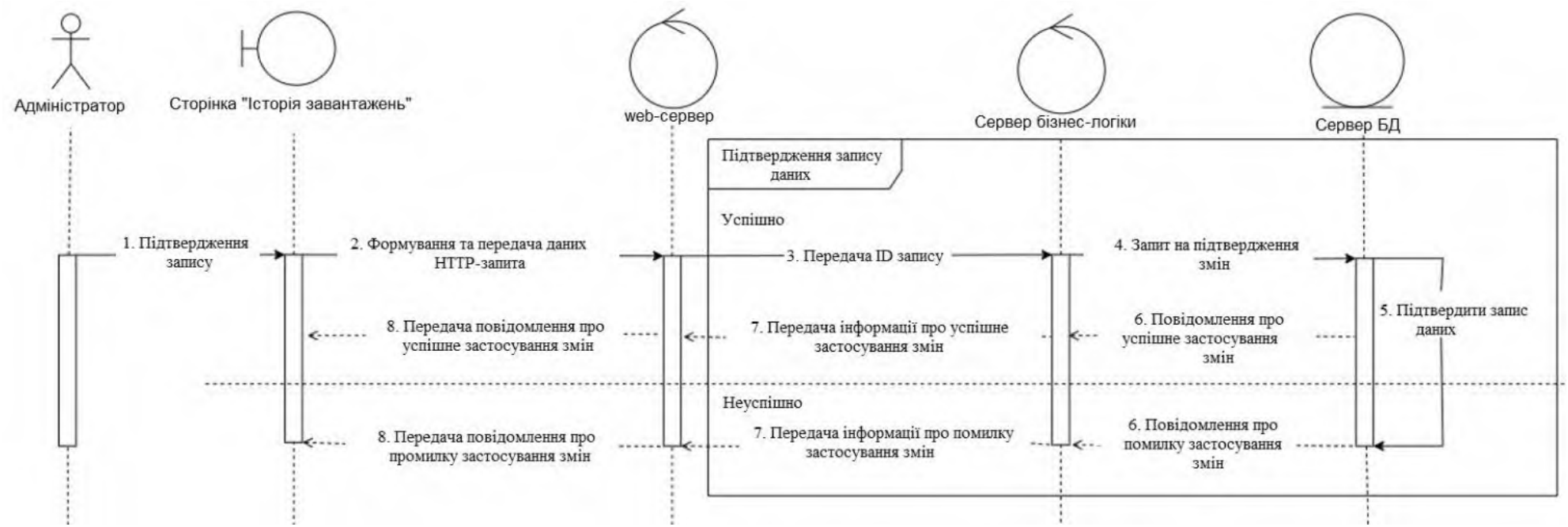

Рисунок 2.11 – Діаграма послідовності внесення адміністратором змін у історію завантажень

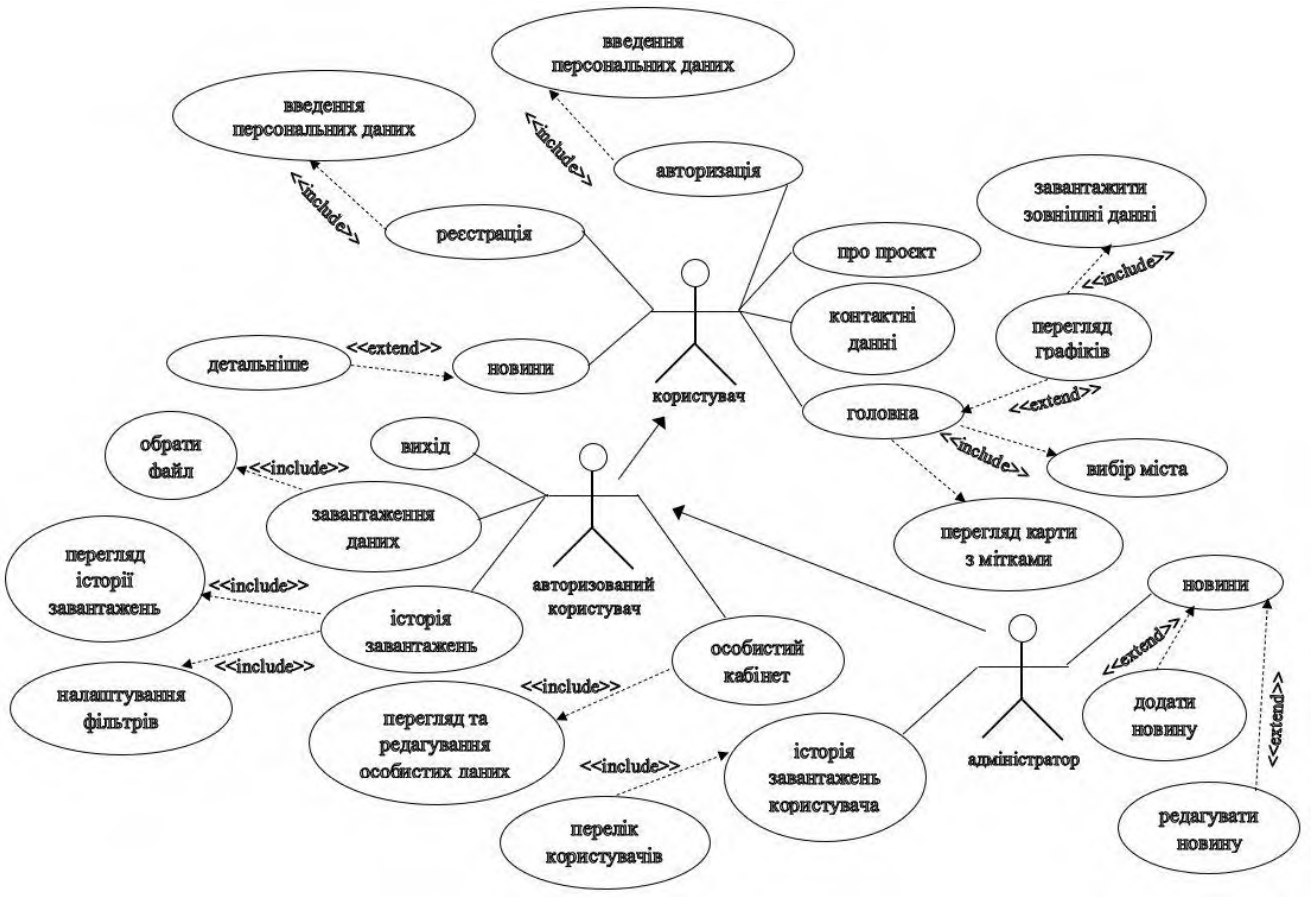

Рисунок 2.12 – Діаграма варіантів використання інформаційної системи

2.6 Висновки за розділом

За результатами аналізу та складання функціональних вимог визначено функціонал ІС моніторингу параметрів атмосферного повітря. Визначено засоби розробки, за допомогою яких вдасться реалізувати весь перерахований функціонал. Під час проектування інформаційної системи отримано результати:

– сформульовано функціональні вимоги до сервісу інформаційної системи;

– виконано вибір засобів розробки ІС, зокрема для організації БД, побудови графіків, роботи з наборами даних, доступу до геосервісів;

– побудовані UML-діаграми ієрархії класів, діаграма варіантів використання ІС, діаграма послідовності внесення адміністратором змін;

– проаналізована структура зовнішнього набору даних метеорологічних показників;

– розроблена логічну модель БД.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Опис головних етапів програмної реалізації

Програмна реалізація інформаційної системи виконана за допомогою сучасних інструментів, здатних ефективно та повною мірою вирішити те чи інше завдання. Для розробки ІС використана мова програмування C# та середовище розробки Visual Studio Community 2022.

Для розробки графіків застосовано двовимірний графічний пакет SkiaSharp. Для зберігання даних ІС вибрано бібліотеку SQLite.NET. Для відображення карт з метеорологічними показниками використано геосервіс Google Maps API. Для переводу зовнішніх CSV-наборів даних у формат XML використана бібліотека CsvHelper.

Розробка інтерфейсу ІС виконана засобами Xamarin.Forms для універсальної платформи Windows (UWP), яка забезпечує загальну платформу програм на кожному пристрою починаючі з версії Windows 10 (рис. 3.1). Платформа UWP дозволяє застосунку працювати на будь-якому пристрої: настільному комп'ютері, ноутбуці, смартфоні IPhone або Android, планшеті, приставці Xbox. Основні API UWP однакові на всіх пристроях Windows. Для створення ІС використано IDE Visual Studio Community 2022 у ОС Windows 11 (рис. 3.2). UWP є одним із багатьох способів створення клієнтських програм для Windows. Програми UWP використовують API WinRT для забезпечення інтерфейсу користувача та асинхронних функцій, які підходять для пристроїв, підключених до Інтернету.

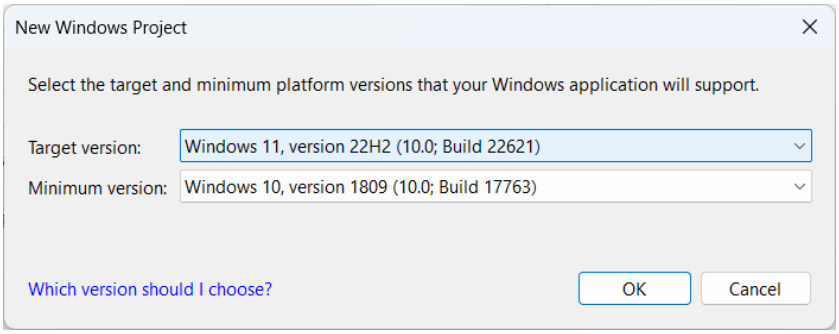

Рисунок 3.1 – Вибір версії платформи для Windows-застосунку

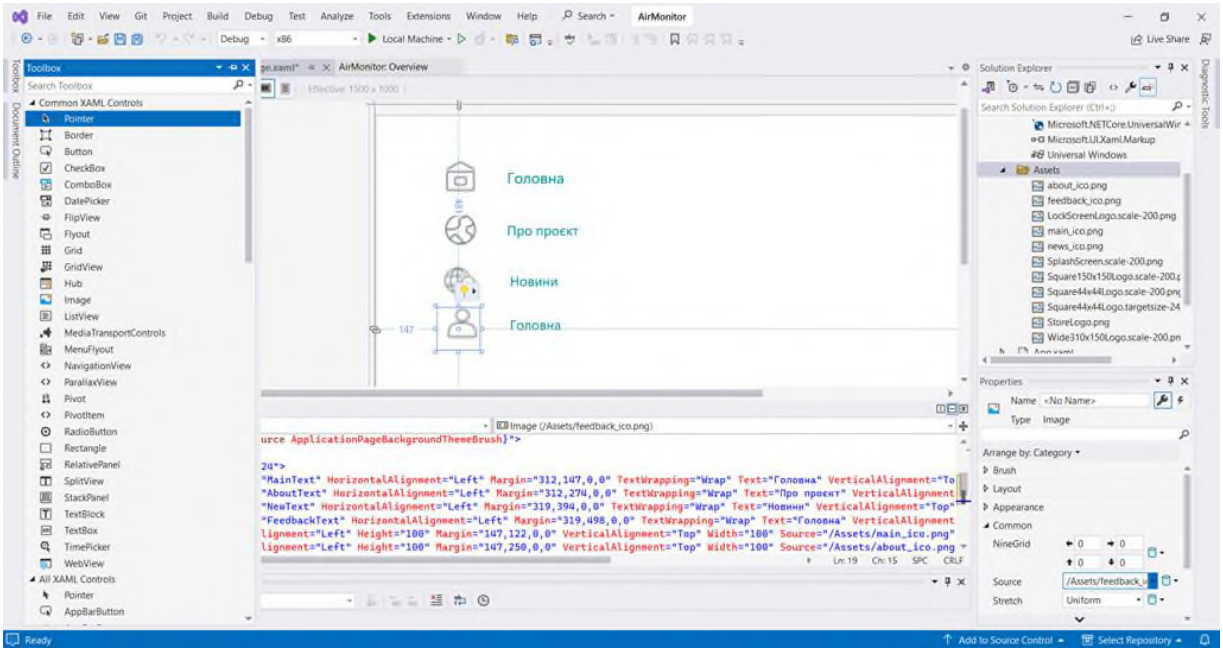

Рисунок 3.2 – Процес розробки навігаційного меню у візуальному режимі для платформи UWP

Окрім платформи UWP середовище Visual Studio надає розробникам можливість розробки застосунків з графічної підсистемою .NET Framework під назвою Windows Presentation Foundation (WPF). WPF є сучасним аналогом звичайних WinForms, тобто дизайнером для побудови клієнтських застосунків Windows з використанням декларативної мови розмітки користувацького інтерфейсу XAML.

Майстер Template Studio для WPF – це розширення Visual Studio, яке прискорює створення WPF-застосунків (рис. 3.3). Отриманий проект WPF містить найновіші функції Windows із застосуванням перевірених шаблонів. Template Studio виконує важку роботу зі створення коду, щоб крок за кроком пройти через процес додавання функцій і представлень до проекту (рис. 3.4).

Розробка структури інтерфейсу виконана на основі мови XAML із вбудованою підтримкою адаптації до змін розміру екрана (рис. 3.5). Для організації навігаційного меню використано елемент управління «CollectionView», за допомогою якого можна прив'язати колекцію графічних об'єктів до елементів TextBlock. Елементи інтерфейсу користувача реагують на розмір і роздільну здатність екрана, на якому працює програма, регулюючи їх макет і масштаб.

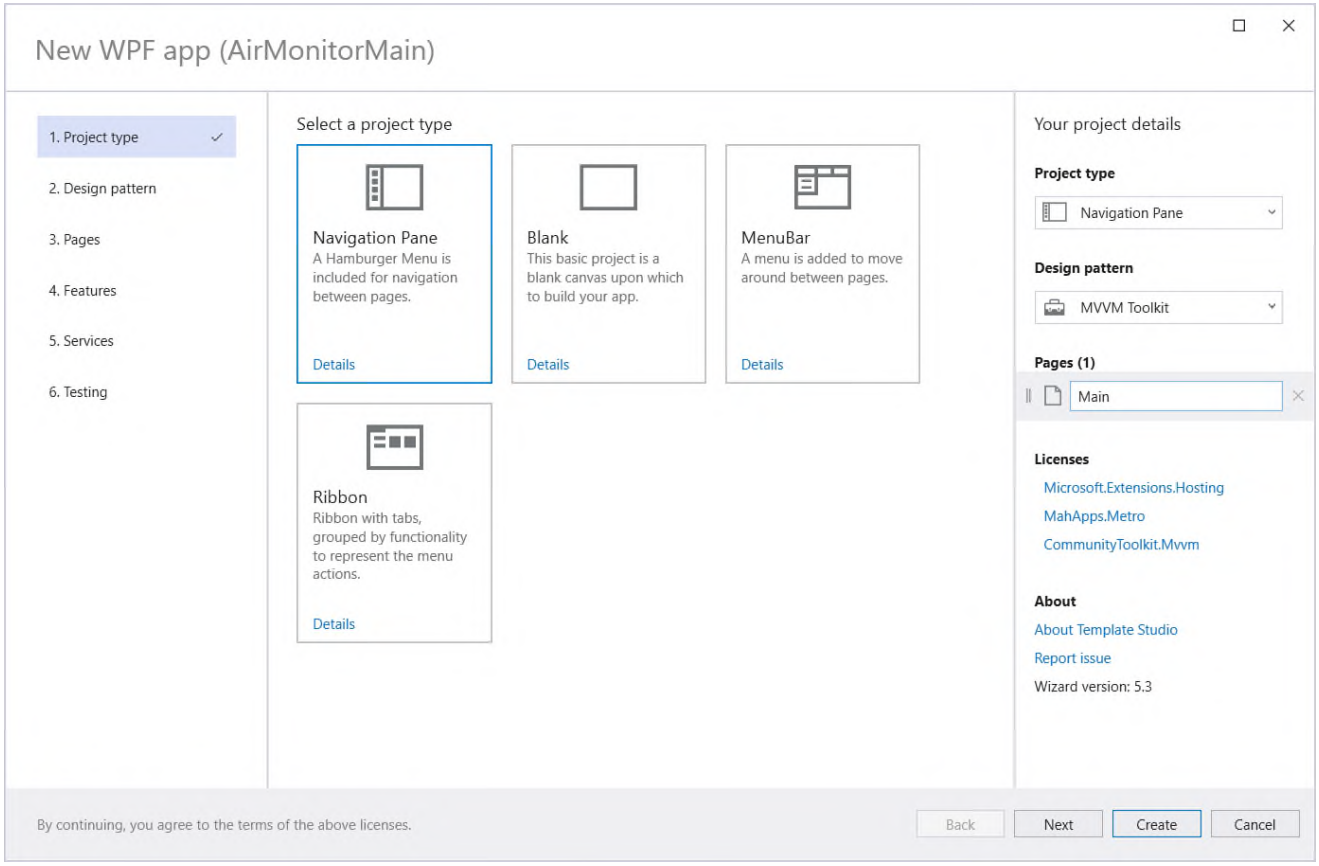

Рисунок 3.3 – Процес розробки головної сторінки за допомогою Template Studio

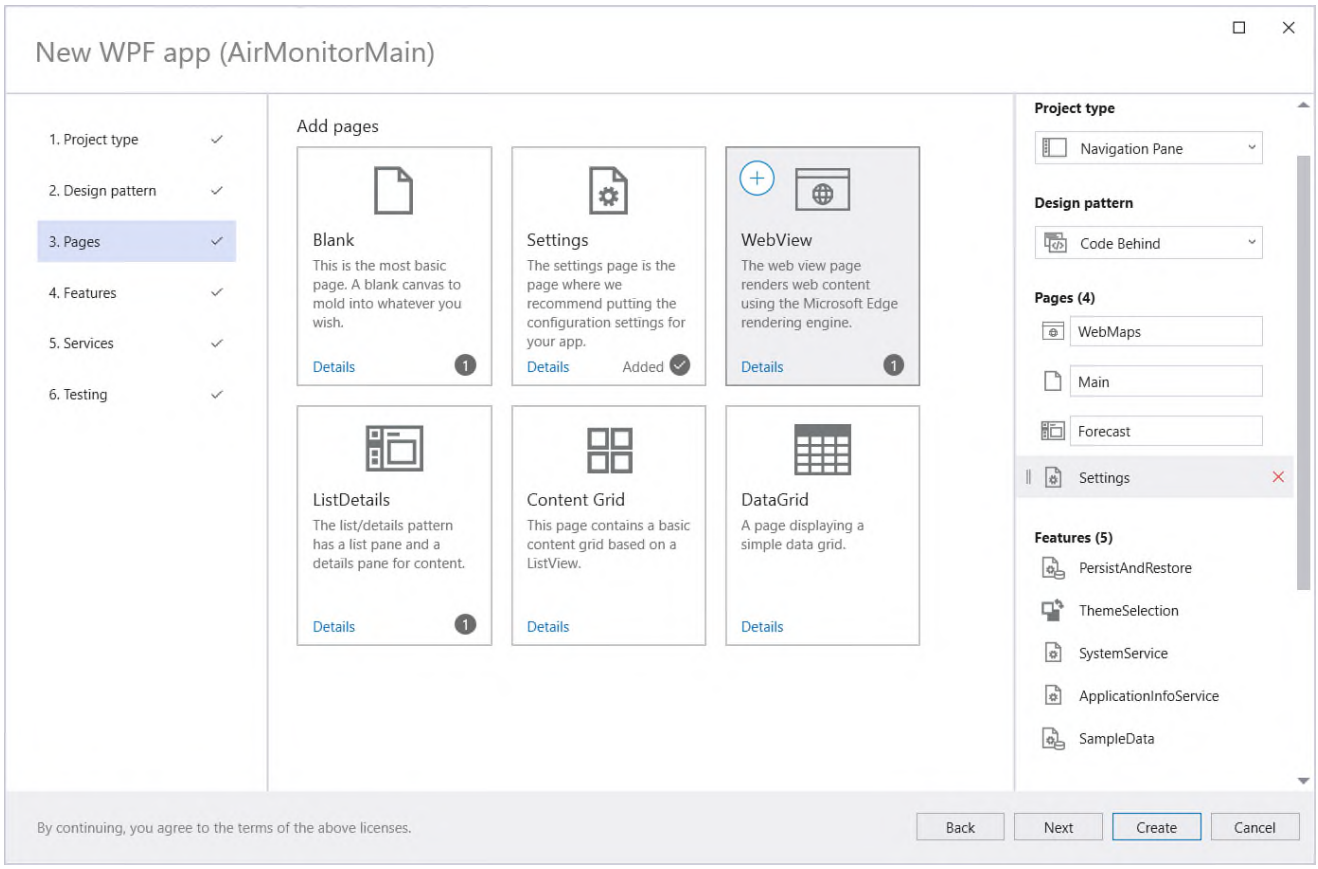

Рисунок 3.4 – Процес додавання нових сторінок до застосунку

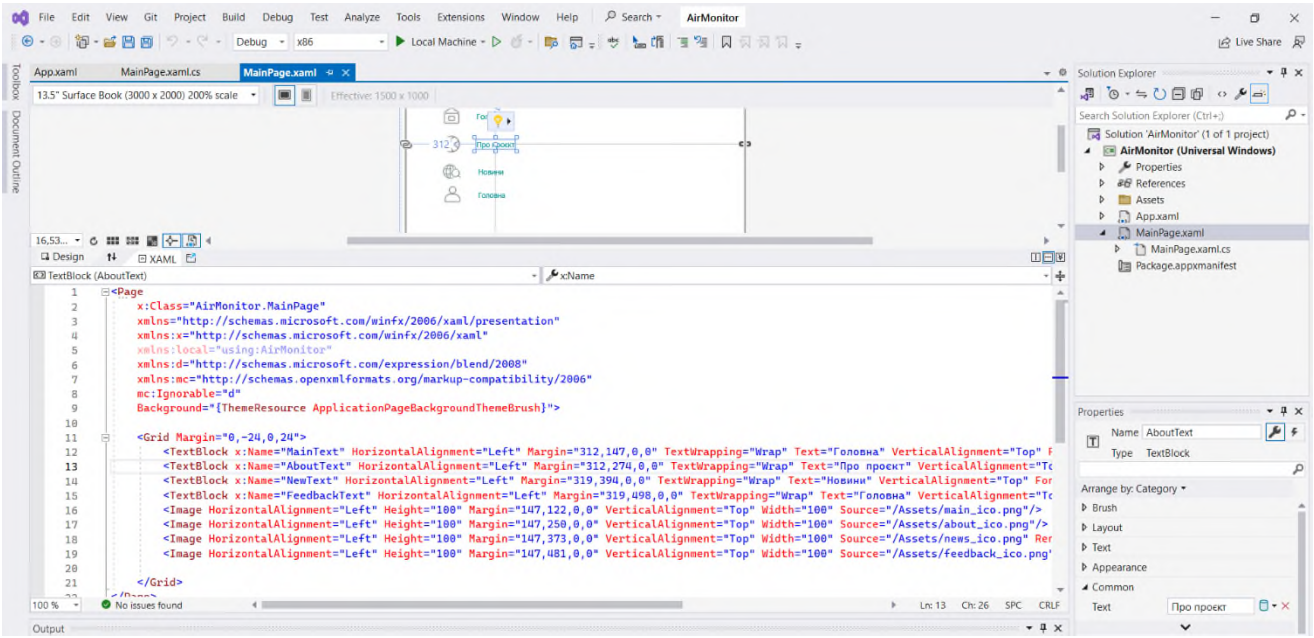

Рисунок 3.5 – Процес налаштування елементів інтерфейсу мовою Xaml

Далі необхідно зв’язати навігаційні елементи та команди для переходу до відповідних сторінок за допомогою мови C#. Передбачено, що ІС може відображати мапи з різних геосервісів, зокрема Google Maps та радарної мапи MSN [21]. На рис. 3.6 показано процес додавання віджета для відображення радарної мапи MSN з метеорологічними показниками до сторінки ІС. Результат роботи віджета показано на рис. 3.7.

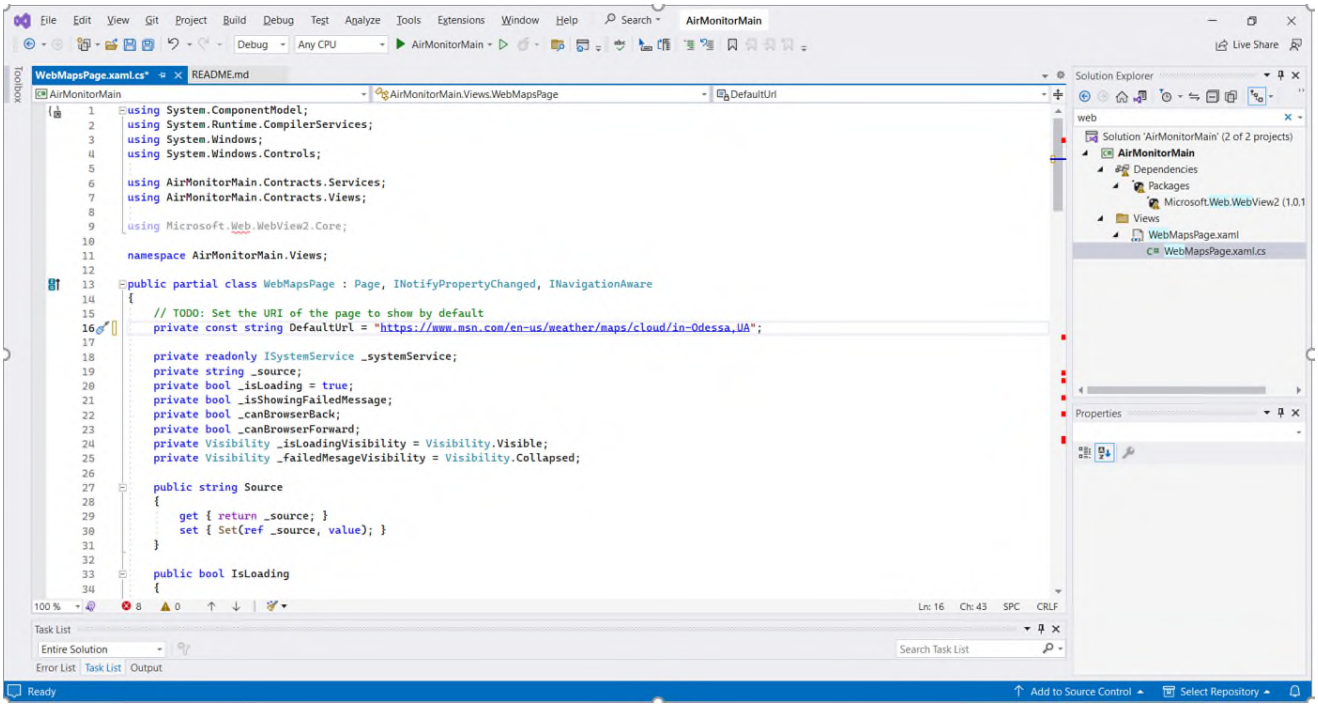

Рисунок 3.6 – Процесу налаштування віджета для відображення мапи Одеси

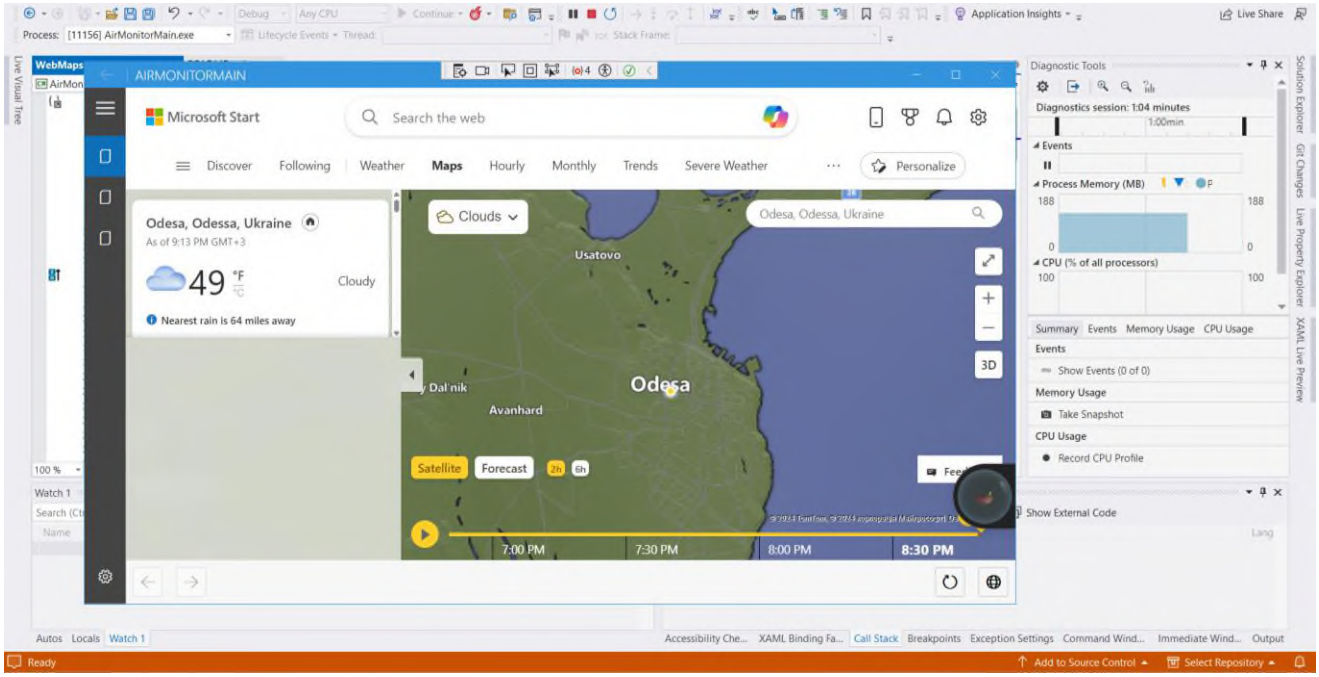

Рисунок 3.7 – Приклад виконання застосунку у середовищі Visual Studio

3.2 Опис інтерфейсу інформаційної системи

В рамках реалізації Windows-застосунку для платформи UWP розроблені такі сторінки: «Головна», «Про проєкт», «Особистий кабінет», «Вхід», «Реєстрація», «Новини», «Завантаження», «Історія», «Зворотний зв’язок».

Головна сторінка ІС відображена на рис. 3.8-3.9. На цій сторінці, як і на всіх інших (крім сторінки реєстрації, авторизації) розміщено блок навігації у вигляді меню, його розташування залежить від роздільної здатності екрана.

Наповнення блоку навігації залежить від того, авторизований користувач чи ні. Якщо користувач не авторизувався, праворуч у верхній частині сторінки, знаходяться посилання на сторінки авторизації та реєстрації, інакше розташовуються посилання на особистий кабінет та вихід із системи.

Ця сторінка містить посилання та віджети:

– кнопку переходу на сторінку хронології;

– віджети містять у собі поточне значення та найменування метрологічного параметра у верхній частині;

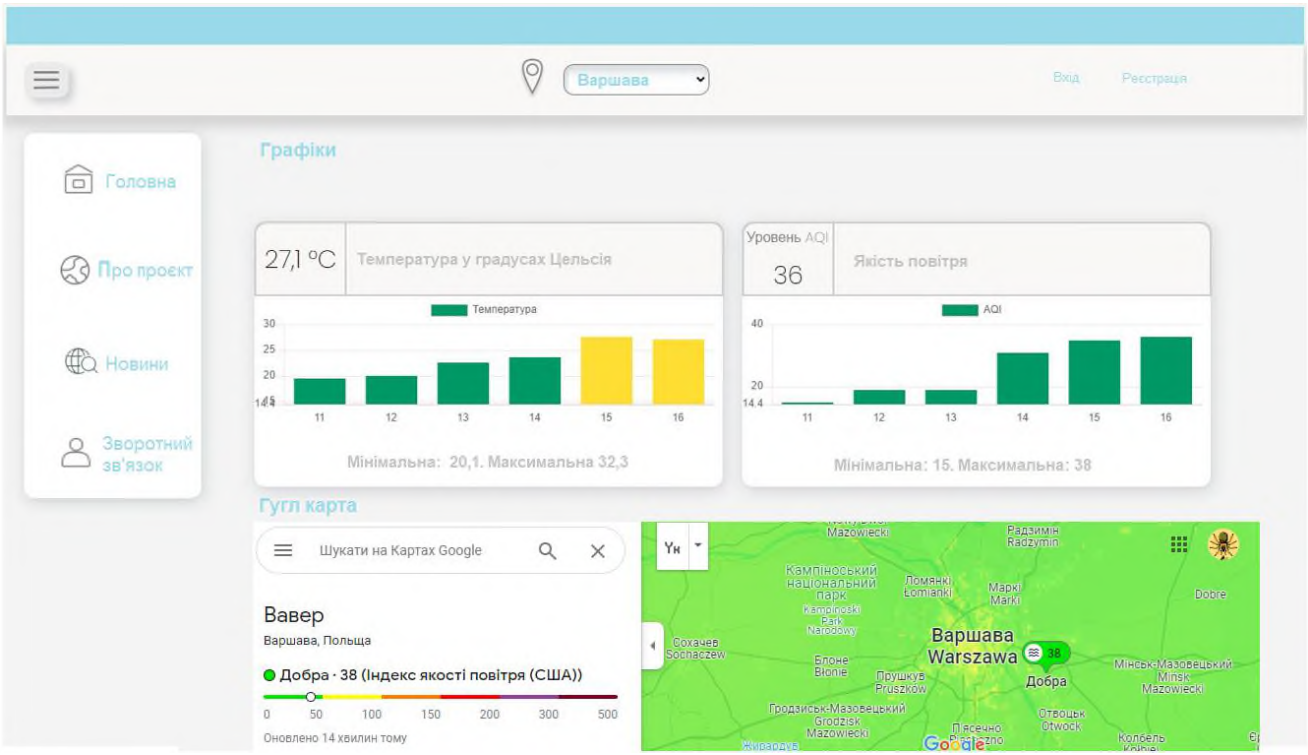

Рисунок 3.8 – Головна сторінка ІС

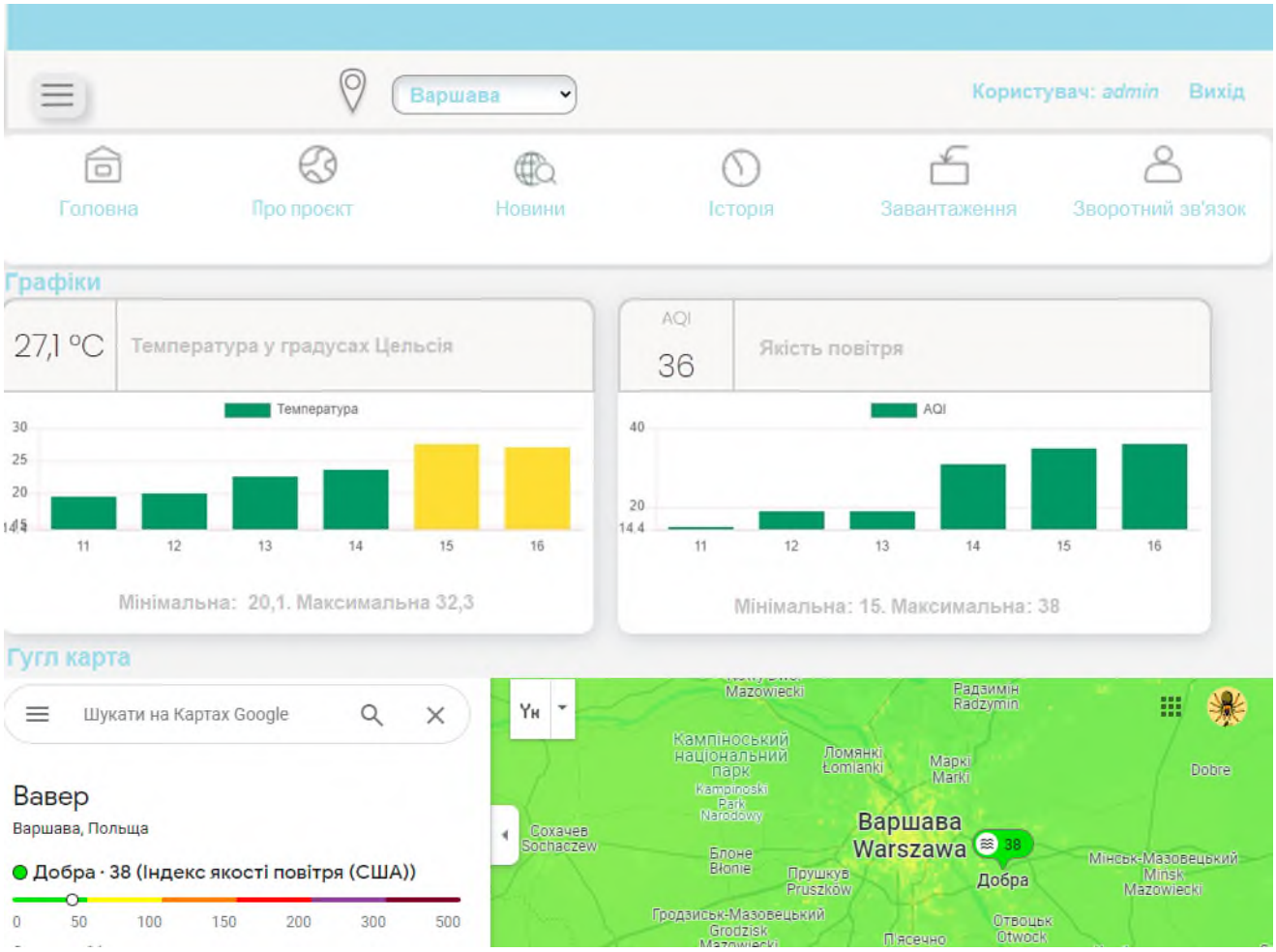

Рисунок 3.9 – Вигляд головної сторінки при зміні масштабу

– у середній частині відображається графік зі значеннями виміру (вісь X – це час у годинах, вісь Y– це значення показника), залежно від кількісної величини вимірювання колонка перефарбовується у той чи інший колір; у нижній частині віджету розташовується коротка інформація за графіком, а саме, максимальне та мінімальне значення;

– карту місцевості, де було знято метеорологічні показники.

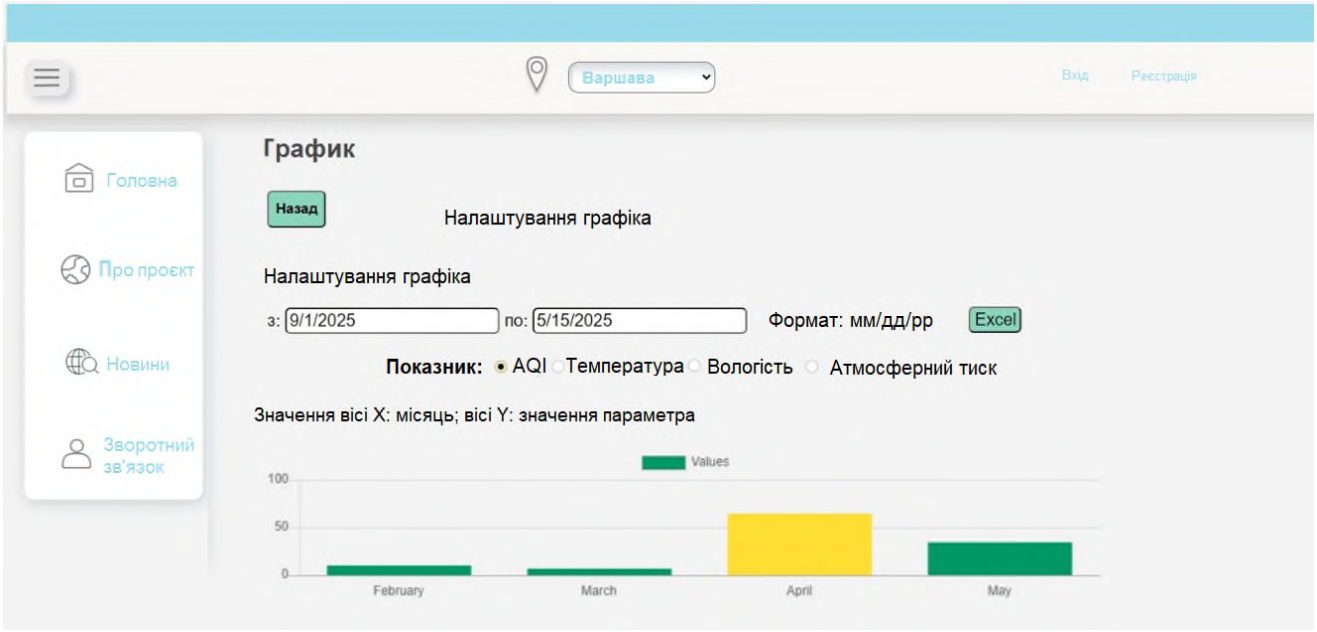

Рисунок 3.10 – Налаштування віджета для демонстрації графіку з метеорологічними показниками

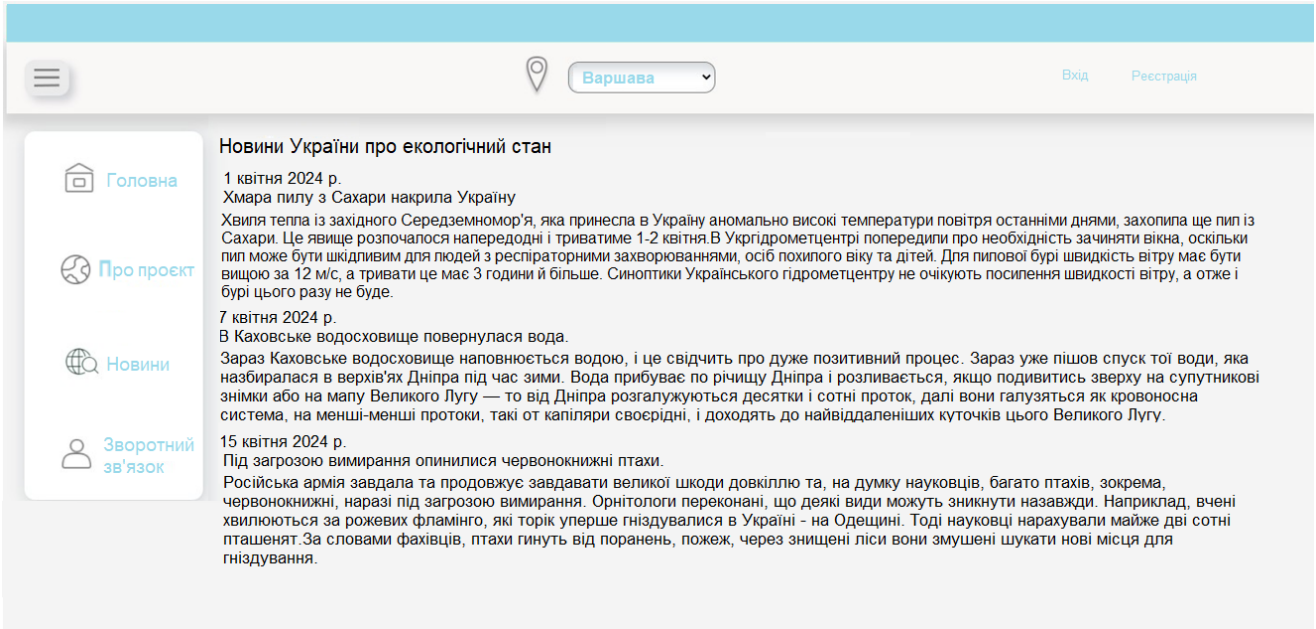

Рисунок 3.11 – Відображення новин з зовнішніх ресурсів

3.3 Модульне тестування

Перевірка коректної роботи деяких окремих модулів інформаційної системи проводилася за допомогою модульних тестів на етапі кодування програми. Після того, як модуль успішно пройшов усі тести, він включається до «основного» рішення. Використання модульних тестів дозволяє виправити помилки на ранніх етапах розробки та краще поринути у кодову базу. На рис. 3.13 наведено перелік виконаних тестів у MS Visual Studio, зокрема перевірка авторизації користувача, введеного пароля, декодування зовнішнього набору даних, вибору міста.

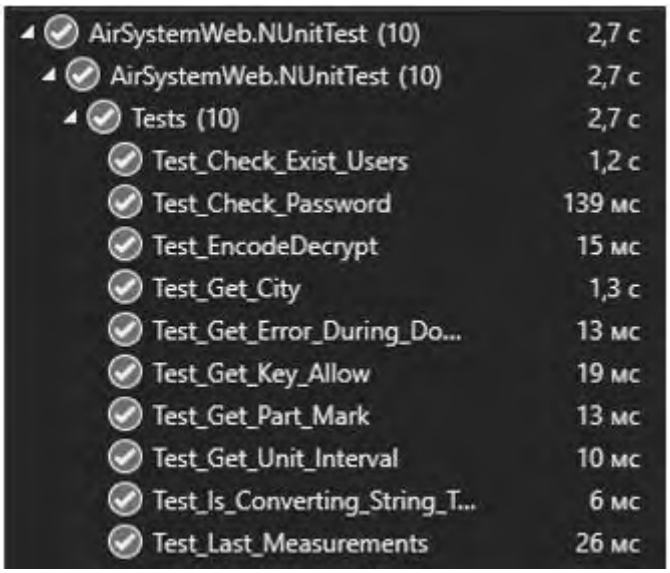

Рисунок 3.12 – Перелік виконаних модульних тестів

Тест "Test_Check_Password" показав, що метод "IsPasswordMeetRequirements" з інтерфейсу "IUserService", реалізований у класі "UserService", який використовується для перевірки пароля на відповідність обов'язкових вимог при реєстрації або його зміні, є успішним.

Вимоги до пароля:

– довжина понад 6 символів;

– мінімум одна велика та одна маленька букви;

– мінімум одна цифра;

– може містити розділові знаки та/або спеціальні символи.

```
public void Test_Check_Password()
{
    UserService service = new UserService(this.GetContext());
    string NicePassword1 = "AAAaabb123";
    string NicePassword2 = "A-B_c-1asdcxc";
    string NicePassword3 = "xwA3dfr445";
    string NicePassword4 = "--sdsdfdfAsdsd3";
    string BadPassword1 = "AAAAaaaa";
    string BadPassword2 = "BBBBBBBB";
    string BadPassword3 = "111111111";
    string BadPassword4 = "Ab432";
    string BadPassword5 = "bbbbbbb";

    Assert.IsTrue(service.IsPasswordMeetRequirements(NicePassword1));
    Assert.IsTrue(service.IsPasswordMeetRequirements(NicePassword2));
    Assert.IsTrue(service.IsPasswordMeetRequirements(NicePassword3));
    Assert.IsTrue(service.IsPasswordMeetRequirements(NicePassword4));
    Assert.IsTrue(!service.IsPasswordMeetRequirements(BadPassword1));
    Assert.IsTrue(!service.IsPasswordMeetRequirements(BadPassword2));
    Assert.IsTrue(!service.IsPasswordMeetRequirements(BadPassword3));
    Assert.IsTrue(!service.IsPasswordMeetRequirements(BadPassword4));
    Assert.IsTrue(!service.IsPasswordMeetRequirements(BadPassword5));
}
```

Рисунок 3.13 – Модульне тестування паролів

3.4 Висновки за розділом

Виконана програмна реалізація веборієнтованої інформаційної системи з використанням мови програмування C# та середовища розробки Visual Studio Community. При розробці використаний мультиплатформовий інструмент Xamarin.Forms, універсальна платформа UWP та майстер Template Studio для WPF. Для відображення карт з метеорологічними показниками використано геосервіс Google Maps API та радарну мапу MSN.

У рамках перевірки коректної роботи інформаційної системи виконано модульне тестування на етапі кодування програми.

Практичною новизною роботи є розробка веборієнтованої інформаційної системи у вигляді мультиплатформного застосунку, за допомогою якого можна інформувати користувача про якість атмосферного повітря вибраного міста України з формуванням метеорологічних міток на онлайн-карті, з можливістю візуалізації динаміки змін навколишнього середовища.

ВИСНОВКИ

В роботі виконано проєктування, програмна реалізація та модульне тестування веборієнтованої інформаційної системи моніторингу параметрів атмосферного повітря.

Проведено аналіз предметної галузі та огляд наявних аналогів. На основі результатів аналізу визначено функціональні вимоги до вебсервісу, вибрано засоби розробки програмного застосунку.

В результаті проектування розроблені: функціонал рівнів користувачів інформаційної системи, логічна модель БД, UML-діаграми ієрархії класів, діаграма варіантів використання ІС, діаграма послідовності внесення адміністратором змін.

Виконана програмна реалізація інформаційної з використанням мови програмування C# та середовища розробки Visual Studio Community. При розробці використаний мультиплатформовий інструмент Xamarin.Forms, універсальна платформа UWP та майстер Template Studio для WPF. Для відображення карт з метеорологічними показниками використано геосервіс Google Maps API та радарну мапу MSN.

У рамках перевірки коректної роботи інформаційної системи виконано модульне тестування на етапі кодування програми, зокрема перевірка авторизації користувача, введеного пароля, декодування зовнішнього набору даних, вибору міста.

Практичною новизною роботи є розробка веборієнтованої інформаційної системи у вигляді мультиплатформного застосунку, за допомогою якого можна інформувати користувача об якості атмосферного повітря вибраного міста України з формуванням метеорологічних міток на онлайн-карті, з можливістю візуалізації динаміки змін навколишнього середовища.

У напрямку подальших досліджень можливе додавання опитування локальних фізичних датчиків для розширення спектру метеорологічних показників, зокрема атмосферного тиску, вологості, радіаційного фону.

Апробація кваліфікаційної роботи виконана на Всеукраїнської науково-практичної конференції здобувачів вищої освіти «Суспільство та держава в умовах воєнного стану: виклики та перспективи» (м. Одеса, 27 квітня 24 р.) та Всеукраїнської науково-технічної конференції «Новітні технології в освіті, науці та виробництві» (м. Луцьк, 25-26 квітня 2024 р.).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ткаченко В.В. Підхід до збору інформації щодо екологічної обстановки при виникненні надзвичайних ситуацій техногенного характеру / В.В. Ткаченко, О.Ю. Чередніченко, М.А. Вовк, С.І. Єршова // Проблеми *інформаційних технологій,* №23, 2108, с. 219-226.

2. Бахарєв В.С. Інформаційно-технологічні аспекти управління екологічною безпекою в системах муніципального моніторингу атмосферного повітря / В.С. Бахарєв, І.В. Шевченко, С.С. Коваль, О.Л. Корцова // *Вісник КрНУ імені Михайла Остроградського.* Випуск 4/2017 (105), с. 68-73.

3. Мокін В.Б. Інформаційна аналітична система моніторингу атмосферного повітря міста Вінниці / В.Б. Мокін, Є.М. Крижановський, В.П. Пінчук // *Матеріали LI Науково-технічної конференції підрозділів ВНТУ (м. Вінниця 31 травня 2022 р.).* Вінниця, ВНТУ, 2022 р.

4. Боголюбов В.М. Моніторинг довкілля: підручник / В.М. Боголюбов, М.О. Клименко, В.Б. Мокін та ін.; за ред. проф. В.М. Боголюбова. Вид. 2-ге, переробл. і доповн. – Київ: НУБіПУ, 2018. – 435 с.

5. Аналітична записка щодо стану та перспектив розвитку державної системи моніторингу довкілля. URL: https://mepr.gov.ua/wp-content/uploads/2023/02/Monitoring-Green-Paper_15_02_2022.pdf.

6. Постанова Кабінету Міністрів України "Про Єдину екологічну платформу “ЕкоСистема”" від 11.10.2021 № 1065. URL: https://zakon.rada.gov.ua/laws/show/1065-2021-п#Text.

7. Earth Observation component of the European Union's space programme. URL: https://www.copernicus.eu/en.

8. Дані про якість повітря в Google. URL: https://support.google.com/maps/answer/11270845?hl=uk-GB.

9. U.S. Environmental Protection Agency | US EPA. URL: https://www.epa.gov/home.

10. Google Maps Platform APIs by Platform. URL: https://developers.google.com/maps/apis-by-platform.

11. Using SQLite.NET. URL: https://learn.microsoft.com/uk-ua/xamarin/android/data-cloud/data-access/using-sqlite-orm.

12. About Open Street Map. URL: https://www.openstreetmap.org/about.

13. SkiaSharp Graphics in Xamarin.Forms. URL: https://learn.microsoft.com/uk-ua/xamarin/xamarin-forms/user-interface/graphics/skiasharp.

14. Xamarin.Forms documentation. URL: https://learn.microsoft.com/uk-ua/xamarin/xamarin-forms.

15. CsvHelper. A .NET library for reading and writing CSV. files. URL: https://joshclose.github.io/CsvHelper.

16. Nuget GoogleMapsApi 1.2.1. URL: https://www.nuget.org/packages/GoogleMapsApi#readme-body-tab.

17. Турчин М.М. Інформаційна система моніторингу параметрів атмосферного повітря / Матеріали Всеукраїнської науково-практичної конференції здобувачів вищої освіти: *Суспільство та держава в умовах воєнного стану: виклики та перспективи (м. Одеса, 27 квітня 2024 р)*. – депоновано.

18. Чикунов П.О. Застосування Xamarin.Forms для розробки мультиплатформових рішень / *Інформаційне суспільство: проблеми та перспективи : матеріали VIII Всеукр. наук.-практич. конф. (м. Одеса, 2 черв. 2023 р.) / відп. ред. Н. І. Логінова*. - Одеса, 2023. - 146 с.

19. Чикунов П.О. Використання SCADA/HMI для моніторингу та керування виробничими процесами // Матеріали VI Міжнародної науково-практичної конференції: *Актуальні тенденції розвитку освіти, науки та технологій : (м. Бахмут, м. Харків, 18 травня 2023 р.). / За заг. ред. Г. Г. Михальченко.* Бахмут – Харків: ННППІ УІПА, 2023. Ч 2. – С. 67-69.

20. Чикунов П.О. Інструментальні засоби програмування Вебдодатків навчального призначення / П.О. Чикунов, А.П. Ларін // Матеріали II Міжнародної науково-практичної конференції*: Актуальні тенденції розвитку освіти, науки та*

технологій (м. Бахмут, 25 квітня 2018 р.). – Т. 1. – Бахмут: ННППІ УІПА, 2018. – С. 95-97.

21. Microsoft Build: MSN Weather. URL: https://learn.microsoft.com/en-us/connectors/msnweather.

22. Чикунов П. О., Турчин М. М. Розробка інформаційної системи для моніторингу параметрів атмосферного повітря / Матеріали І Всеукраїнської науково-технічної конференції: *Новітні технології в освіті, науці та виробництві (м. Луцьк, 25-26 квітня 2024 р.)*. Луцьк, ДонНТУ. – депоновано.

23. Трофименко О. Г., Логінова Н. І., Манаков С. Ю. Методичні вказівки до виконання кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти за спеціальностями 121 «Інженерія програмного забезпечення» та 122 «Комп’ютерні науки» [Електронне видання] / О. Г. Трофименко, Н. І. Логінова, С. Ю. Манаков. – Одеса, 2024. – 39 с. Режим доступу: https://hdl.handle.net/11300/27211.

ДОДАТКИ

Додаток А. Програмний код

Лістинг файлу MainPage.xaml

```
<Page
    x:Class="AirMonitorMain.Views.MainPage"
    Style="{DynamicResource MahApps.Styles.Page}"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:properties="clr-namespace:AirMonitorMain.Properties"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">
    <Page.Resources>
        <DataTemplate x:Key="ItemTemplate">
            <Grid Height="60">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto" />
                    <ColumnDefinition Width="*" />
                </Grid.ColumnDefinitions>
                <TextBlock
                    Grid.Column="0"
                    Text="{Binding Symbol}"
                    Style="{StaticResource MediumIconStyle}"
                    Margin="{StaticResource XSmallLeftMargin}"
                    AutomationProperties.Name="List item icon" />
                <StackPanel
                    Grid.Column="1"
                    Margin="{StaticResource SmallLeftMargin}"
                    VerticalAlignment="Center">
                    <TextBlock Style="{StaticResource ListTitleStyle}" Text="{Binding
Company}" />
                    <TextBlock Style="{StaticResource ListSubTitleStyle}" Text="{Binding
Status}" />
                </StackPanel>
            </Grid>
        </DataTemplate>

        <DataTemplate x:Key="DetailTemplate">
            <ScrollViewer
                Name="ForegroundElement"
                IsTabStop="True"
                HorizontalAlignment="Stretch">
                <StackPanel HorizontalAlignment="Left" Margin="{StaticResource
MediumLeftTopRightBottomMargin}">
                    <StackPanel Orientation="Horizontal" HorizontalAlignment="Stretch">
                        <TextBlock
                            Text="{Binding Symbol}"
                            Style="{StaticResource BigIconStyle}"
                            AutomationProperties.Name="Detail icon" />
                        <TextBlock
                            Margin="{StaticResource SmallLeftMargin}"
                            VerticalAlignment="Center"
                            Text="{Binding Company, Mode=OneWay}"
                            Style="{StaticResource PageTitleStyle}" />
                    </StackPanel>
                    <StackPanel Margin="{StaticResource SmallTopMargin}">
                        <TextBlock
                            Style="{StaticResource SubtitleTextStyle}"
                            Text="Головна" />
                        <TextBlock
                            Style="{StaticResource BodyTextStyle}"
                            Text="{Binding Status, Mode=OneWay}" />
```

```
                            Style="{StaticResource SubtitleTextStyle}"
                            Text="Про проєкт" />
                        <TextBlock
                            Style="{StaticResource BodyTextStyle}"
                            Text="{Binding OrderDate, Mode=OneWay}" />
                        <TextBlock
                            Margin="{StaticResource SmallTopMargin}"
                            Style="{StaticResource SubtitleTextStyle}"
                            Text="Новини" />
                        <TextBlock
                            Style="{StaticResource BodyTextStyle}"
                            Text="{Binding Company, Mode=OneWay}" />
                        <TextBlock
                            Margin="{StaticResource SmallTopMargin}"
                            Style="{StaticResource SubtitleTextStyle}"
                            Text="Історія" />
                        <TextBlock
                            Style="{StaticResource BodyTextStyle}"
                            Text="{Binding ShipTo, Mode=OneWay}" />
                        <TextBlock
                            Margin="{StaticResource SmallTopMargin}"
                            Style="{StaticResource SubtitleTextStyle}"
                            Text="Завантаження" />
                        <TextBlock
                            Style="{StaticResource BodyTextStyle}"
                            Text="{Binding OrderTotal, Mode=OneWay}" />

                        <TextBlock
                            Margin="{StaticResource MediumTopMargin}"
                            Style="{StaticResource SubtitleTextStyle}"
                            Text="Зворотний зв'язок" />

                    </StackPanel>
                </StackPanel>
            </ScrollViewer>
        </DataTemplate>
    </Page.Resources>
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition MinWidth="180" MaxWidth="300" Width="*" />
            <ColumnDefinition Width="2*" />
        </Grid.ColumnDefinitions>
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="48" />
                <RowDefinition Height="*" />
            </Grid.RowDefinitions>
            <TextBlock
                Style="{StaticResource PageTitleStyle}"
                Margin="{StaticResource MediumLeftMargin}"
                Text="{x:Static properties:Resources.ForecastPageTitle}" />
            <ListView
                Grid.Row="1"
                ItemsSource="{Binding SampleItems}"
                ItemTemplate="{StaticResource ItemTemplate}"
                SelectedItem="{Binding Selected, Mode=TwoWay}" />
        </Grid>
        <ContentControl
            Grid.Column="1"
            IsTabStop="False"
            Content="{Binding Selected}"
            ContentTemplate="{StaticResource DetailTemplate}" />
    </Grid>
</Page>
```

Лістинг файлу MainPage.xaml.cs

```
using System.ComponentModel;
using System.Runtime.CompilerServices;
using System.Windows.Controls;

namespace AirMonitorMain.Views;

public partial class MainPage : Page, INotifyPropertyChanged
{
    public MainPage()
    {
        InitializeComponent();
        DataContext = this;
    }

    public event PropertyChangedEventHandler PropertyChanged;

    private void Set<T>(ref T storage, T value, [CallerMemberName] string propertyName =
null)
    {
        if (Equals(storage, value))
        {
            return;
        }

        storage = value;
        OnPropertyChanged(propertyName);
    }

    private void OnPropertyChanged(string propertyName) => PropertyChanged?.Invoke(this,
new PropertyChangedEventArgs(propertyName));
}
```

Лістинг файлу WebMapsPage.xaml

```
<Page
    x:Class="AirMonitorMain.Views.WebMapsPage"
    Style="{DynamicResource MahApps.Styles.Page}"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:properties="clr-namespace:AirMonitorMain.Properties"
    xmlns:wv2="clr-
namespace:Microsoft.Web.WebView2.Wpf;assembly=Microsoft.Web.WebView2.Wpf"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">
    <Page.InputBindings>
        <KeyBinding Key="F5" Command="{Binding RefreshCommand}" />
        <KeyBinding Modifiers="Alt" Key="Left" Command="{Binding BrowserBackCommand}" />
        <KeyBinding Modifiers="Alt" Key="Right" Command="{Binding
BrowserForwardCommand}" />
        <KeyBinding Modifiers="Ctrl" Key="T" Command="{Binding OpenInBrowserCommand}" />
    </Page.InputBindings>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="Auto" />
        </Grid.RowDefinitions>
        <wv2:WebView2
            x:Name="webView"
            Source="{Binding Source}"
            NavigationCompleted="OnNavigationCompleted"/>
        <Grid
```

```
            VerticalAlignment="Bottom"
            Grid.Row="1"
            Background="{DynamicResource MahApps.Brushes.Gray10}">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto" />
                <ColumnDefinition Width="*" />
                <ColumnDefinition Width="Auto" />
            </Grid.ColumnDefinitions>
            <StackPanel Grid.Column="0" Orientation="Horizontal">
                <Button
                    Click="OnBrowserBack"
                    IsEnabled="{Binding CanBrowserBack}"
                    Style="{StaticResource WebViewButtonStyle}"
                    ToolTip="{x:Static
properties:Resources.WebMapsPageBackButtonTooltip}">
                    <TextBlock
                        Style="{StaticResource SmallIconStyle}"
                        Text="&#xE72B;" />
                </Button>
                <Button
                    Click="OnBrowserForward"
                    IsEnabled="{Binding CanBrowserForward}"
                    Style="{StaticResource WebViewButtonStyle}"
                    ToolTip="{x:Static
properties:Resources.WebMapsPageForwardButtonTooltip}">
                    <TextBlock
                        Style="{StaticResource SmallIconStyle}"
                        Text="&#xE72A;" />
                </Button>
            </StackPanel >

            <TextBlock
                Grid.Column="1"
                VerticalAlignment="Center"
                Style="{StaticResource BodyTextStyle}"
                Margin="{StaticResource SmallLeftMargin}"
                Visibility="{Binding IsLoadingVisibility}"
                Text="{x:Static properties:Resources.WebMapsPageLoadingText}" />
            <TextBlock
                Grid.Column="1"
                VerticalAlignment="Center"
                Style="{StaticResource BodyTextStyle}"
                Margin="{StaticResource SmallLeftMargin}"
                Visibility="{Binding FailedMesageVisibility}"
                Text="{x:Static properties:Resources.WebMapsPageFaildedMessage}" />
            <StackPanel Grid.Column="2" Orientation="Horizontal">
                <Button
                    Click="OnRefresh"
                    Style="{StaticResource WebViewButtonStyle}"
                    ToolTip="{x:Static
properties:Resources.WebMapsPageRefreshButtonTooltip}"
                    AutomationProperties.Name="{x:Static
properties:Resources.WebMapsPageRefreshButtonTooltip}">
                    <TextBlock
                        Style="{StaticResource SmallIconStyle}"
                        Text="&#xE72C;" />
                </Button>
                <Button
                    Click="OnOpenInBrowser"
                    Style="{StaticResource WebViewButtonStyle}"
                    ToolTip="{x:Static
properties:Resources.WebMapsPageOpenInBrowserButtonTooltip}"
                    AutomationProperties.Name="{x:Static
properties:Resources.WebMapsPageOpenInBrowserButtonTooltip}">
                    <TextBlock
```

```
                        Style="{StaticResource SmallIconStyle}"
                        Text="&#xE774;" />
                </Button>
            </StackPanel>
        </Grid>
    </Grid>
</Page>
```

Лістинг файлу WebMapsPage.xaml.cs

```
using System.ComponentModel;
using System.Runtime.CompilerServices;
using System.Windows;
using System.Windows.Controls;

using AirMonitorMain.Contracts.Services;
using AirMonitorMain.Contracts.Views;

using Microsoft.Web.WebView2.Core;

namespace AirMonitorMain.Views;

public partial class WebMapsPage : Page, INotifyPropertyChanged, INavigationAware
{
    // TODO: Set the URI of the page to show by default
    private const string DefaultUrl = "https://www.msn.com/en-us/weather/maps/cloud/in-
Odessa,UA";

    private readonly ISystemService _systemService;
    private string _source;
    private bool _isLoading = true;
    private bool _isShowingFailedMessage;
    private bool _canBrowserBack;
    private bool _canBrowserForward;
    private Visibility _isLoadingVisibility = Visibility.Visible;
    private Visibility _failedMesageVisibility = Visibility.Collapsed;

    public string Source
    {
        get { return _source; }
        set { Set(ref _source, value); }
    }

    public bool IsLoading
    {
        get => _isLoading;
        set
        {
            Set(ref _isLoading, value);
            IsLoadingVisibility = value ? Visibility.Visible : Visibility.Collapsed;
        }
    }

    public bool IsShowingFailedMessage
    {
        get => _isShowingFailedMessage;
        set
        {
            Set(ref _isShowingFailedMessage, value);
            FailedMesageVisibility = value ? Visibility.Visible : Visibility.Collapsed;
        }
    }

    public bool CanBrowserBack
    {
```

```
        get { return _canBrowserBack; }
        set { Set(ref _canBrowserBack, value); }
    }

    public bool CanBrowserForward
    {
        get { return _canBrowserForward; }
        set { Set(ref _canBrowserForward, value); }
    }

    public Visibility IsLoadingVisibility
    {
        get { return _isLoadingVisibility; }
        set { Set(ref _isLoadingVisibility, value); }
    }

    public Visibility FailedMesageVisibility
    {
        get { return _failedMesageVisibility; }
        set { Set(ref _failedMesageVisibility, value); }
    }

    public WebMapsPage(ISystemService systemService)
    {
        _systemService = systemService;
        Source = DefaultUrl;
        InitializeComponent();
        DataContext = this;
    }

    public void OnNavigatedTo(object parameter)
    {
    }

    public void OnNavigatedFrom()
    {
    }

    private void OnBrowserBack(object sender, RoutedEventArgs e)
        => webView.GoBack();

    private void OnBrowserForward(object sender, RoutedEventArgs e)
        => webView.GoForward();

    private void OnRefresh(object sender, RoutedEventArgs e)
    {
        IsShowingFailedMessage = false;
        IsLoading = true;
        webView.Reload();
    }

    private void OnOpenInBrowser(object sender, RoutedEventArgs e)
        => _systemService.OpenInWebBrowser(Source);

    private void OnNavigationCompleted(object sender,
CoreWebView2NavigationCompletedEventArgs e)
    {
        IsLoading = false;
        if (e != null && !e.IsSuccess)
        {
            // Use `e.WebErrorStatus` to vary the displayed message based on the error
reason
            IsShowingFailedMessage = true;
        }
```

```
        CanBrowserBack = webView.CanGoBack;
        CanBrowserForward = webView.CanGoForward;
    }

    public event PropertyChangedEventHandler PropertyChanged;

    private void Set<T>(ref T storage, T value, [CallerMemberName] string propertyName =
null)
    {
        if (Equals(storage, value))
        {
            return;
        }

        storage = value;
        OnPropertyChanged(propertyName);
    }

    private void OnPropertyChanged(string propertyName) => PropertyChanged?.Invoke(this,
new PropertyChangedEventArgs(propertyName));
}
```

Додаток Б. Апробація результатів роботи

Турчин Михайло Михайлович
студент 4-го курсу факультету кібербезпеки
та інформаційних технологій
Національного університету «Одеська юридична академія»

Інформаційна система моніторингу параметрів атмосферного повітря

Система моніторингу екологічної обстановки – це комплекс технологій, методів і засобів, що використовуються для постійного контролю за станом довкілля, включаючи якість повітря, води, ґрунтів, рослинності та інші параметри навколишнього середовища. До основних компонент системи моніторингу можна віднести датчики та вимірювальні прилади, системи збору та передачі даних, програмні модулі аналізу та обробки даних, системи візуалізації та звітності, системи сповіщення та аварійного реагування. Подібні системи моніторингу є важливим інструментом для збереження природних ресурсів, здоров'я людей та загального благополуччя суспільства [1, с. 220].

Будь-яка система моніторингу екологічної обстановки обов'язково має накопичувати інформацію зі знятих показань із різних джерел, які мають достатню точність, для надання коректних результатів [2, с. 69].

Більшість комплексів моніторингу акцентують свою увагу найбільш поширених і небезпечних забруднюючих довкілля речовинах. Окремо можна відокремити системи контролю якості повітря з відстеженням відсотку оксиду вуглецю, азоту, діоксиду сірки, озону, пилу.

Метою дослідження є проектування, розробка та тестування веборієнтованої інформаційної системи (ІС) для моніторингу параметрів атмосферного повітря, здатної отримувати регулярну, своєчасну та достовірну інформацію, проводити оцінку зібраних даних, надавати необхідну інформацію у зручному форматі кінцевому користувачеві.

Об'єктами дослідження є методи та способи моніторингу параметрів повітря, способи реєстрації даних за допомогою метеорологічного приладу та візуалізації отриманих даних у вигляді метеорологічних міток на онлайн-карті місцевості карті.

Проведення оцінки параметрів навколишнього середовища потребує виявлення та відстеження конкретного характеру забруднення. Отримання подібної інформації дозволяє отримати ступінь впливу на здоров'я населення та основу для ретельно продуманих рішень. Зазвичай збір даних проводиться у стаціонарних метеостанціях та пересувних постах. Контроль якості повітря проводиться у хіміко-аналітичній лабораторії. Обробка, зберігання та аналіз даних відбувається в інформаційно-аналітичних центрах [3].

Аналіз предметної області, наявних аналогів інформаційної системи моніторингу повітря [1-3] дозволив виконати постановку функціональних та нефункціональних вимог до інформаційної системи:

– можливість доопрацювання, підтримки протягом усього життєвого циклу ІС (виправлення помилок, збільшення функціоналу);

– повинен працювати у всіх популярних веббраузерах;

– надійність: стабільна робота ІС та підтримка підказок користувачеві;

– відмовостійкість: під час виникнення помилки система повинна гарантувати працездатний стан;

– швидкодія: робота у ІС, починаючи від переходу між сторінками, закінчуючи розрахунками та виведенням інформації, повинна виконуватися за прийнятний час;

– ергономічність: експлуатація ІС повинна бути інтуїтивно зрозуміла і зручна, для всіх кінцевих користувачів, не враховуючи їх кваліфікацію.

Дослідження предметної області дозволило виділити три ролі користувачів інформаційної системи: адміністратор, авторизований та гість (неавторизований користувачі).

Зокрема, адміністратор повинен мати можливість:

– затверджувати/відкидати користувальницькі завантаження та вносити зміни до найменування станції та міста, в яких проводилися вимірювання;

– переглядати карту, графіки та статистику, що відображають актуальний стан та динаміку у часовому інтервалі навколишнього середовища;

– завантажити дані вибраного екологічного показника у заданому діапазоні;

– додавати/редагувати та видаляти новини;

– завантажувати датасет із показаннями датчика;

– переглядати історію своїх завантажень та інших користувачів;

– отримувати доступ до особистого кабінету з можливістю редагувати особисту інформацію;

– переглядати інформацію про проект, контактну інформацію та новини;

– редагувати/вилучати/створювати новини;

– вийти з системи.

Функціонал всіх перерахованих користувачів продемонстровано на рис. 1.

Одним із основних моментів у процесі розробки та контролю роботи програмного продукту є система логування. Протокол ІС працює для всіх користувачів з можливістю авторизації у системі та веденням журналу дій для кожного авторизованого користувача.

Реалізація проекту повинна виконуватися за допомогою сучасних інструментів, здатних ефективно та повною мірою вирішити те чи інше завдання. Виконано вибір засобів розробки ІС. Для розробки серверної частини ІС вибрано мову програмування C#, багатофункціональну платформу ASP.NET Core MVC, яка надає розробнику інтелектуальний обробник коду динамічних вебсторінок Razor. Для написання клієнтської частині вибрано бібліотеку jQuery разом з бібліотекою jQuery-UI, а також бібліотеку Chart.js для створення графіків. Для зберігання даних ІС як СКБД обрано MS SQLServer. Для розробки мікропрограмного забезпечення спеціалізованого пристрою на платформі Arduino обрана мова C++ з бібліотекою AVR Libc.

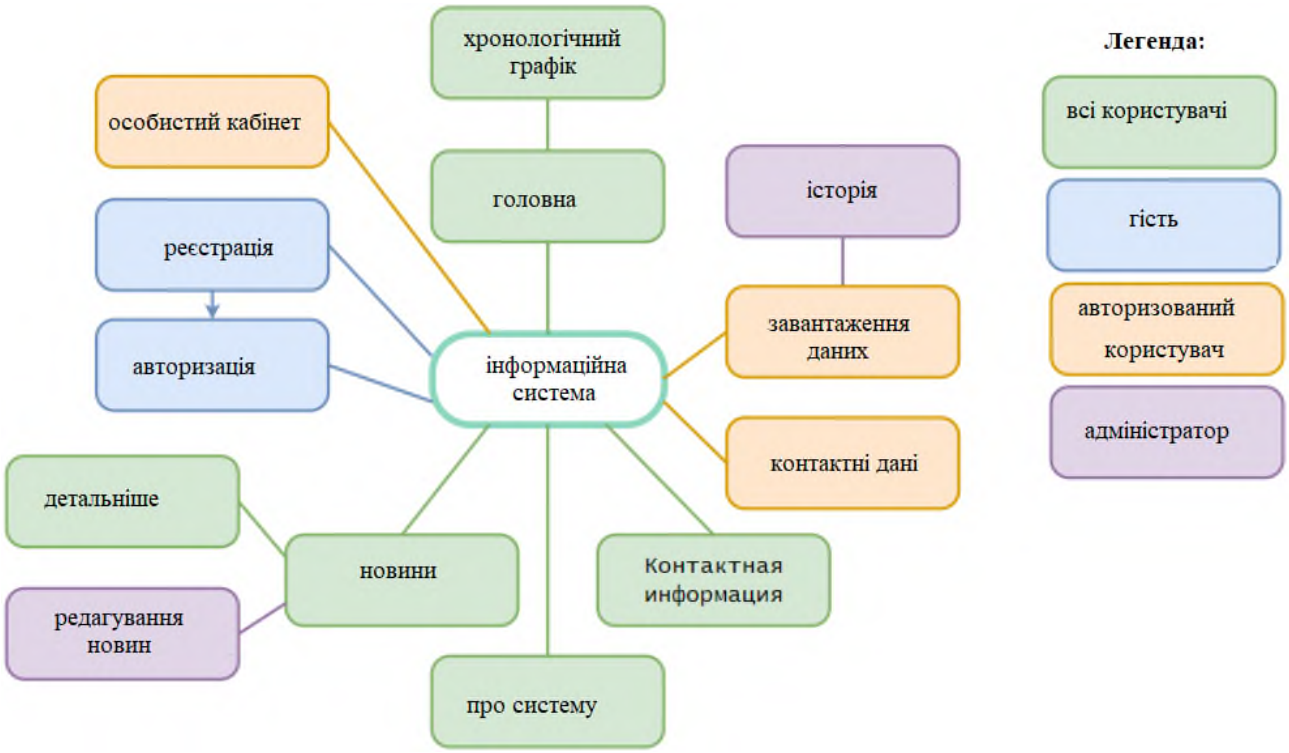

Рис. 1. Функціонал всіх рівнів користувачів інформаційної системи

Список використаних джерел

1. Ткаченко В.В. Підхід до збору інформації щодо екологічної обстановки при виникненні надзвичайних ситуацій техногенного характеру / В.В. Ткаченко, О.Ю. Чередніченко, М.А. Вовк, С.І. Єршова // *Проблеми інформаційних технологій,* №23, 2108, с. 219-226.

2. Бахарєв В.С. Інформаційно-технологічні аспекти управління екологічною безпекою в системах муніципального моніторингу атмосферного повітря / В.С. Бахарєв, І.В. Шевченко, С.С. Коваль, О.Л. Корцова // *Вісник КрНУ імені Михайла Остроградського.* Випуск 4/2017 (105), с. 68-73.

3. Мокін В.Б. Інформаційна аналітична система моніторингу атмосферного повітря міста Вінниці / В.Б. Мокін, Є.М. Крижановський, В.П. Пінчук // *Матеріали LI Науково-технічної конференції підрозділів ВНТУ (м. Вінниця 31 травня 2022 р.).* Вінниця, ВНТУ, 2022 р.

Ключові слова: інформаційна система, моніторинг, функціональні вимоги, рівні користувачів.
Key words: information system, monitoring, functional requirements, levels of users.

Науковий керівник: к.т.н., доцент Чикунов П.О.

Чикунов П. О., канд. техн. наук, **Турчин М. М.,** бакалавр,
Національний університет «Одеська юридична академія», м. Одеса, Україна

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ МОНІТОРИНГУ ПАРАМЕТРІВ АТМОСФЕРНОГО ПОВІТРЯ

Система моніторингу екологічної обстановки – це комплекс технологій, методів і засобів, що використовуються для постійного контролю за станом довкілля. Система моніторингу має накопичувати інформацію із різних джерел, які мають достатню точність. Для надання послуг населенню щодо моніторингу параметрів атмосферного повітря, необхідна інформаційна система для у вигляді Windows-застосунку, з візуалізацією показників якості повітря у вигляді мапи та графіків.

Ключові слова: інформаційна система, моніторинг, вебдодаток.

На даний момент існує низка геоінформаційних проектів, здатних виконувати моніторинг довкілля та надавати актуальні метеорологічні дані, а також формувати рекомендації для користувачів [1]. В Європейському Союзі діє глобальна програма моніторингу стану планети Copernicus. Її мета полягає в наданні інформації про якість повітря та склад атмосфери як у межах, так і поза межами Європи, використовуючи дані з супутникових і наземних спостережень в поєднанні з моделями прогнозування.

В федеральному уряді США створено Федеральну агенцію з охорони навколишнього середовища (EPA), яке відповідає за розробку та впровадження стандартів якості повітря, а також за контроль і оцінку рівнів забруднення повітря. Велика кількість моніторингових станцій вимірює параметри якості повітря, збирає дані, далі вони аналізуються і використовуються для оцінки загального стану повітря та розробки заходів щодо його поліпшення.

Закон України «Про охорону навколишнього природного середовища» встановлює створення системи державного моніторингу навколишнього природного середовища та проведення спостережень за його станом і рівнем забруднення [2]. Наразі деякі міста та облдержадміністрації власноруч впроваджують регіональні системи моніторингу довкілля. У Київській області, місті Києві та декількох містах Дніпропетровської області вже запроваджено комунальні мережі спостережень і автоматичні системи моніторингу. Проте наразі ці мережі не координовані, обладнання є різнорідним, методи збору даних та показники не узгоджені з Міндовкіллям, що ускладнює їх використання на національному рівні.

Метою дослідження є проектування, розробка та тестування інформаційної системи (ІС) для моніторингу параметрів атмосферного повітря, здатної отримувати регулярну, своєчасну та достовірну інформацію, проводити оцінку зібраних даних, надавати необхідну інформацію у зручному форматі кінцевому користувачеві.

Реалізація проекту повинна виконуватися за допомогою сучасних інструментів, здатних ефективно та повною мірою вирішити те чи інше завдання. Для реалізації ІС проведено аналіз предметної галузі та огляд наявних аналогів. На основі результатів аналізу було визначено функціональні вимоги до вебсервісу,

вибрано засоби розробки серверної та клієнтської частин вебсервісу, а також комплектуючі для спеціалізованого метеорологічного пристрою.

В результаті виконання дослідження спроектовано логічну модель бази даних, створено UML-діаграми варіантів використання та класів, а також карту вебсервісу. Для представлення етапів роботи деяких модулів було створено UML-діаграми діяльності та послідовності.

Програмна реалізація ІС виконана за допомогою сучасних інструментів, здатних ефективно та повною мірою вирішити те чи інше завдання. Для розробки серверної частини обрана сучасна мова програмування C# та IDE Visual Studio Community. Xamarin – це відкрита платформа, яка дозволяє розробляти продуктивні додатки для iOS, Android та Windows, з використанням спільного коду. Завдяки Xamarin, значна частина коду може бути використана на різних платформах, що дозволяє розробникам писати бізнес-логіку додатка однією мовою (XAML та C#), а отримувати характеристики, що специфічні для кожної платформи. Програми Xamarin використовують бібліотеку BCL, яка надає доступ до тисяч бібліотек з додатковими функціями.

Для реалізації інформаційної системи обрано платформу Xamarin.Forms разом з NuGet-пакетами SQLite.NET для організації бази даних та SkiaSharp для побудови графіків, CsvHelper для перетворення наборів даних у CSV-форматі у XML-формат, GoogleMapsApi для обгортки вебслужб «Карти Google» для платформи .NET.

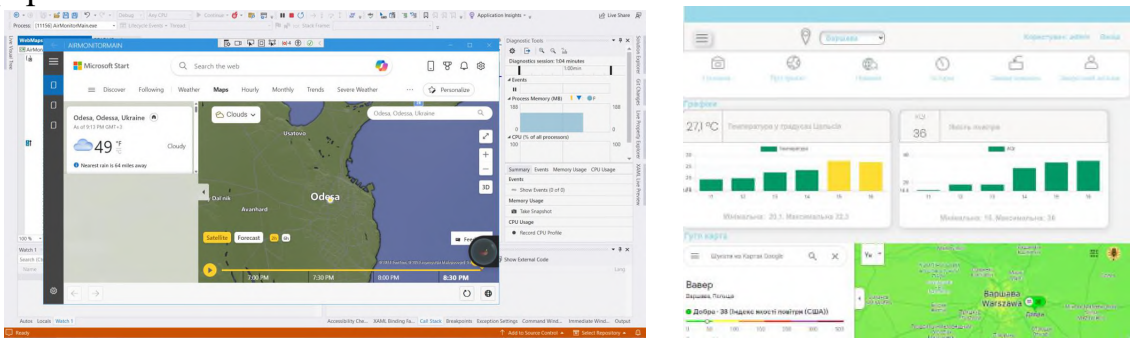

Рисунок 1 – а) процес розробки ІС у Visual Studio, б) вигляд головної сторінки ІС

У рамках реалізації ІС виконано реалізацію: особистого кабінету користувача, завантаження наборів даних зі зовнішніх ресурсів, відображення карти місцевості, що відображає мітки з значеннями показників, формування віджетів, що описують актуальний стан та динаміку змін навколишнього середовища.

У напрямку подальших досліджень можливе додавання можливості опитування зовнішніх фізичних датчиків для розширення спектру метеорологічних показників, зокрема атмосферного тиску, вологості, радіаційного фону.

Список використаної літератури

1. Аналітична записка щодо стану та перспектив розвитку державної системи моніторингу довкілля. URL: https://mepr.gov.ua/wp-content/uploads/2023/02/Monitoring-Green-Paper_15_02_2022.pdf.

2. Постанова Кабінету Міністрів України "Про Єдину екологічну платформу “ЕкоСистема”" від 11.10.2021 № 1065. URL: https://zakon.rada.gov.ua/laws/show/1065-2021-п#Text.