

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

**Факультет інформаційних технологій**

**ПОГОДЖЕНО**  
**Декан факультету**

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**  
**Завідувач кафедри інформаційних систем і  
технологій**

\_\_\_\_\_ **Ігор**  
**БОЛБОТ**

\_\_\_\_\_ **Михайло ШВИДЕНКО**

«\_\_\_» \_\_\_\_\_ 2026 «\_\_\_» \_\_\_\_\_ 2026 р.  
р.

**БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**на тему:** «Розробка системи збору відгуків та формування рейтингу риболовних локацій»

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

**Гарант освітньої програми**  
д.е.н., професор

\_\_\_\_\_ **Роман**  
**РУДЕНСЬКИЙ**

**Керівник бакалаврської  
кваліфікаційної роботи**  
д.ф. з к.н.

\_\_\_\_\_ **Роман**  
**ЗОЛОТУХА**  
(підпис)

**Виконав**

\_\_\_\_\_ **Максим**  
**Бикодір**

**Київ-2026**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ  
УКРАЇНИ  
Факультет інформаційних технологій**

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**  
інформаційних систем і технологій  
(назва кафедри)

к.т.н., доцент Швиденко М.З  
(науковий ступінь, вчене звання) (підпис) (ПІБ)  
“ ” 2026 р.

**З А В Д А Н Я**  
**на виконання бакалаврської кваліфікаційної роботи студенту**  
Бикодіра Максима Ярославовича

Спеціальність 122 – «Комп’ютерні науки»

Освітньо-професійна програма: «Комп’ютерні науки»

Тема бакалаврської кваліфікаційної роботи «Розробка системи збору відгуків та формування рейтингу риболовних локацій» затверджена наказом ректора НУБіП України від “6” січня 2026 № 46 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проекту):

Предметна область риболовних локацій; дані про користувачів, локації, відгуки та оцінки; вимоги до каталогу, рейтингу, пошуку й Google Maps; засоби розробки C#, Windows Forms, SQLite.

**Перелік питань, які потрібно розробити:**

1. Проаналізувати предметну область та існуючі рішення для збору відгуків і рейтингування риболовних локацій.
2. Спроекувати структуру бази даних для зберігання користувачів, локацій, відгуків і оцінок.
3. Розробити модулі авторизації, каталогу локацій, додавання відгуків і формування рейтингу.
4. Реалізувати перехід до місцезнаходження локації через Google Maps.

**Дата видачі завдання “6” Січня 2026 р.**

**Керівник бакалаврської кваліфікаційної роботи**

PhD старший викладач Золотуха Р.А.  
(науковий ступінь та вчене звання) (підпис) (ПІБ)

**Завдання прийняв до виконання**

Бикодір М.Я.  
(підпис) (ПІБ студента)

# ЗМІСТ

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| <b>ВСТУП.....</b>                                                     | <b>4</b>  |
| <b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>                               | <b>7</b>  |
| 1.1 Опис предметної області .....                                     | 7         |
| 1.2 Огляд існуючих рішень .....                                       | 9         |
| 1.3 Постановка завдання.....                                          | 14        |
| 1.4 Функціональні та нефункціональні вимоги .....                     | 16        |
| 1.5 Вимоги до інтерфейсу користувача .....                            | 17        |
| <b>2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>                          | <b>19</b> |
| 2.1 Загальні відомості .....                                          | 19        |
| 2.2 Об'єктне та функціональне моделювання .....                       | 21        |
| 2.3 Абстракції предметної області.....                                | 34        |
| 2.4 Діаграма класів .....                                             | 35        |
| <b>3 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ.....</b>                         | <b>40</b> |
| 3.1 Логічна модель даних.....                                         | 40        |
| 3.2 Вибір системи управління базою даних та її реалізація.....        | 44        |
| 3.3 Архітектура програмного забезпечення .....                        | 51        |
| 3.4 Організаційна структура програмного забезпечення .....            | 56        |
| 3.5 Вибір інструментарію для створення програмного забезпечення ..... | 61        |
| <b>4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ.....</b>                    | <b>64</b> |
| 4.1 Вимоги до апаратного та програмного забезпечення .....            | 64        |
| 4.2 Склад інсталяційного пакету та порядок запуску програми .....     | 68        |
| 4.3 Тестування системи .....                                          | 71        |
| <b>ВИСНОВКИ .....</b>                                                 | <b>88</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>                                | <b>91</b> |
| <b>ДОДАТОК А.....</b>                                                 | <b>93</b> |
| <b>ДОДАТОК Б.....</b>                                                 | <b>94</b> |
| <b>ДОДАТОК В.....</b>                                                 | <b>95</b> |
| <b>ДОДАТОК Г .....</b>                                                | <b>96</b> |

## ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій цифрові сервіси дедалі активніше впроваджуються у сферу туризму, відпочинку та дозвілля. Одним із популярних видів активного відпочинку є риболовля, для якої важливе значення має наявність актуальної інформації про риболовні локації, умови перебування, рівень комфорту та реальні відгуки інших користувачів. Більшість рибалок здійснюють пошук відповідних місць через соціальні мережі, форуми або окремі вебресурси, однак така інформація часто є неповною, неструктурованою або застарілою.

У зв'язку з цим виникає необхідність створення програмного забезпечення, яке забезпечить централізований збір інформації про риболовні локації, надання можливості користувачам залишати власні відгуки, оцінювати місця для риболовлі та формувати загальний рейтинг локацій. Використання подібної системи дозволить значно спростити процес пошуку якісних місць для риболовлі, покращити інформованість користувачів та підвищити зручність взаємодії між рибалками.

Актуальність теми дослідження визначається зростанням попиту на інформаційні системи для автоматизованого збору та обробки користувацьких відгуків, необхідністю створення єдиного каталогу риболовних місць та потребою у швидкому доступі до достовірної інформації про локації. Розробка системи збору відгуків та формування рейтингу риболовних локацій дозволить користувачам отримувати актуальні дані про місця для риболовлі, переглядати оцінки інших відвідувачів, а також ділитися власним досвідом.

Метою бакалаврської роботи є розробка програмного забезпечення для збору відгуків та формування рейтингу риболовних локацій. Створена система повинна забезпечувати реєстрацію та авторизацію користувачів, перегляд доступних локацій, отримання детальної інформації про місця для риболовлі,

додавання текстових відгуків та оцінювання локацій за допомогою рейтингової системи.

Об'єктом дослідження є процес збору, зберігання та обробки інформації про риболовні локації у програмних інформаційних системах.

Предметом дослідження є методи, моделі та програмні засоби розробки системи збору відгуків і формування рейтингу риболовних локацій із використанням технологій C#, Windows Forms та SQLite.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. провести аналіз існуючих інформаційних систем та сервісів для пошуку і оцінювання риболовних локацій;
2. визначити функціональні та нефункціональні вимоги до програмного забезпечення;
3. спроектувати структуру системи та бази даних;
4. реалізувати механізми реєстрації та авторизації користувачів;
5. розробити функціонал перегляду каталогу риболовних локацій та детальної інформації про них;
6. реалізувати систему додавання відгуків і виставлення рейтингу локаціям;
7. забезпечити можливість переходу до географічного розташування локації за допомогою картографічного сервісу;
8. провести тестування програмного забезпечення та оцінити коректність його роботи.

Перевірка працездатності системи здійснювалася шляхом функціонального тестування основних модулів програмного застосунку, зокрема авторизації користувачів, роботи з базою даних, відображення інформації про локації та додавання відгуків.

Апробація результатів роботи здійснювалася шляхом подання тез доповіді на тему «Розробка системи збору відгуків та формування рейтингу риболовних локацій» на VII Всеукраїнську науково-практичну конференцію студентів та

аспірантів «Теоретичні та прикладні аспекти розробки комп'ютерних систем '2026», що відбулася 29 квітня 2026 року на факультеті інформаційних технологій НУБіП України, м. Київ. У межах апробації було представлено ідею створення програмної системи, обґрунтовано її актуальність, визначено основні функціональні можливості майбутнього застосунку, зокрема каталог риболовних локацій, механізм збору відгуків, формування рейтингу та можливість переходу до місцезнаходження водойми через Google Maps.

Структура бакалаврської кваліфікаційної роботи складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг пояснювальної записки становить 79 сторінок. Робота містить 22 рисунки, 7 таблиць та 3 додатки. Список використаних джерел налічує 24 найменувань.

У першому розділі виконано аналіз предметної області, розглянуто особливості систем збору відгуків і формування рейтингу риболовних локацій, а також проведено огляд існуючих програмних рішень. У другому розділі здійснено моделювання предметної області засобами UML, зокрема описано основні сценарії взаємодії користувача із системою. У третьому розділі наведено проєктування та реалізацію програмного забезпечення, обґрунтовано вибір інструментальних засобів розробки, описано структуру бази даних та основні програмні модулі. У четвертому розділі визначено вимоги до впровадження й експлуатації системи, наведено діаграму розгортання та результати тестування програмного застосунку.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис предметної області

У сучасних умовах інформаційні технології активно впроваджуються у сферу відпочинку та туризму, зокрема у напрямок організації риболовлі. Значна кількість рибалок використовує інтернет-ресурси для пошуку нових місць для риболовлі, отримання інформації про водойми, умови перебування та відгуки інших користувачів. Проте більшість наявних ресурсів не забезпечують централізованого та зручного доступу до структурованої інформації про риболовні локації, що створює певні труднощі для користувачів.

Однією з основних проблем у даній сфері є відсутність єдиної системи, яка б дозволяла користувачам швидко знаходити перевірені місця для риболовлі, переглядати оцінки та відгуки інших рибалок, а також отримувати інформацію про особливості конкретної локації. Найчастіше інформація про риболовні місця розміщується у соціальних мережах, форумах або окремих вебсайтах, що ускладнює процес пошуку та перевірки актуальності даних.

Ще однією проблемою є недостатня інформативність більшості наявних ресурсів. Часто користувачі не мають можливості переглядати детальний опис локації, оцінювати її за певними критеріями або залишати власні коментарі щодо умов риболовлі. Відсутність рейтингової системи також ускладнює визначення найбільш популярних та якісних місць для відпочинку.

Важливим аспектом є й географічне розташування риболовних локацій. Для багатьох користувачів важливо не лише переглянути опис водойми, а й швидко отримати доступ до її місцезнаходження на карті. Без інтеграції картографічних сервісів пошук необхідної локації може займати значний час та створювати додаткові незручності.

Сучасні програмні технології дозволяють ефективно вирішувати подібні проблеми шляхом створення спеціалізованих інформаційних систем.

Використання програмного забезпечення для збору відгуків та формування рейтингу риболовних локацій дає можливість автоматизувати процес накопичення та обробки інформації про водойми, забезпечити зручний доступ до даних та покращити взаємодію між користувачами системи.

Розроблювана система дозволяє користувачам проходити реєстрацію та авторизацію, переглядати каталог доступних риболовних локацій, отримувати детальну інформацію про обрані місця, залишати текстові відгуки та оцінки за п'ятибальною шкалою. На основі оцінок користувачів формується загальний рейтинг локацій, що допомагає визначати найбільш популярні місця для риболовлі.

Предметною областю даної бакалаврської роботи є процес збору, зберігання та аналізу інформації про риболовні локації із використанням сучасних інформаційних технологій. Основна увага приділяється створенню програмної системи, яка забезпечує зручну взаємодію користувачів із каталогом риболовних місць та реалізує механізми оцінювання й коментування локацій.

Система передбачає використання клієнтського інтерфейсу, розробленого за допомогою технології Windows Forms, а також бази даних SQLite для зберігання інформації про користувачів, риболовні локації та відгуки. Для реалізації переходу до місцезнаходження водойм використовується інтеграція з картографічним сервісом Google Maps.

До основних функцій системи належать:

- реєстрація та авторизація користувачів;
- перегляд списку риболовних локацій;
- відображення детальної інформації про локацію;
- додавання текстових відгуків;
- оцінювання локацій за допомогою рейтингової системи;
- формування середнього рейтингу локацій;
- перехід до географічного розташування місця через Google Maps.

Аналіз предметної області показує, що впровадження подібної системи дозволить значно спростити процес пошуку риболовних місць, підвищити зручність отримання інформації та забезпечити користувачів актуальними відомостями про водойми на основі реального досвіду інших рибалок. Це підтверджує актуальність розробки системи збору відгуків та формування рейтингу риболовних локацій.

## **1.2 Огляд існуючих рішень**

З розвитком цифрових технологій та популяризацією активного відпочинку значно зросла кількість онлайн-сервісів, призначених для пошуку місць риболовлі, обміну досвідом між користувачами та оцінювання риболовних локацій. Більшість сучасних платформ надають можливість перегляду інформації про водойми, публікації фотографій, залишення коментарів та використання картографічних сервісів. Проте далеко не всі системи поєднують необхідний функціонал у межах одного зручного програмного рішення.

Одним із найбільш популярних сервісів для пошуку місць риболовлі є платформа Fishbrain, яка поєднує функції пошуку водойм, соціальної взаємодії між рибалками та перегляду інформації про риболовні локації [10].



Рис. 1.1 Платформа “Fishbrain”

Основною перевагою Fishbrain є велика база риболовних місць та активна спільнота користувачів. Сервіс дозволяє переглядати популярні водойми, оцінювати ефективність риболовлі, а також аналізувати статистику вилову. Крім того, система використовує геолокацію та інтеграцію з картами для відображення точного місцезнаходження риболовних локацій.

Водночас Fishbrain має низку обмежень. Частина функціональних можливостей доступна лише за платною підпискою, а деякі дані про водойми можуть бути неповними або недоступними для окремих регіонів. Також система орієнтована переважно на мобільні пристрої та не дозволяє гнучко адаптувати функціонал під потреби конкретного користувача.

Іншим популярним рішенням є сервіс FishAngler, який поєднує функції карти риболовних місць, пошуку водойм поблизу, перегляду інформації про

умови риболовлі та соціальної взаємодії користувачів [11].



Рис. 1.2 Платформа “FishAngler”

Система дозволяє користувачам знаходити водойми поблизу, переглядати відгуки інших рибалок, додавати власні оцінки та обмінюватися інформацією щодо умов риболовлі. Перевагою FishAngler є інтеграція з GPS-навігацією та відображення погодних умов, що є важливим фактором для планування риболовлі.

Проте даний сервіс має і певні недоліки. Інтерфейс системи є перевантаженим великою кількістю функцій, що ускладнює користування для нових користувачів. Крім того, значна частина функціоналу орієнтована на англomовну аудиторію, а інформація про локальні водойми в окремих країнах може бути обмеженою.

Ще одним прикладом інформаційної системи для рибалок є Navionics Boating, основним призначенням якої є робота з навігаційними картами, водоймами та маршрутами [12].



Рис. 1.3 Платформа “Navionics Boating”

Система надає детальні карти водойм, інформацію про глибину, рельєф дна та навігаційні дані. Користувачі можуть зберігати власні маршрути, додавати позначки та переглядати популярні місця для риболовлі. Платформа активно використовується як любителями, так і професійними рибалками.

Недоліком Navionics є те, що сервіс більше орієнтований на навігацію та картографію, ніж на систему відгуків та рейтингу локацій. Крім того, частина картографічних матеріалів є платною, а сам функціонал може бути надмірно складним для звичайних користувачів.

Серед українських ресурсів також існують тематичні форуми та сайти для рибалок, де користувачі можуть обмінюватися інформацією про водойми та залишати власні враження. Однак більшість таких ресурсів не мають сучасного програмного інтерфейсу, централізованої системи рейтингу або зручної інтеграції з картографічними сервісами.

Проведений аналіз існуючих рішень показує, що сучасні сервіси для рибалок мають широкий набір можливостей, проте більшість із них характеризуються складністю використання, обмеженим функціоналом безкоштовної версії або недостатньою адаптацією до локальних потреб користувачів. Частина платформ орієнтована переважно на мобільні застосунки, тоді як інші не забезпечують повноцінної системи оцінювання та збору відгуків.

Таким чином, розробка власної системи збору відгуків та формування рейтингу риболовних локацій є актуальним завданням. Створюване програмне забезпечення повинно забезпечувати простий та зрозумілий інтерфейс, можливість перегляду інформації про водойми, систему оцінювання локацій.

Для узагальнення результатів огляду існуючих рішень було сформовано порівняльну характеристику, наведену в табл. 1.1.

Таблиця 1.1 Порівняльна характеристика існуючих рішень

| Система           | Переваги                                                                                                                                              | Недоліки                                                                                                                                                    | Що враховано у власній системі                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fishbrain         | Велика база риболовних локацій; активна спільнота користувачів; можливість перегляду інформації про водойми та улов; підтримка картографічних функцій | Частина функцій доступна лише за підпискою; не вся інформація актуальна для локальних водойм; система орієнтована переважно на мобільні пристрої            | У власній системі враховано необхідність створення каталогу риболовних локацій, можливість перегляду інформації про водойми та формування рейтингу на основі оцінок користувачів |
| FishAngler        | Підтримка пошуку водойм поблизу; наявність відгуків користувачів; інтеграція з GPS; додаткова інформація про погодні умови                            | Інтерфейс може бути перевантаженим; орієнтація переважно на англomовну аудиторію; обмежена кількість локальних даних для окремих регіонів                   | У власній системі зроблено акцент на простому інтерфейсі, зрозумілій структурі каталогу та можливості швидкого перегляду відгуків і рейтингу                                     |
| Navionics Boating | Детальні карти водойм; навігаційні можливості; інформація про глибину та рельєф дна; корисність для професійного планування маршруту                  | Основний акцент зроблено на навігації, а не на відгуках; частина картографічних матеріалів є платною; система може бути складною для звичайного користувача | У власній системі враховано необхідність інтеграції з картографічним сервісом, але без ускладнення інтерфейсу; реалізовано простий перехід до локації через Google Maps          |

На основі проведеного порівняння можна зробити висновок, що існуючі рішення мають корисні функціональні можливості, однак не завжди забезпечують простоту використання, локальну адаптацію та зручну систему оцінювання риболовних локацій. Тому під час розробки власної системи було враховано необхідність створення зрозумілого настільного застосунку з каталогом локацій, можливістю додавання відгуків, формуванням середнього рейтингу та переходом до місцезнаходження водойми через Google Maps.

### **1.3 Постановка завдання**

Організація системи збору відгуків та формування рейтингу риболовних локацій ґрунтується на необхідності забезпечення зручного доступу користувачів до актуальної інформації про місця для риболовлі. Сучасні користувачі потребують швидкого пошуку водойм, можливості переглядати оцінки інших рибалок, отримувати детальний опис локацій та ділитися власним досвідом. Водночас існуючі інформаційні ресурси часто не забезпечують централізованого зберігання даних, структурованої системи оцінювання та зручного механізму взаємодії між користувачами.

Основним завданням програмної системи є автоматизація процесу збору, обробки та відображення інформації про риболовні локації, а також формування рейтингу на основі оцінок користувачів. Система повинна забезпечувати зберігання інформації про водойми, підтримувати механізм додавання відгуків та реалізовувати зручний інтерфейс для перегляду даних.

До основних параметрів, які підлягають обробці та зберіганню в системі, належать:

- інформація про риболовні локації, включаючи назву, опис та координати місцезнаходження;
- облікові записи користувачів, необхідні для авторизації та взаємодії із системою;

- текстові відгуки користувачів щодо конкретних локацій;
- рейтингові оцінки, які використовуються для формування загального рейтингу водойм;
- середній рейтинг кожної локації, що дозволяє визначати популярність місць для риболовлі;
- географічні координати водойм для подальшої інтеграції з картографічними сервісами.

Програмний застосунок повинен використовувати централізовану базу даних для зберігання інформації про користувачів, риболовні локації та залишені відгуки. Особливу увагу необхідно приділити цілісності та узгодженості даних, оскільки система має забезпечувати коректне збереження оцінок, уникнення дублювання інформації та стабільну роботу з базою даних.

Процес взаємодії користувача із системою організований за визначеним сценарієм. Після проходження реєстрації та авторизації користувач отримує доступ до каталогу риболовних локацій. Користувач може переглядати доступні місця для риболовлі, відкривати детальну інформацію про вибрану локацію, залишати текстові коментарі та виставляти оцінки за п'ятибальною шкалою. На основі всіх отриманих оцінок система автоматично формує середній рейтинг локації.

Важливим компонентом системи є інтеграція з картографічним сервісом Google Maps. Дана функція дозволяє користувачам швидко переходити до географічного розташування риболовної локації, що значно спрощує процес пошуку водойми та планування маршруту.

Інтерфейс користувача повинен бути простим, зрозумілим та забезпечувати інтуїтивну навігацію між основними функціональними модулями системи. Окремі форми передбачені для авторизації користувачів, перегляду каталогу локацій, відображення детальної інформації про водойми та додавання відгуків.

З архітектурної точки зору система реалізується за багаторівневим принципом із розподілом на модулі інтерфейсу користувача, бізнес-логіки та

роботи з базою даних. Такий підхід забезпечує зручність підтримки програмного забезпечення, спрощує подальше розширення функціоналу та підвищує надійність роботи системи.

Таким чином, розробка системи збору відгуків та формування рейтингу риболовних локацій дозволить покращити процес пошуку місць для риболовлі, забезпечити користувачів актуальною інформацією та створити єдину платформу для обміну досвідом між рибалками.

## **1.4 Функціональні та нефункціональні вимоги**

Функціональні вимоги визначають основні можливості програмної системи збору відгуків та формування рейтингу риболовних локацій. Вони описують функції, які повинна виконувати система для забезпечення коректної взаємодії користувачів із програмним застосунком та реалізації основного функціоналу.

До функціональних вимог системи належать:

- реєстрація нових користувачів у системі;
- авторизація користувачів із використанням логіна та пароля;
- перегляд списку доступних риболовних локацій;
- відображення детальної інформації про вибрану локацію;
- збереження інформації про риболовні місця у базі даних;
- можливість додавання текстових відгуків користувачами;
- можливість оцінювання локацій за шкалою від 1 до 5;
- автоматичне обчислення середнього рейтингу риболовної локації;
- перегляд списку відгуків інших користувачів;
- перехід до місцезнаходження локації через сервіс Google Maps;
- взаємодія з базою даних для збереження та отримання інформації;
- відображення повідомлень про успішне виконання операцій або виникнення помилок;
- підтримка навігації між основними формами програмного застосунку.

Нефункціональні вимоги визначають якісні характеристики системи, які впливають на її продуктивність, стабільність та зручність використання.

До нефункціональних вимог системи належать:

- забезпечення швидкої обробки запитів користувача;
- стабільна робота програмного застосунку без критичних помилок;
- зручний та інтуїтивно зрозумілий інтерфейс користувача;

- забезпечення цілісності та коректності даних у базі даних;
- можливість подальшого розширення функціональних можливостей системи;
- простота використання для користувачів без спеціальної технічної підготовки;
- підтримка роботи із локальною базою даних SQLite;
- коректне функціонування системи на персональних комп'ютерах під керуванням операційної системи Windows;
- захист облікових записів користувачів за допомогою авторизації;
- забезпечення надійного збереження інформації про локації та відгуки;
- підтримка інтеграції із зовнішніми картографічними сервісами.

## **1.5 Вимоги до інтерфейсу користувача**

Інтерфейс користувача системи повинен забезпечувати просту, зрозумілу та зручну взаємодію з програмним застосунком. Основна увага приділяється легкості навігації між функціональними модулями, швидкому доступу до інформації про риболовні локації та комфортному використанню системи користувачами.

Головне вікно авторизації повинно містити поля для введення логіна та пароля, а також кнопки для входу та реєстрації користувача. Після успішної авторизації користувач отримує доступ до основного меню системи.

Інтерфейс каталогу риболовних локацій повинен забезпечувати зручний перегляд списку доступних місць для риболовлі. Для кожної локації повинна відображатися її назва та основна інформація. Користувач повинен мати можливість швидко перейти до детального перегляду вибраної водойми.

Форма детальної інформації про локацію повинна містити:

- назву риболовної локації;
- опис місця;
- середній рейтинг;
- список відгуків користувачів;
- поле для введення нового коментаря;
- елемент для вибору рейтингової оцінки;
- кнопку додавання відгуку;

- кнопку переходу до Google Maps.

Усі елементи інтерфейсу повинні бути логічно структурованими та зручно розташованими. Навігація між формами має бути простою та зрозумілою навіть для користувачів без технічного досвіду.

Інтерфейс програмного застосунку повинен відповідати сучасним вимогам до дизайну настільних програм, мати приємне візуальне оформлення та забезпечувати комфортне використання системи протягом тривалого часу. Особлива увага приділяється читабельності тексту, розміру елементів керування та зрозумілості повідомлень системи.

Також інтерфейс повинен коректно відображати повідомлення про помилки, результати виконання операцій та інші системні сповіщення, що забезпечує зручність взаємодії користувача із програмою та підвищує загальну ефективність роботи системи.

## 2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

### 2.1 Загальні відомості

Для проєктування системи збору відгуків та формування рейтингу риболовних локацій було використано засоби UML-моделювання, які застосовуються для опису структури, поведінки та взаємодії компонентів програмних систем [13–15].

У межах даної бакалаврської роботи моделювання виконувалося з метою формалізації основних функцій настільного застосунку, а також для відображення зв'язку між користувацьким інтерфейсом, сервісною логікою та локальною базою даних SQLite. Оскільки система реалізована з використанням C#, Windows Forms та SQLite, під час побудови моделей враховувалася фактична структура програмного продукту.

Основними об'єктами предметної області є користувач, риболовна локація та відгук. Користувач взаємодіє із системою через графічний інтерфейс, проходить реєстрацію або авторизацію, переглядає каталог локацій, відкриває детальну інформацію про водойму, залишає коментар і виставляє оцінку. Локація містить назву, опис та дані для переходу до місцезнаходження через Google Maps. Відгук містить текст коментаря, оцінку та дату створення. На основі оцінок система формує середній рейтинг локації.

Для опису системи було використано такі UML-діаграми:

- діаграму прецедентів;
- діаграму послідовності;
- діаграму активності;
- діаграму класів;
- діаграму пакетів;
- діаграму розгортання.

Діаграма прецедентів використовується для визначення основних сценаріїв взаємодії користувача із системою. Вона показує, які дії може

виконувати користувач: реєструватися, входити до системи, переглядати локації, додавати відгуки, оцінювати водойми та відкривати їх у Google Maps.

Діаграма послідовності демонструє порядок взаємодії між основними компонентами програми під час виконання ключових операцій. Зокрема, вона відображає процес вибору локації, отримання даних із бази SQLite, додавання відгуку та оновлення рейтингу.

Діаграма активності використовується для опису логіки роботи користувача із системою. Вона дозволяє показати послідовність дій від запуску програми та авторизації до перегляду локації, додавання коментаря й отримання оновленого рейтингу.

Діаграма класів відображає статичну структуру системи та основні класи, які відповідають фактичній реалізації програми: User, Location та Review. Вона показує, які дані зберігаються в системі та як ці сутності пов'язані між собою.

Діаграма пакетів використовується для відображення організаційної структури програмного забезпечення. У системі виділено пакети інтерфейсу користувача, моделей, сервісів та роботи з базою даних. Такий поділ відповідає структурі проєкту у Visual Studio та спрощує супровід програмного коду.

Діаграма розгортання демонструє фізичну структуру роботи системи. Вона показує, що настільний застосунок працює на персональному комп'ютері користувача, взаємодіє з локальною базою даних SQLite та відкриває зовнішній сервіс Google Maps через веббраузер.

Таким чином, UML-моделювання у даній роботі використовується не лише як теоретичний засіб опису системи, а як практичний інструмент для проєктування структури, функціональності та логіки роботи настільного застосунку. Побудовані моделі узгоджені з фактично реалізованим програмним забезпеченням і відображають основні функції системи збору відгуків та формування рейтингу риболовних локацій.

## 2.2 Об'єктне та функціональне моделювання

### 2.2.1 Діаграма прецедентів.

Діаграма прецедентів використовується для відображення основних функціональних можливостей системи та взаємодії користувача з програмним забезпеченням. У межах даної бакалаврської роботи вона застосовується для опису сценаріїв роботи користувача з настільним застосунком збору відгуків та формування рейтингу риболовних локацій.

Для розробленої системи основним актором є **«Користувач»**, який взаємодіє із програмою через графічний інтерфейс Windows Forms. Користувач може створити обліковий запис, виконати вхід до системи, переглядати каталог риболовних локацій, відкривати детальну інформацію про вибрану водойму, залишати відгуки, оцінювати локації, переглядати середній рейтинг та відкривати місцезнаходження водойми у Google Maps.

До основних прецедентів системи належать:

- реєстрація;
- вхід у систему;
- перегляд каталогу локацій;
- перегляд інформації про локацію;
- додавання відгуку;
- оцінювання локації;
- перегляд середнього рейтингу;
- відкриття локації на карті.

Після запуску програми користувач потрапляє на форму авторизації, де може виконати вхід до системи або зареєструвати новий обліковий запис. Після успішної авторизації відкривається головна форма застосунку, у якій відображається каталог доступних риболовних локацій.

Після вибору конкретної локації користувач може відкрити форму детального перегляду. На цій формі відображається назва локації, її опис, список відгуків та середній рейтинг. Також користувач має можливість ввести

власний коментар і виставити оцінку за шкалою від 1 до 5. Після додавання відгуку система зберігає інформацію у локальній базі даних SQLite та оновлює середній рейтинг локації.

Окремим прецедентом є відкриття локації на карті. Для цього користувач натискає відповідну кнопку, після чого система формує посилання та відкриває місцезнаходження риболовної локації у веббраузері за допомогою сервісу Google Maps.

Розроблена діаграма прецедентів використання представлена на рис. 2.1.

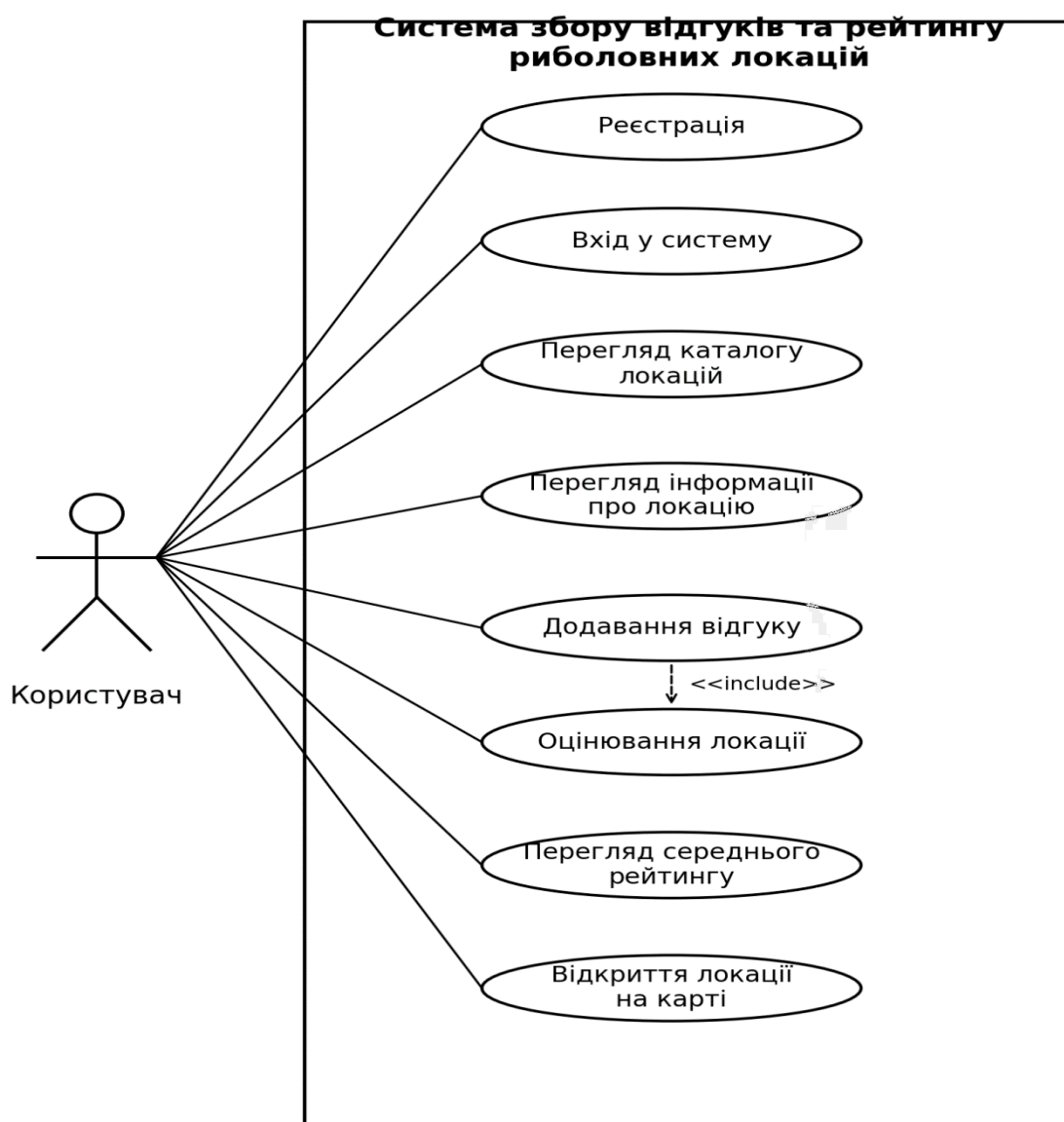


Рис. 2.1 Діаграма прецедентів

Створена діаграма прецедентів відображає основні сценарії взаємодії користувача з настільним застосунком. Вона узгоджена з фактично реалізованим функціоналом програмної системи та охоплює авторизацію, роботу з каталогом локацій, перегляд інформації, додавання відгуків, оцінювання, перегляд рейтингу та перехід до локації через Google Maps.

### **Варіант використання: Перегляд інформації про локацію**

**Актор:** Користувач.

#### **Опис:**

Варіант використання «Перегляд інформації про локацію» описує процес отримання користувачем детальної інформації про вибране риболовне місце. Користувач після входу до системи переглядає каталог доступних локацій, обирає потрібну водойму та відкриває форму з її описом, відгуками та середнім рейтингом. Даний сценарій дозволяє користувачу швидко ознайомитися з інформацією про риболовне місце та оцінити його на основі даних, збережених у системі.

#### **Основний потік:**

1. Користувач запускає програму.
2. Система відкриває форму авторизації.
3. Користувач виконує вхід у систему.
4. Система відкриває головне вікно з каталогом риболовних локацій.
5. Користувач обирає потрібну локацію зі списку.
6. Користувач натискає кнопку відкриття локації.
7. Система відкриває форму детальної інформації.
8. Система відображає назву локації, опис, список відгуків і середній рейтинг.
9. Користувач переглядає інформацію про вибране місце.

#### **Альтернативні потоки:**

1. Якщо користувач не обрав локацію зі списку, система виводить повідомлення про необхідність вибору локації.

2. Якщо список локацій порожній, система не відображає доступних записів.
3. Якщо виникає помилка під час роботи з базою даних, система повідомляє користувача про неможливість отримання даних.
4. Користувач може закрити форму детального перегляду та повернутися до каталогу локацій.

**Результат:**

Користувач отримує детальну інформацію про вибрану риболовну локацію, включаючи опис, список відгуків та середній рейтинг.

**Варіант використання: Додавання відгуку та оцінювання локації**

**Актор:** Користувач.

**Опис:**

Варіант використання «Додавання відгуку та оцінювання локації» описує процес створення користувачем текстового коментаря та виставлення рейтингової оцінки для вибраної риболовної локації. Після перегляду інформації про водойму користувач може залишити власний відгук, обрати оцінку від 1 до 5 та зберегти ці дані у системі. Після додавання нового відгуку система оновлює список коментарів і автоматично перераховує середній рейтинг локації.

**Основний потік:**

1. Користувач відкриває форму детальної інформації про локацію.
2. Система відображає поточну інформацію про водойму, список відгуків та середній рейтинг.
3. Користувач вводить текст коментаря у відповідне поле.
4. Користувач обирає рейтингову оцінку від 1 до 5.
5. Користувач натискає кнопку «Додати відгук».
6. Система зберігає відгук у локальній базі даних SQLite.
7. Система оновлює список відгуків на формі.

8. Система перераховує середній рейтинг локації.
9. Користувач бачить оновлену інформацію на екрані.

**Альтернативні потоки:**

1. Якщо поле коментаря порожнє, система може повідомити користувача про необхідність введення тексту.
2. Якщо під час збереження виникає помилка бази даних, система виводить повідомлення про помилку.
3. Якщо користувач вирішує не додавати відгук, він може закрити форму або залишити поля без змін.
4. Якщо оцінка не була змінена користувачем, система використовує значення, встановлене за замовчуванням.

**Результат:**

Новий відгук зберігається у базі даних, відображається у списку коментарів, а середній рейтинг локації оновлюється відповідно до доданої оцінки.

**Варіант використання: Відкриття локації на карті**

**Актор:** Користувач.

**Опис:**

Варіант використання «Відкриття локації на карті» описує процес переходу користувача до місцезнаходження вибраної риболовної локації через сервіс Google Maps. Дана функція дозволяє користувачу швидко переглянути розташування водойми у веббраузері та за потреби спланувати маршрут до неї.

**Основний потік:**

1. Користувач відкриває форму детальної інформації про локацію.
2. Система відображає інформацію про вибрану риболовну локацію.
3. Користувач натискає кнопку «Відкрити на карті».
4. Система формує посилання для відкриття локації у Google Maps.
5. Система відкриває посилання у веббраузері користувача.
6. Google Maps відображає вибрану риболовну локацію на карті.
7. Користувач переглядає місцезнаходження водойми.

**Альтернативні потоки:**

1. Якщо на комп'ютері відсутнє підключення до Інтернету, сторінка Google Maps може не завантажитися.
2. Якщо веббраузер недоступний або не налаштований, система не зможе відкрити посилання.
3. Якщо Google Maps не знаходить локацію за назвою, користувач може вручну уточнити пошуковий запит у браузері.
4. Користувач може повернутися до настільного застосунку після перегляду карти.

**Результат:**

Користувач отримує можливість переглянути місцезнаходження вибраної риболовної локації у Google Maps.

### ***2.2.2 Діаграма послідовності.***

Діаграма послідовності використовується для відображення порядку взаємодії між основними компонентами системи під час виконання певного сценарію. Вона дозволяє показати, які дії виконує користувач, які форми програмного застосунку беруть участь в обробці запиту, а також у якій послідовності відбувається звернення до бази даних і зовнішніх сервісів.

Для системи збору відгуків та формування рейтингу риболовних локацій діаграма послідовності описує один із ключових сценаріїв роботи: авторизацію користувача, перегляд каталогу риболовних локацій, відкриття детальної інформації про вибрану локацію, додавання відгуку та перехід до Google Maps. Це дозволяє наочно відобразити логіку функціонування настільного застосунку та взаємозв'язок між його складовими.

На діаграмі послідовності виділено такі основні учасники взаємодії:

- **Користувач;**
- **Form1 (Авторизація);**

- **MainForm (Каталог);**
- **LocationDetailsForm;**
- **База даних SQLite;**
- **Google Maps.**

Після запуску програми користувач у формі авторизації вводить електронну пошту та пароль. Форма Form1 передає запит на перевірку облікових даних до локальної бази даних SQLite та отримує результат авторизації. Якщо введені дані є правильними, система відкриває головну форму MainForm, яка містить каталог риболовних локацій.

На наступному етапі користувач обирає потрібну локацію зі списку. Головна форма звертається до бази даних для отримання відомостей про вибране місце риболовлі, після чого відкривається форма LocationDetailsForm. У цій формі завантажуються відгуки користувачів і поточний рейтинг локації, що дає змогу ознайомитися з детальною інформацією перед залишенням власного коментаря.

Якщо користувач бажає поділитися власним досвідом, він вводить текст відгуку та оцінку. Після цього форма LocationDetailsForm передає дані до бази даних SQLite, де відгук зберігається, а також оновлюються пов'язані з локацією дані. У результаті користувач отримує оновлений список відгуків і актуальне значення рейтингу.

Окремим варіантом взаємодії є відкриття місцезнаходження локації на карті. Для цього користувач натискає кнопку «Відкрити на карті», після чого форма LocationDetailsForm ініціює відкриття відповідного посилання у сервісі Google Maps.

Розроблена діаграма послідовності представлена на **рис. 2.2**.

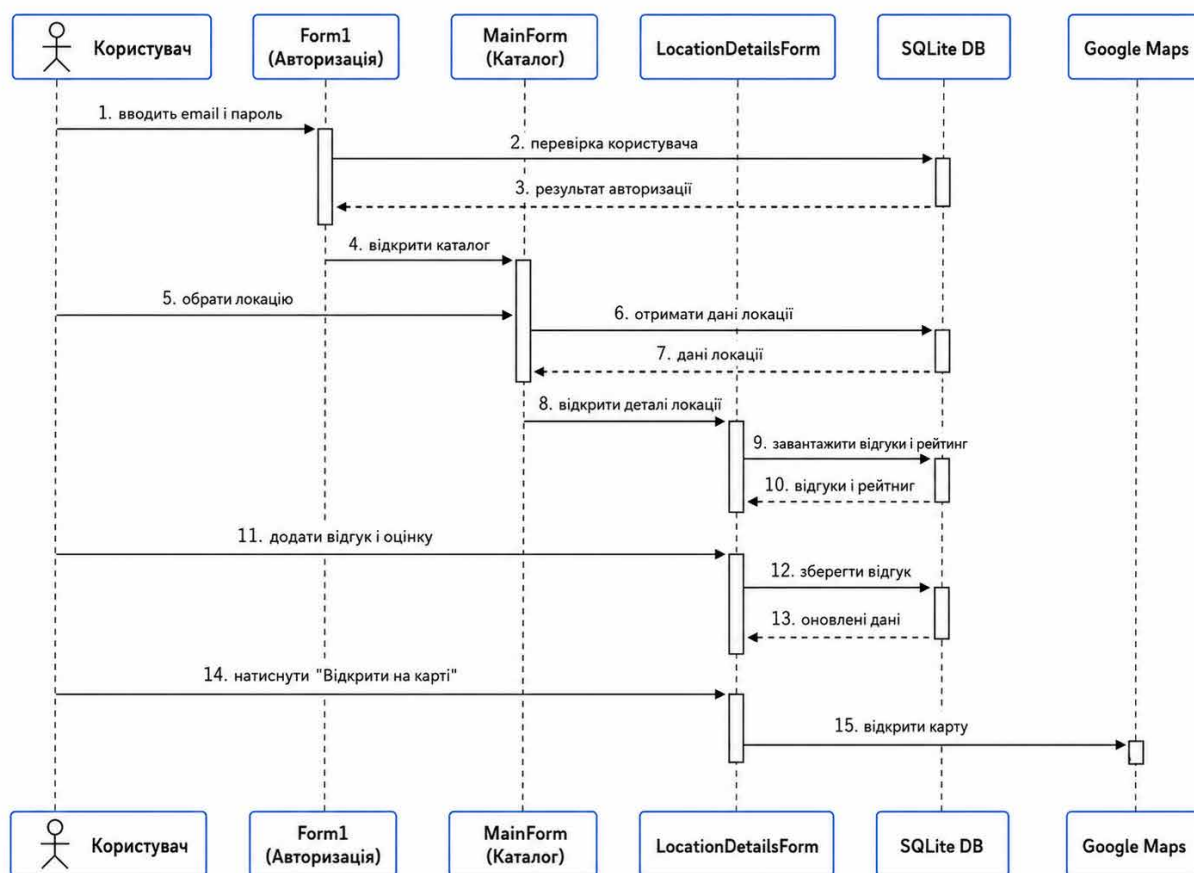


Рис.2.2 Діаграма послідовності

На рис. 2.2 зображено послідовність взаємодії користувача з основними компонентами розробленого настільного застосунку. Діаграма демонструє типовий сценарій роботи системи: від авторизації користувача до перегляду риболовної локації, додавання відгуку, оновлення рейтингу та відкриття місцезнаходження водойми у Google Maps.

Початковим етапом взаємодії є введення користувачем електронної пошти та пароля у формі Form1. Після цього система виконує перевірку введених облікових даних. У розробленому застосунку ця перевірка здійснюється локально, без використання окремого серверного компонента. Дані користувача порівнюються з інформацією, що зберігається у базі даних SQLite. Якщо авторизація проходить успішно, користувач отримує доступ до головної форми програми.

Після успішного входу відкривається форма MainForm, яка виконує роль каталогу риболовних локацій. На цьому етапі користувач має можливість переглядати список доступних водойм, виконувати пошук потрібної локації та сортувати записи за середнім рейтингом. Коли користувач обирає конкретну локацію, головна форма отримує необхідні дані про вибране місце з бази даних і передає їх до форми детального перегляду.

Форма LocationDetailsForm призначена для відображення повної інформації про вибрану риболовну локацію. На ній користувач бачить назву водойми, її опис, список відгуків та поточне значення середнього рейтингу. Для цього форма звертається до бази даних SQLite, звідки завантажуються відгуки, пов'язані з відповідною локацією, а також значення рейтингу, сформоване на основі оцінок користувачів.

Окремий блок діаграми відображає процес додавання нового відгуку. Користувач вводить текст коментаря та обирає рейтингову оцінку. Після натискання кнопки додавання відгуку система зберігає новий запис у базі даних. Після цього дані на формі оновлюються: користувач бачить доданий коментар, а рейтинг локації перераховується з урахуванням нової оцінки. Така послідовність забезпечує актуальність інформації, яка відображається у системі.

На діаграмі також показано взаємодію із зовнішнім сервісом Google Maps. Після натискання кнопки «Відкрити на карті» форма LocationDetailsForm формує запит на відкриття карти. У цьому випадку застосунок не використовує окремий сервер або складну інтеграцію з API, а відкриває відповідне посилання у веббраузері користувача. Це дозволяє швидко перейти до перегляду місцезнаходження вибраної риболовної локації.

Важливо зазначити, що обробка дій користувача у розробленій системі виконується всередині настільного застосунку. Функції авторизації,

отримання даних про локації, додавання відгуків і формування рейтингу реалізовані у програмній логіці застосунку. База даних SQLite використовується як локальне сховище інформації, у якому зберігаються користувачі, риболовні локації, відгуки та оцінки.

Діаграма послідовності підтверджує, що система має зрозумілу логіку роботи та не потребує окремої серверної частини. Усі основні операції виконуються локально: користувач взаємодіє з формами Windows Forms, програма звертається до SQLite для читання або запису даних, а Google Maps використовується лише як зовнішній сервіс для перегляду місцезнаходження локації.

Такий підхід відповідає фактичній реалізації програмного забезпечення та демонструє послідовність виконання основних функцій системи збору відгуків і формування рейтингу риболовних локацій. Діаграма також показує, що після кожної важливої дії користувача система повертає результат: відкриває каталог, завантажує дані локації, відображає відгуки, оновлює рейтинг або відкриває карту. Це робить роботу застосунку передбачуваною та зручною для користувача.

### ***2.2.3 Діаграма активності.***

Діаграма активності використовується для відображення логіки виконання процесів у системі та послідовності дій користувача під час роботи з програмним застосунком. У межах даної роботи вона демонструє основний сценарій використання системи збору відгуків та формування рейтингу риболовних локацій.

На відміну від діаграми послідовності, яка показує порядок взаємодії між окремими компонентами системи, діаграма активності відображає загальний алгоритм роботи користувача із застосунком. Вона дозволяє простежити шлях користувача від запуску програми та авторизації до

перегляду риболовної локації, додавання відгуку, оновлення рейтингу або відкриття місцезнаходження водойми через Google Maps.

Основний процес починається із запуску настільного застосунку. Після цього користувач вводить електронну пошту та пароль. Система перевіряє коректність введених даних. Якщо дані неправильні, користувачу відображається повідомлення про помилку, після чого він може повторно ввести облікові дані. Якщо авторизація проходить успішно, відкривається головне вікно програми з каталогом риболовних локацій.

Після відкриття каталогу користувач обирає потрібну локацію зі списку. Якщо локацію не вибрано, система повідомляє про необхідність вибору. Якщо локація обрана коректно, відкривається форма детальної інформації про неї. На цій формі відображається опис риболовного місця, список відгуків користувачів та середній рейтинг.

Далі користувач може виконати одну з основних дій: додати відгук або відкрити локацію на карті. У разі додавання відгуку користувач вводить текст коментаря та обирає оцінку. Після цього система зберігає відгук у локальній базі даних SQLite, оновлює список відгуків і перераховує середній рейтинг локації.

Якщо користувач обирає відкриття карти, система формує посилання на Google Maps та відкриває місцезнаходження риболовної локації у веббраузері. Після виконання обраної дії користувач може завершити роботу з локацією або повернутися до подальшого використання програми.

Розроблена діаграма активності представлена на рис. 2.3.

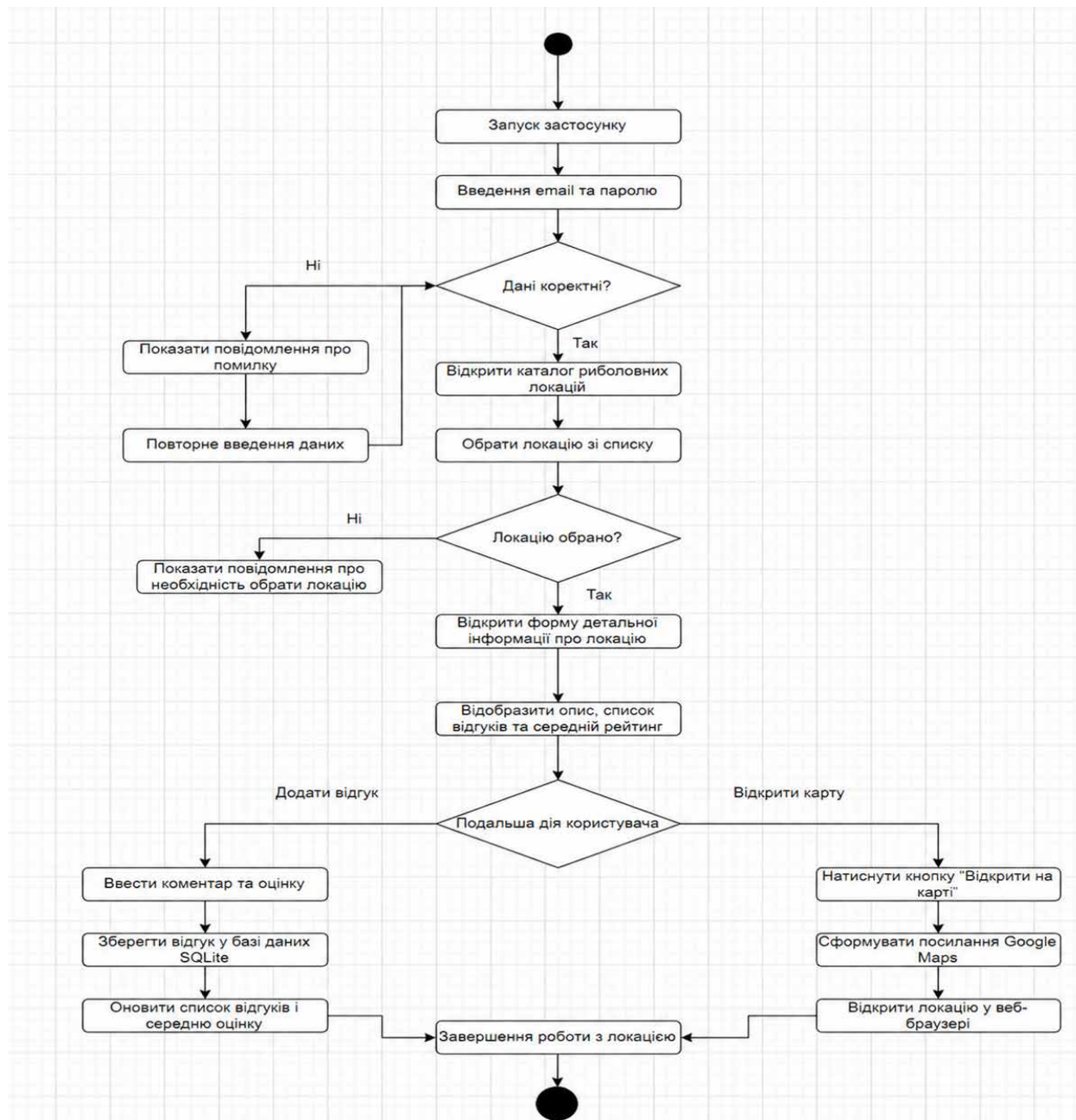


Рис. 2.3 Діаграма активності

Діаграма активності демонструє повний шлях користувача під час роботи з розробленим настільним застосунком. Вона відображає не лише основний сценарій використання системи, а й можливі альтернативні варіанти розвитку подій, зокрема помилкове введення облікових даних або відсутність вибраної локації у списку. Завдяки цьому можна краще простежити, як система реагує на дії користувача на кожному етапі.

Початковим етапом роботи є запуск застосунку та введення користувачем електронної пошти й пароля. Після цього система виконує перевірку введених даних. Якщо дані некоректні, користувачу відображається повідомлення про помилку та надається можливість повторного введення. Такий механізм забезпечує базовий контроль доступу до функціональних можливостей програми.

Після успішного входу користувач переходить до каталогу риболовних локацій. На цьому етапі система відображає список доступних місць для риболовлі, які зберігаються у локальній базі даних SQLite. Користувач має обрати одну з локацій для подальшого перегляду. Якщо локацію не вибрано, система повідомляє про необхідність виконати вибір, що запобігає некоректному переходу до наступного етапу роботи.

Після вибору локації відкривається форма детальної інформації, де користувач може переглянути назву водойми, її опис, наявні відгуки та середню оцінку. Саме цей етап є центральним у роботі системи, оскільки він забезпечує користувача основною інформацією для прийняття рішення щодо вибору місця для риболовлі.

Далі користувач може виконати одну з двох основних дій. Перший варіант передбачає додавання власного відгуку та оцінки. У цьому випадку користувач вводить текст коментаря, обирає рейтингову оцінку, після чого система зберігає дані у базі SQLite. Після збереження відгуку список коментарів оновлюється, а середній рейтинг локації перераховується з урахуванням нової оцінки.

Другий варіант передбачає відкриття вибраної риболовної локації на карті. Після натискання відповідної кнопки система формує посилання на сервіс Google Maps і відкриває його у веббраузері. Це дозволяє користувачу швидко переглянути розташування водойми та за потреби спланувати маршрут до неї.

Таким чином, діаграма активності підтверджує логічну послідовність роботи програмної системи та демонструє взаємозв'язок між основними

функціями застосунку: авторизацією, переглядом каталогу локацій, відображенням детальної інформації, додаванням відгуків, формуванням рейтингу та переходом до картографічного сервісу. Побудована модель відповідає фактично реалізованому функціоналу програми та може використовуватися як основа для подальшого вдосконалення системи.

## 2.3 Абстракції предметної області

Абстракції предметної області використовуються для виділення основних сутностей системи, їх властивостей та взаємозв'язків. Вони дозволяють описати предметну область у спрощеному вигляді, зосередившись на тих об'єктах, які є важливими для реалізації програмного забезпечення.

У межах розробки системи збору відгуків та формування рейтингу риболовних локацій було визначено основні абстракції, які відповідають фактично реалізованому функціоналу настільного застосунку. До них належать: **«Користувач»**, **«Локація»** та **«Відгук»**.

Абстракція **«Користувач»** описує особу, яка взаємодіє із системою через графічний інтерфейс Windows Forms. Користувач може пройти реєстрацію, виконати вхід до системи, переглядати каталог риболовних локацій, відкривати детальну інформацію про водойми, залишати текстові відгуки та виставляти рейтингові оцінки. Для ідентифікації користувача у системі зберігаються його ім'я, електронна пошта та пароль.

Абстракція **«Локація»** представляє риболовне місце, інформація про яке зберігається у базі даних. Для кожної локації передбачено назву, опис, широту та довготу. Ці дані використовуються для відображення інформації про водойму у програмі та для переходу до її місцезнаходження через сервіс Google Maps.

Абстракція **«Відгук»** описує коментар користувача щодо конкретної риболовної локації. Відгук містить рейтингову оцінку, текст коментаря та дату створення. На основі оцінок, що зберігаються у відгуках, система обчислює

середній рейтинг локації. Це дозволяє користувачам орієнтуватися на досвід інших рибалок під час вибору місця для риболовлі.

Між абстракціями існують логічні зв'язки. Користувач взаємодіє з локаціями через інтерфейс програми, переглядає інформацію про них та може залишати відгуки. Одна локація може мати багато відгуків, тоді як кожен відгук належить лише одній конкретній локації. Рейтинг локації формується як середнє значення всіх оцінок, залишених користувачами.

Таблиця 2.1 Абстракції предметної області

| Абстракція | Призначення                                                 | Основні атрибути                           |
|------------|-------------------------------------------------------------|--------------------------------------------|
| Користувач | Забезпечує реєстрацію, авторизацію та взаємодію із системою | Id, Username, Email, Password              |
| Локація    | Описує риболовне місце, доступне для перегляду              | Id, Name, Description, Latitude, Longitude |
| Відгук     | Зберігає коментар та оцінку користувача щодо локації        | Id, LocationId, Rating, Comment, Date      |

## 2.4 Діаграма класів

Діаграма класів є одним із основних засобів UML-моделювання, який використовується для опису статичної структури програмної системи. Вона дозволяє визначити основні класи, їх атрибути, методи та зв'язки між ними. Для системи збору відгуків та формування рейтингу риболовних локацій така діаграма є важливим етапом проєктування, оскільки вона відображає структуру даних і показує, як основні сутності взаємодіють між собою.

У межах даної роботи діаграма класів побудована відповідно до фактичної реалізації настільного застосунку, розробленого з використанням C#, Windows Forms та SQLite. На відміну від поведінкових діаграм, які демонструють послідовність дій користувача, діаграма класів описує

внутрішню організацію програмної системи та визначає, які об'єкти використовуються для збереження й обробки інформації.

Основними класами системи є **«Користувач»**, **«Риболовна локація»**, **«Відгук»** та **«Рейтинг»**. Саме ці класи відповідають ключовим функціям застосунку: реєстрації та авторизації користувача, перегляду каталогу риболовних локацій, додаванню відгуків, виставленню оцінок і відображенню середнього рейтингу.

Центральним елементом моделі є клас **«Риболовна локація»**, оскільки саме навколо нього формується основна логіка роботи системи. З ним пов'язані відгуки користувачів та рейтингові оцінки. Клас **«Користувач»** використовується для опису облікового запису особи, яка працює із системою, а клас **«Відгук»** забезпечує збереження коментарів і оцінок щодо конкретної локації. Клас **«Рейтинг»** відображає середню оцінку локації та використовується для представлення узагальненого результату оцінювання.

Між класами встановлені логічні зв'язки. Один користувач може залишати декілька відгуків, а одна риболовна локація може містити багато відгуків. Кожен відгук належить до конкретної локації та містить оцінку, яка використовується для формування рейтингу. Така структура дозволяє організувати дані у зрозумілому вигляді та забезпечує відповідність між предметною областю, логічною моделлю й програмною реалізацією.

Розроблена діаграма класів представлена на рис. 2.5.

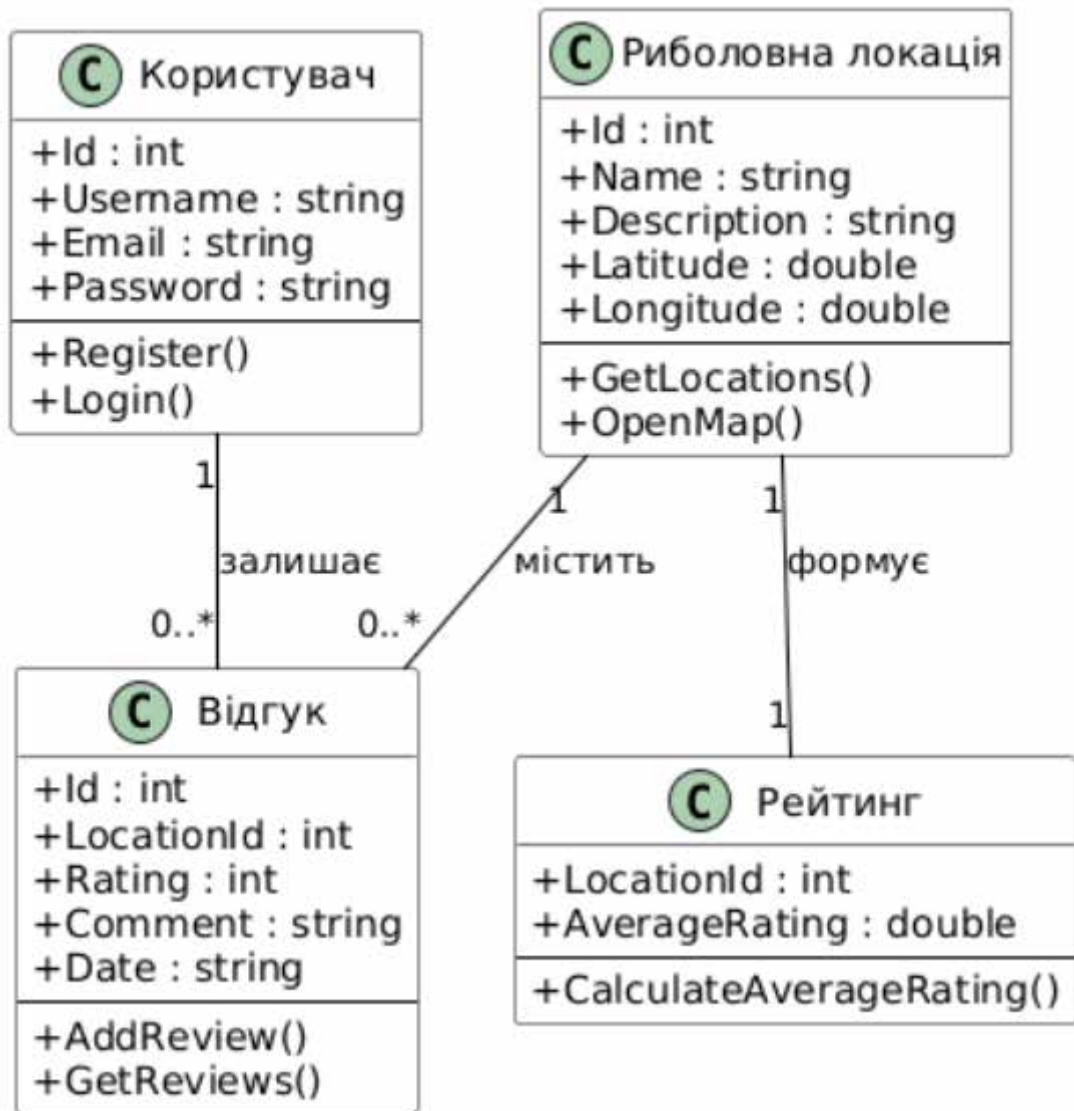


Рис. 2.5 Діаграма класів

На діаграмі класів виділено чотири основні класи, які відображають структуру розробленої системи: **«Користувач»**, **«Риболовна локація»**, **«Відгук»** та **«Рейтинг»**. Кожен із цих класів відповідає окремій частині функціональності настільного застосунку та використовується для збереження, обробки або відображення даних у системі.

Клас **«Користувач»** призначений для представлення особи, яка працює з програмним застосунком. Він містить атрибути Id, Username, Email та Password. Атрибут Id використовується як унікальний ідентифікатор користувача в системі. Поле Username зберігає ім'я користувача, Email використовується для авторизації, а Password — для перевірки права доступу до системи. У класі також передбачено методи Register() та Login(), які

відображають процеси реєстрації нового користувача та входу до системи. Завдяки цьому класу реалізується початковий етап взаємодії користувача із застосунком.

Клас **«Риболовна локація»** є центральним елементом діаграми, оскільки саме риболовні місця є основним об'єктом, з яким працює користувач. Цей клас містить атрибути Id, Name, Description, Latitude та Longitude. Атрибут Id використовується для унікальної ідентифікації локації. Поле Name зберігає назву риболовного місця, а Description містить короткий опис водойми або умов риболовлі. Атрибути Latitude і Longitude призначені для збереження географічних координат локації, які можуть використовуватися для переходу до її місцезнаходження на карті. Методи GetLocations() та OpenMap() відображають основні операції, пов'язані з цим класом: отримання списку доступних локацій і відкриття вибраного місця у Google Maps.

Клас **«Відгук»** відповідає за збереження коментарів і оцінок, які користувачі залишають для риболовних локацій. Він містить атрибути Id, LocationId, Rating, Comment та Date. Атрибут Id є унікальним ідентифікатором відгуку. Поле LocationId використовується для зв'язку відгуку з конкретною риболовною локацією. Атрибут Rating зберігає числову оцінку, яку користувач виставляє локації, а Comment містить текстовий опис вражень або зауважень користувача. Поле Date фіксує дату створення відгуку. Методи AddReview() та GetReviews() відображають операції додавання нового відгуку та отримання списку відгуків для конкретної локації.

Клас **«Рейтинг»** використовується для представлення узагальненої оцінки риболовної локації. Він містить атрибути LocationId та AverageRating. Поле LocationId пов'язує рейтинг із конкретною локацією, а AverageRating зберігає середнє значення оцінки. Метод CalculateAverageRating() відображає процес обчислення середнього рейтингу. Цей клас важливий для системи, оскільки рейтинг дозволяє користувачу швидко оцінити якість або популярність риболовного місця на основі залишених оцінок.

На діаграмі також показано зв'язки між класами. Між класами **«Користувач»** і **«Відгук»** встановлено зв'язок, який показує, що один користувач може залишати декілька відгуків. Це відповідає логіці роботи системи, оскільки користувач після авторизації може переглядати різні локації та додавати власні коментарі до них.

Між класами **«Риболовна локація»** і **«Відгук»** встановлено зв'язок типу **«один-до-багатьох»**. Це означає, що одна риболовна локація може містити багато відгуків, але кожен окремий відгук належить лише одній конкретній локації. Такий зв'язок забезпечує правильну організацію даних і дозволяє відображати на формі детальної інформації тільки ті відгуки, які стосуються вибраного місця.

Між класами **«Риболовна локація»** і **«Рейтинг»** встановлено зв'язок **«один-до-одного»**, оскільки кожна локація має один середній рейтинг. Рейтинг формується для конкретної водойми та використовується під час відображення детальної інформації про неї. Завдяки цьому користувач може не лише прочитати окремі відгуки, а й одразу побачити загальну оцінку локації.

Побудована діаграма класів показує, що структура системи є простою та логічною. Основна робота застосунку зосереджена навколо риболовних локацій, до яких користувачі можуть залишати відгуки та оцінки. Така модель відповідає фактичній реалізації програмного забезпечення, не містить зайвих сутностей і забезпечує узгодженість між предметною областю, базою даних SQLite та програмним кодом настільного застосунку.

## 3 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 3.1 Логічна модель даних

Логічна модель даних визначає структуру збереження інформації у програмній системі, склад основних сутностей, їх атрибути, ключі та зв'язки між ними. Для системи збору відгуків та формування рейтингу риболовних локацій логічна модель була побудована з урахуванням фактичного призначення настільного застосунку, розробленого з використанням C#, Windows Forms та SQLite.

Основне завдання логічної моделі полягає в організації даних про користувачів, риболовні локації, відгуки та рейтингові оцінки. Модель повинна забезпечувати можливість реєстрації та авторизації користувачів, перегляду каталогу риболовних локацій, додавання відгуків, виставлення оцінок, пошуку локацій і сортування їх за рейтингом.

У розробленій логічній моделі виділено такі основні сутності:

- **Users** — користувачі системи;
- **Locations** — риболовні локації;
- **Reviews** — відгуки та оцінки користувачів;
- **LocationRating** — логічне представлення рейтингу локації.

Сутність **Users** призначена для збереження облікових даних користувачів. Вона містить первинний ключ Id, а також поля Username, Email і Password. Ці дані використовуються під час реєстрації та авторизації користувача у настільному застосунку.

Сутність **Locations** є центральною в логічній моделі, оскільки саме риболовна локація є основним об'єктом, з яким працює користувач. Для кожної локації зберігаються первинний ключ Id, назва Name, опис Description, а також координати Latitude і Longitude. Ці атрибути використовуються для відображення інформації про водойму у графічному інтерфейсі та для переходу до картографічного сервісу Google Maps.

Сутність **Reviews** використовується для збереження відгуків і рейтингових оцінок. Вона містить первинний ключ **Id**, зовнішній ключ **LocationId**, а також поля **Rating**, **Comment** і **Date**. Поле **LocationId** пов'язує відгук із конкретною риболовною локацією. Поле **Rating** зберігає числову оцінку за шкалою від 1 до 5, **Comment** містить текстовий відгук, а **Date** фіксує дату створення запису. Така структура відповідає поточній реалізації системи, у якій відгуки прив'язуються до локації, а не до конкретного користувача.

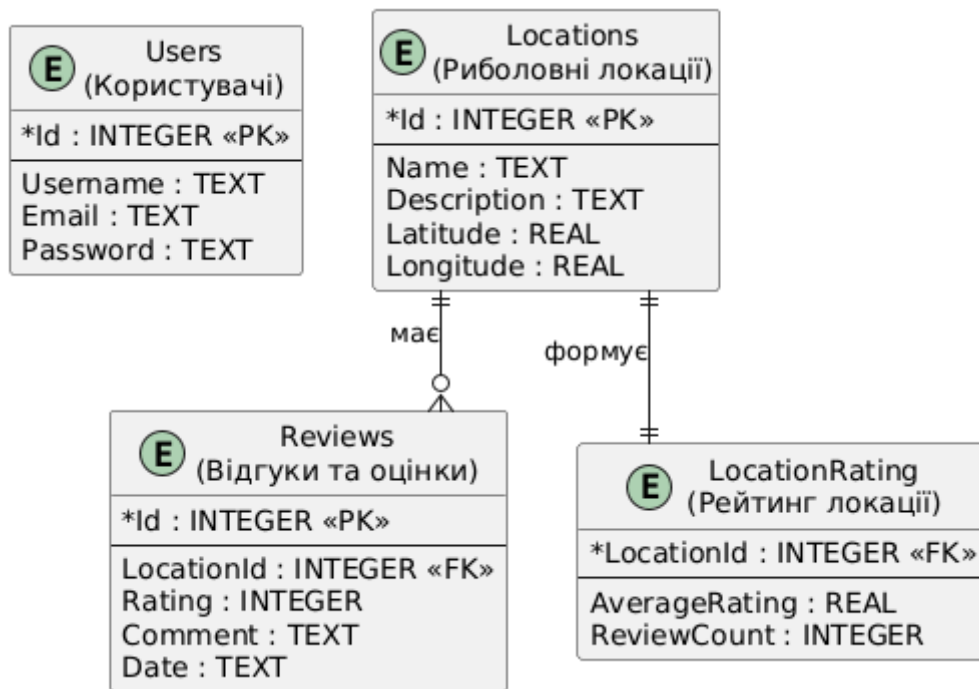


Рис. 3.1 ER-діаграма

На логічній моделі показано зв'язки між основними сутностями системи. Сутність **Users** використовується для збереження облікових записів користувачів і забезпечення авторизації, однак у поточній реалізації вона не має прямого зв'язку з таблицею **Reviews**. Між **Locations** і **Reviews** встановлено зв'язок типу «один-до-багатьох»: одна локація може мати багато відгуків, але кожен відгук стосується лише однієї конкретної локації. Зв'язок між **Locations** і **LocationRating** показує, що для кожної локації може бути сформований один узагальнений рейтинг.

Оскільки тема роботи пов'язана не лише зі збереженням відгуків, а й із формуванням рейтингу, важливо окремо визначити принцип його розрахунку. Середній рейтинг риболовної локації обчислюється за формулою:

$$R = \frac{\sum_{i=1}^n r_i}{n}$$

де  $R$  — середній рейтинг риболовної локації;

$r_i$  — окрема оцінка користувача;

$n$  — кількість оцінок, залишених для конкретної локації.

Наприклад, якщо для локації залишено оцінки 5, 4, 5 і 3, то середній рейтинг буде обчислено так:

$$R = \frac{5 + 4 + 5 + 3}{4} = 4,25$$

Рейтинг перераховується після додавання нового відгуку. У межах системи він не зберігається як окреме поле таблиці Locations, оскільки таке зберігання призвело б до дублювання даних. Натомість середній рейтинг обчислюється на основі значень поля Rating у таблиці Reviews для відповідної локації. Це забезпечує актуальність рейтингу та усуває ризик розбіжності між збереженим середнім значенням і фактичними оцінками.

Якщо для локації ще не залишено жодного відгуку, рейтинг не обчислюється, а в інтерфейсі відображається повідомлення про відсутність оцінок. Такий підхід дозволяє коректно обробляти нові локації, які ще не мають відгуків.

Окрему увагу було приділено відповідності логічної моделі принципам реляційної моделі даних і нормалізації. Кожна сутність представлена окремою таблицею, кожна таблиця має первинний ключ, а зв'язки між таблицями реалізуються за допомогою зовнішніх ключів. Це дозволяє забезпечити цілісність даних і уникнути надмірного дублювання інформації.

Логічна модель відповідає першій нормальній формі, оскільки всі атрибути таблиць містять атомарні значення. У таблицях відсутні списки значень в одному полі або повторювані групи атрибутів. Наприклад, кожен

відгук зберігається окремим записом у таблиці Reviews, а не як набір коментарів усередині таблиці Locations.

Модель відповідає другій нормальній формі, оскільки всі неключові атрибути повністю залежать від первинного ключа своєї таблиці. У таблиці Users ім'я, електронна пошта та пароль залежать від ідентифікатора користувача. У таблиці Locations назва, опис і координати залежать від ідентифікатора локації. У таблиці Reviews оцінка, коментар і дата створення залежать від ідентифікатора відгуку. Часткові залежності від частини складеного ключа відсутні, оскільки основні таблиці мають прості первинні ключі.

Модель також відповідає третій нормальній формі, оскільки в ній відсутні транзитивні залежності між неключовими атрибутами. Наприклад, у таблиці Reviews не зберігається назва локації, оскільки ці дані вже містяться у таблиці Locations. У таблиці відгуків зберігається лише ідентифікатор LocationId, який вказує на відповідну риболовну локацію. Це дозволяє уникнути дублювання даних і спрощує їх оновлення.

Середній рейтинг також не дублюється у таблиці локацій, оскільки є похідним значенням. Його логічне представлення через LocationRating використовується для пояснення результату обчислення, але фактичне значення формується динамічно на основі таблиці Reviews. Такий підхід підтримує нормалізовану структуру бази даних і знижує ризик появи неузгоджених даних.

Запропонована логічна модель охоплює основні інформаційні об'єкти системи та їх взаємозв'язки. Вона забезпечує збереження користувачів, риболовних локацій, відгуків і оцінок, а також дозволяє формувати рейтинг локацій на основі реальних оцінок і відгуків. Така структура відповідає принципам реляційної організації даних, задовольняє вимоги першої, другої та третьої нормальних форм і може бути реалізована засобами SQLite у настільному застосунку Windows Forms.

### 3.2 Вибір системи управління базою даних та її реалізація

Після побудови логічної моделі даних було виконано її реалізацію у вигляді локальної бази даних SQLite. Використання SQLite є доцільним для даного програмного застосунку, оскільки система реалізована як настільний застосунок Windows Forms і не потребує окремого серверного середовища для зберігання даних. База даних зберігається у файлі `fishing.db`, що забезпечує просте розгортання програми на персональному комп'ютері користувача.

Для організації підключення до бази даних у програмі використовується окремий клас `Database`, який відповідає за створення об'єкта підключення до SQLite. Такий підхід дозволяє відокремити логіку підключення до бази даних від інших частин програми та спростити подальшу підтримку коду. Підключення до бази даних виконується за допомогою бібліотеки `System.Data.SQLite`.

Структура бази даних включає таблиці для збереження користувачів, риболовних локацій та відгуків. Таблиця користувачів використовується для реєстрації й авторизації, таблиця локацій містить інформацію про доступні місця для риболовлі, а таблиця відгуків зберігає коментарі та рейтингові оцінки. Зв'язок між локаціями та відгуками забезпечується за допомогою поля `LocationId`, яке дозволяє отримувати відгуки лише для конкретної вибраної локації.

Для роботи з риболовними локаціями використовується сервісний клас `LocationService`. Він відповідає за додавання початкових тестових локацій до бази даних, отримання списку локацій та передачу цих даних до графічного інтерфейсу. При першому запуску програми сервіс перевіряє, чи є вже записи у таблиці локацій. Якщо таблиця порожня, до неї додаються початкові риболовні місця. Це дозволяє уникнути дублювання однакових записів під час повторного запуску програми.

Окремий сервісний клас `ReviewService` відповідає за роботу з відгуками та рейтинговими оцінками. Він реалізує додавання нового відгуку, отримання

списку відгуків для вибраної локації та обчислення середнього рейтингу. Після додавання нового відгуку система оновлює список коментарів і перераховує рейтинг локації, що дозволяє користувачу одразу бачити актуальне значення середньої оцінки.

Середній рейтинг локації не зберігається як окреме постійне поле таблиці Locations, а обчислюється на основі оцінок, що містяться у таблиці Reviews. Для цього використовується середнє значення поля Rating для конкретного LocationId. Такий підхід дозволяє уникнути дублювання даних і забезпечує відповідність рейтингу фактичним оцінкам користувачів.

На рівні програмної логіки рейтинг використовується не лише для відображення на формі детальної інформації про локацію, а й для сортування каталогу. Після завантаження списку локацій система для кожної з них отримує середній рейтинг через сервіс роботи з відгуками. Це значення додається до об'єкта локації та відображається у списку разом із назвою водойми. Якщо для локації ще немає оцінок, у списку виводиться відповідне повідомлення про відсутність рейтингу.

Функція пошуку реалізована на рівні графічного інтерфейсу користувача. Після завантаження списку локацій користувач може вводити текст у поле пошуку, а програма фільтрує наявний список за назвою або описом локації. Це дозволяє швидко знаходити потрібні риболовні місця без додаткових запитів до бази даних.

Сортування за рейтингом також виконується на рівні інтерфейсу після завантаження даних. Користувач може натиснути кнопку сортування, після чого список локацій впорядковується за спаданням середнього рейтингу. Якщо сортування скасовано, список повертається до стандартного відображення. Такий механізм покращує зручність використання системи, оскільки дозволяє швидко визначати найбільш популярні або високо оцінені риболовні локації.

Для додавання відгуку користувач відкриває форму детальної інформації про локацію, вводить текст коментаря та виставляє оцінку від 1 до

5. Перед збереженням система перевіряє, чи не є поле коментаря порожнім. Якщо відгук заповнено коректно, дані передаються до ReviewService, після чого зберігаються у базі даних SQLite. Після успішного додавання коментаря поле введення очищується, список відгуків оновлюється, а середній рейтинг перераховується.

Взаємодія між формами, сервісами та базою даних організована за принципом розділення відповідальності. Форми Windows Forms відповідають за відображення інформації та взаємодію з користувачем. Сервісні класи виконують обробку даних і звернення до бази. Моделі даних описують основні об'єкти системи, такі як користувач, локація та відгук. Завдяки такій структурі програмний код є більш зрозумілим, а зміни в одному модулі менше впливають на інші частини застосунку.

Наведена реалізація забезпечує збереження основних даних системи, підтримує перегляд і пошук риболовних локацій, додавання відгуків, формування середнього рейтингу та сортування локацій за оцінками користувачів. Це відповідає основній меті програмного застосунку — надати користувачу зручний інструмент для вибору риболовних місць на основі опису, відгуків та рейтингових оцінок.

Для підтвердження реалізації основних функцій роботи з даними нижче наведено фрагменти програмного коду, які демонструють підключення до бази даних SQLite, додавання відгуків, отримання списку відгуків, обчислення середнього рейтингу та фільтрацію локацій у каталозі.

### Лістинг 3.2.1 — Підключення до бази даних SQLite

```
using System.Data.SQLite;

namespace FishingLocationsApp.Data
{
    public class Database
    {
        private string connectionString =
            "Data Source=fishing.db;Version=3;Journal Mode=WAL;";

        public SQLiteConnection GetConnection()
        {
            return new SQLiteConnection(connectionString);
        }
    }
}
```

У цьому фрагменті реалізовано клас Database, який відповідає за створення підключення до локальної бази даних SQLite. Використання окремого класу для підключення дозволяє уникнути дублювання рядка з'єднання в різних частинах програми.

**Лістинг 3.2.2 — Додавання відгуку до бази даних**

```
public void AddReview(int locationId, int rating, string comment)
{
    using (var conn = new Database().GetConnection())
    {
        conn.Open();

        using (var cmd = new SQLiteCommand(
            "INSERT INTO Reviews (LocationId, Rating, Comment, Date) " +
            "VALUES (@l, @r, @c, @d)", conn))
        {
            cmd.Parameters.AddWithValue("@l", locationId);
            cmd.Parameters.AddWithValue("@r", rating);
            cmd.Parameters.AddWithValue("@c", comment);
            cmd.Parameters.AddWithValue("@d", DateTime.Now.ToString());

            cmd.ExecuteNonQuery();
        }
    }
}
```

Метод `AddReview()` використовується для збереження нового відгуку у таблиці `Reviews`. До бази передаються ідентифікатор локації, оцінка користувача, текст коментаря та дата створення відгуку. Для безпечної передачі значень застосовуються параметризовані SQL-запити.

**Лістинг 3.2.3 — Отримання відгуків для вибраної локації**

```

public List<Review> GetReviews(int locationId)
{
    var list = new List<Review>();

    using (var conn = new Database().GetConnection())
    {
        conn.Open();

        using (var cmd = new SQLiteCommand(
            "SELECT * FROM Reviews WHERE LocationId=@id", conn))
        {
            cmd.Parameters.AddWithValue("@id", locationId);

            using (var reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    list.Add(new Review
                    {
                        Id = reader.GetInt32(0),
                        LocationId = reader.GetInt32(1),
                        Rating = reader.GetInt32(2),
                        Comment = reader.GetString(3),
                        Date = reader.GetString(4)
                    });
                }
            }
        }
    }

    return list;
}

```

Метод `GetReviews()` отримує всі відгуки, що належать конкретній риболовній локації. Відбір виконується за полем `LocationId`, що забезпечує відображення на формі лише тих коментарів, які пов'язані з вибраною водоймою.

### Лістинг 3.2.4 — Обчислення середнього рейтингу локації

```

public double GetAverageRating(int locationId)
{
    using (var conn = new Database().GetConnection())
    {
        conn.Open();

        using (var cmd = new SQLiteCommand(
            "SELECT AVG(Rating) FROM Reviews WHERE LocationId=@id", conn))
        {
            cmd.Parameters.AddWithValue("@id", locationId);

            var result = cmd.ExecuteScalar();

            if (result != DBNull.Value)
            {
                return Convert.ToDouble(result);
            }

            return 0;
        }
    }
}

```

Метод **GetAverageRating()** реалізує формування рейтингу локації. Він обчислює середнє значення поля **Rating** для всіх відгуків, пов'язаних із конкретною локацією. Якщо для локації ще немає оцінок, метод повертає значення **0**, після чого в інтерфейсі відображається повідомлення про відсутність рейтингу.

```

private void ApplyFiltersAndSort()
{
    IEnumerable<Location> result = allLocations;

    string searchText = txtSearch.Text.Trim().ToLower();

    if (!string.IsNullOrEmpty(searchText))
    {
        result = result.Where(location =>
            (location.Name != null &&
             location.Name.ToLower().Contains(searchText)) ||
            (location.Description != null &&
             location.Description.ToLower().Contains(searchText))
        );
    }

    if (sortByRatingDescending)
    {
        result = result
            .OrderByDescending(location => location.AverageRating)
            .ThenBy(location => location.Name);
    }

    DisplayLocations(result.ToList());
}

```

У наведеному фрагменті реалізовано фільтрацію та сортування каталогу риболовних локацій. Пошук виконується за назвою або описом локації, а сортування — за спаданням середнього рейтингу. Завдяки цьому користувач може швидко знайти потрібну водойму або переглянути найвище оцінені локації.

Наведені фрагменти коду демонструють основну логіку роботи системи з даними: підключення до SQLite, збереження відгуків, отримання коментарів, обчислення рейтингу та обробку каталогу локацій. Саме ці методи забезпечують функціональність, пов'язану з формуванням рейтингу риболовних місць і зручним переглядом інформації у настільному застосунку.

### 3.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення визначає загальну структуру системи, склад її основних компонентів та принципи взаємодії між ними. Для системи збору відгуків та формування рейтингу риболовних локацій було обрано модульну архітектуру настільного застосунку, реалізованого з використанням мови програмування C#, технології Windows Forms та локальної бази даних SQLite.

Обрана архітектура не передбачає використання окремого вебсервера або клієнт-серверної взаємодії через мережу. Усі основні компоненти системи працюють у межах настільного застосунку, який встановлюється та запускається на персональному комп'ютері користувача. Дані зберігаються у локальному файлі бази даних `fishing.db`, а взаємодія з ними здійснюється через бібліотеку `System.Data.SQLite`.

Структура програмного забезпечення побудована з розділенням відповідальності між окремими модулями. Це дозволяє зробити код зрозумілішим, спростити супровід програми та забезпечити можливість подальшого розширення функціональності. У проєкті можна виділити такі основні компоненти:

- модуль графічного інтерфейсу користувача;
- модуль моделей даних;
- сервісний модуль;
- модуль доступу до бази даних.

Модуль графічного інтерфейсу користувача реалізований за допомогою Windows Forms і містить основні форми застосунку. Форма авторизації відповідає за вхід користувача до системи та реєстрацію нового облікового запису. Головна форма застосунку відображає каталог риболовних локацій, забезпечує пошук за назвою або описом, а також дозволяє сортувати локації за середнім рейтингом. Форма детального перегляду локації використовується для відображення опису водойми, списку відгуків, середнього рейтингу, додавання нового коментаря та переходу до Google Maps.

Модуль моделей даних містить класи, які описують основні об'єкти предметної області. До них належать User, Location та Review. Клас User використовується для збереження облікових даних користувача. Клас Location описує риболовну локацію та містить її назву, опис, координати й середній рейтинг, який використовується для відображення та сортування. Клас Review представляє відгук користувача, що містить оцінку, коментар, дату створення та прив'язку до конкретної локації.

Сервісний модуль відповідає за основну логіку роботи з даними. Клас LocationService забезпечує отримання списку риболовних локацій із бази даних та додавання початкових тестових записів при першому запуску програми. Клас ReviewService реалізує функції додавання відгуків, отримання списку коментарів для вибраної локації та обчислення середнього рейтингу. Окремо можна виділити логіку авторизації, яка забезпечує перевірку облікових даних користувача та створення нового запису в таблиці користувачів.

Модуль доступу до бази даних представлений класом Database, який формує підключення до локальної бази SQLite. Такий підхід дозволяє централізовано керувати рядком підключення та не дублювати його в різних частинах програми. Усі запити до бази даних виконуються через сервісні класи, що забезпечує чіткий поділ між інтерфейсом користувача та роботою з даними.

Особливе місце в архітектурі займає механізм формування рейтингу. Після завантаження списку локацій система для кожної з них отримує середній рейтинг на основі оцінок, збережених у таблиці Reviews. Це значення записується у відповідний об'єкт Location і використовується для відображення в каталозі та сортування локацій за рейтингом. Після додавання нового відгуку рейтинг перераховується, а список відгуків і оцінка локації оновлюються на формі детального перегляду.

Пошук локацій реалізований на рівні головної форми застосунку. Після завантаження каталогу користувач може ввести фрагмент назви або опису

локації у поле пошуку. Програма фільтрує вже завантажений список і відображає лише ті записи, які відповідають введеному запиту. Це дозволяє швидко знаходити потрібні водойми без створення додаткових складних запитів до бази даних.

Сортування за рейтингом також виконується у головній формі. Після натискання відповідної кнопки список локацій упорядковується за спаданням середнього рейтингу. Якщо для локації ще немає відгуків, у списку відображається повідомлення про відсутність оцінок. Така функція робить систему зручнішою, оскільки користувач може швидко визначити найбільш високо оцінені риболовні місця.

Загальна схема взаємодії компонентів системи має такий вигляд: користувач працює з формами Windows Forms, форми передають запити до сервісних класів, сервіси виконують операції з моделями даних і звертаються до SQLite через клас підключення до бази даних. Результати обробки повертаються назад до інтерфейсу та відображаються користувачу у вигляді списку локацій, відгуків, рейтингу або повідомлень про результат виконання дії.

Обрана архітектура відповідає призначенню розробленої системи, оскільки забезпечує просту локальну роботу з даними, не потребує складного серверного середовища та дозволяє реалізувати всі основні функції настільного застосунку: авторизацію, перегляд каталогу локацій, пошук, сортування за рейтингом, додавання відгуків, формування середньої оцінки та відкриття місцезнаходження водойми через Google Maps. Така структура є достатньо гнучкою для подальшого розвитку, наприклад додавання прив'язки відгуків до конкретного користувача, розширення каталогу локацій або впровадження додаткових аналітичних функцій.

Таблиця 3.2 Компоненти архітектури ПЗ

| Рівень / компонент              | Елементи реалізації                                | Призначення                                                                                                                                                     |
|---------------------------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Графічний інтерфейс користувача | Form1, MainForm, LocationDetailsForm               | Забезпечує взаємодію користувача із системою: авторизацію, перегляд каталогу, пошук, сортування, перегляд деталей локації, додавання відгуків і відкриття карти |
| Моделі даних                    | User, Location, Review                             | Описують основні об'єкти предметної області, які використовуються у програмі для збереження та передачі даних                                                   |
| Сервісний шар                   | LocationService, ReviewService, логіка авторизації | Виконує основну програмну логіку: отримання локацій, додавання тестових даних, збереження відгуків, отримання коментарів і обчислення середнього рейтингу       |
| Шар доступу до даних            | Database                                           | Забезпечує підключення до локальної бази даних SQLite та створення об'єкта з'єднання                                                                            |
| База даних                      | fishing.db                                         | Зберігає інформацію про користувачів, риболовні локації, відгуки та рейтингові оцінки                                                                           |
| Зовнішній сервіс                | Google Maps                                        | Використовується для відкриття місцезнаходження вибраної локації у веббраузері через сформоване посилання                                                       |

Наведена структура показує, що система побудована за принципом розділення відповідальності. Форми Windows Forms відповідають за взаємодію з користувачем, сервісні класи реалізують основну логіку роботи, моделі описують дані предметної області, а клас Database забезпечує доступ до локальної бази SQLite. Завдяки такій організації програмний код є зрозумілішим, а окремі частини застосунку можна змінювати без суттєвого впливу на інші модулі.

Наприклад, під час відкриття каталогу головна форма звертається до LocationService, який отримує список локацій із бази даних. Після цього для кожної локації через ReviewService визначається середній рейтинг, що

використовується для відображення у списку та сортування. При додаванні нового відгуку форма детальної інформації передає дані до ReviewService, сервіс зберігає запис у таблиці Reviews, після чого рейтинг локації оновлюється.

### 3.4 Організаційна структура програмного забезпечення

Організаційна структура програмного забезпечення відображає поділ проєкту на окремі логічні модулі, кожен із яких відповідає за певну частину функціональності системи. Для розробленого настільного застосунку збору відгуків та формування рейтингу риболовних локацій було використано модульний підхід, що дозволяє відокремити графічний інтерфейс, моделі даних, сервісну логіку та роботу з базою даних.

Проєкт реалізовано мовою C# у середовищі Visual Studio з використанням технології Windows Forms. Структура застосунку побудована таким чином, щоб кожна група файлів виконувала чітко визначену роль. Це спрощує супровід програмного коду, полегшує пошук потрібних компонентів і створює основу для подальшого розширення системи.

У структурі програмного забезпечення виділено такі основні пакети:

- Forms — містить форми графічного інтерфейсу користувача;
- Models — містить класи основних сутностей предметної області;
- Services — містить класи, що реалізують логіку роботи з даними;
- Data — містить клас для підключення до локальної бази даних SQLite.

Пакет **Forms** відповідає за взаємодію користувача із застосунком. До нього належать форма авторизації та реєстрації Form1, головна форма каталогу MainForm, а також форма детального перегляду локації LocationDetailsForm. Через ці форми користувач виконує вхід до системи, переглядає список риболовних локацій, здійснює пошук, сортування за

рейтингом, відкриває інформацію про водойму, додає відгуки та переходить до Google Maps.

Пакет **Models** містить класи, що описують основні об'єкти системи: User, Location та Review. Клас User використовується для збереження даних користувача, Location описує риболовну локацію, а Review представляє відгук та оцінку для конкретної водойми. Наявність окремого пакета моделей дозволяє використовувати ці класи в різних частинах програми без дублювання структури даних.

Пакет **Services** реалізує основну логіку роботи з даними. Клас LocationService відповідає за отримання списку локацій, додавання початкових даних і передачу інформації до головної форми. Клас ReviewService забезпечує додавання відгуків, отримання списку коментарів для вибраної локації та обчислення середнього рейтингу. Саме сервісний шар зв'язує інтерфейс користувача з базою даних і дозволяє не розміщувати SQL-запити безпосередньо у формах.

Пакет **Data** містить клас Database, який відповідає за створення підключення до локальної бази даних SQLite. Завдяки винесенню підключення в окремий клас спрощується робота з базою, оскільки всі сервіси використовують єдиний механізм доступу до файлу fishing.db.

Загальна організація проєкту забезпечує розділення відповідальності між компонентами. Форми відповідають за відображення інформації та обробку дій користувача, моделі описують структуру даних, сервіси виконують основну логіку, а пакет Data забезпечує доступ до бази даних. Такий підхід робить програму зрозумілішою та зменшує залежність між окремими частинами системи.

### 3.4.1 Діаграма пакетів

Для наочного представлення організаційної структури програмного забезпечення було розроблено діаграму пакетів. Вона показує, як основні модулі системи пов'язані між собою та який напрям взаємодії між ними використовується.

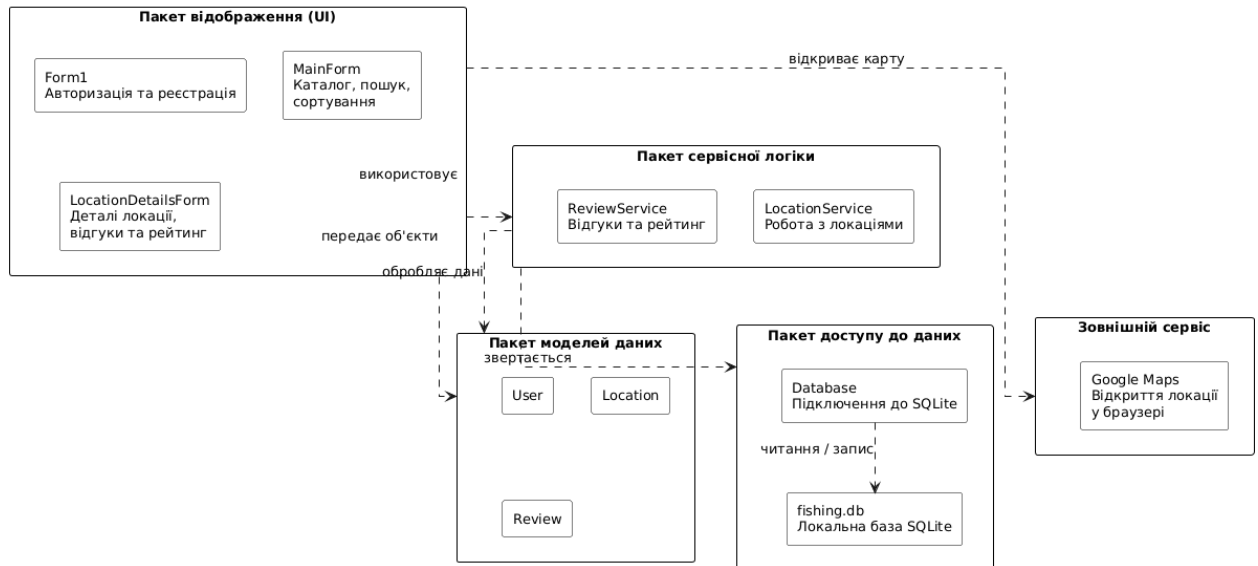


Рис. 3.2 Діаграма пакетів

На рис. 3.2 представлено організаційну структуру програмного забезпечення системи збору відгуків та формування рейтингу риболовних локацій. Діаграма відображає поділ настільного застосунку на окремі логічні пакети, кожен із яких відповідає за певну частину функціональності. Такий підхід дозволяє уникнути змішування інтерфейсу, бізнес-логіки та роботи з базою даних в одному місці, що позитивно впливає на зрозумілість і супровід програмного коду.

**Пакет відображення (UI)** містить форми графічного інтерфейсу користувача, створені за допомогою Windows Forms. До цього пакета належать Form1, MainForm та LocationDetailsForm. Форма Form1 відповідає за авторизацію та реєстрацію користувача. Після успішного входу відкривається MainForm, де користувач може переглядати каталог риболовних локацій, виконувати пошук за назвою або описом, а також сортувати локації за середнім рейтингом. Форма LocationDetailsForm використовується для

перегляду детальної інформації про вибрану локацію, відображення списку відгуків, додавання нового коментаря, виставлення оцінки та відкриття місцезнаходження локації у Google Maps.

Пакет **сервісної логіки** містить класи, які реалізують основні операції обробки даних. Клас `LocationService` відповідає за роботу з риболовними локаціями: отримання списку локацій із бази даних, додавання початкових записів та передавання інформації до головної форми застосунку. Клас `ReviewService` забезпечує роботу з відгуками та рейтингами: додавання нового відгуку, отримання списку коментарів для конкретної локації та обчислення середньої оцінки. Саме цей пакет виконує роль проміжного шару між інтерфейсом користувача та базою даних.

Пакет **моделей даних** містить класи, які описують основні об'єкти предметної області. Клас `User` використовується для представлення користувача системи, клас `Location` описує риболовну локацію, а клас `Review` відповідає за структуру відгуку. Завдяки винесенню моделей в окремий пакет ці класи можуть використовуватися як у формах інтерфейсу, так і в сервісах без дублювання коду.

Пакет **доступу до даних** відповідає за роботу з локальною базою даних `SQLite`. У ньому розміщено клас `Database`, який створює підключення до файлу бази даних `fishing.db`. Винесення підключення до бази в окремий пакет дозволяє централізовано керувати доступом до даних і не дублювати рядок підключення в різних частинах програми. Файл `fishing.db` зберігає інформацію про користувачів, риболовні локації, відгуки та оцінки.

Окремо на діаграмі виділено **зовнішній сервіс Google Maps**. Він не є частиною програмного застосунку, але використовується для відкриття місцезнаходження вибраної риболовної локації. Перехід до карти реалізовано шляхом формування пошукового посилання та відкриття його у веббраузері користувача. Такий спосіб не потребує складної інтеграції з Google Maps API, але забезпечує зручний доступ до картографічної інформації.

Зв'язки між пакетами показують напрям взаємодії компонентів. Пакет відображення звертається до пакета сервісної логіки для отримання або обробки даних. Наприклад, головна форма викликає методи `LocationService` для завантаження каталогу локацій, а також методи `ReviewService` для отримання середнього рейтингу. Пакет сервісної логіки, у свою чергу, використовує моделі даних і звертається до пакета доступу до даних для виконання операцій читання та запису в `SQLite`.

Під час додавання відгуку користувач вводить коментар і оцінку у формі `LocationDetailsForm`. Після цього форма передає дані до `ReviewService`, який зберігає новий запис у базі даних і забезпечує оновлення рейтингу. Під час перегляду каталогу `MainForm` отримує список локацій через `LocationService`, після чого для кожної локації визначається середній рейтинг. Це значення використовується для відображення рейтингу у списку та для сортування локацій за оцінками користувачів.

Пошук локацій реалізується на рівні пакета відображення після завантаження списку з бази даних. Користувач вводить частину назви або опису, а головна форма фільтрує наявний список. Сортування за рейтингом також виконується у головній формі, але використовує значення середнього рейтингу, отримане через сервісний шар. Завдяки цьому інтерфейс залишається зручним, а логіка обчислення рейтингу не дублюється у різних частинах програми.

Побудована організаційна структура демонструє поділ системи на зрозумілі функціональні частини. Інтерфейс користувача відповідає лише за взаємодію з користувачем, сервіси — за обробку даних, моделі — за опис об'єктів предметної області, а пакет доступу до даних — за підключення до `SQLite`. Такий поділ спрощує подальше розширення застосунку, наприклад додавання нових форм, додаткових сервісів або розширення структури бази даних без повної переробки вже реалізованих компонентів.

### **3.5 Вибір інструментарію для створення програмного забезпечення**

#### **C#:**

Основною мовою програмування було обрано C#, яка використовується для створення застосунків на платформі .NET і підтримує об'єктно-орієнтований підхід [1].

#### **.NET:**

Робота застосунку базується на платформі .NET, яка забезпечує середовище виконання та бібліотеки для розробки настільного програмного забезпечення [2].

#### **Windows Forms:**

Для створення графічного інтерфейсу користувача було використано Windows Forms, що є технологією .NET для розробки настільних застосунків Windows [3].

#### **Visual Studio:**

Середовищем розробки було обрано Microsoft Visual Studio, яке підтримує створення, налагодження та тестування застосунків мовою C# [4].

#### **System.Data.SQLite:**

Для взаємодії програми з локальною базою даних використано бібліотеку System.Data.SQLite [5].

#### **Google Maps:**

Перехід до місцезнаходження локації реалізовано через формування посилання Google Maps URLs та відкриття його у веббраузері користувача [9].

Для розробки програмного забезпечення системи збору відгуків та формування рейтингу риболовних локацій було обрано набір інструментів і технологій, які відповідають вимогам настільного застосунку. Основними критеріями вибору були простота розробки, підтримка графічного

інтерфейсу користувача, можливість роботи з локальною базою даних, зручність тестування та подальшого супроводу програмного продукту.

Оскільки система реалізується як настільний застосунок для операційної системи Windows, основною мовою програмування було обрано **C#**. Ця мова є зручною для створення прикладного програмного забезпечення, має зрозумілий синтаксис, підтримує об'єктно-орієнтований підхід і добре інтегрується з інструментами платформи .NET. Використання **C#** дозволило реалізувати основну логіку роботи програми: авторизацію користувача, завантаження списку локацій, додавання відгуків, обчислення рейтингу, пошук і сортування риболовних місць.

Для створення графічного інтерфейсу користувача було використано технологію **Windows Forms**. Вона дає змогу швидко створювати форми, кнопки, поля введення, списки та інші елементи інтерфейсу, необхідні для взаємодії користувача з програмою. У межах розробленої системи Windows Forms використовується для реалізації форми входу та реєстрації, головної форми з каталогом риболовних локацій, а також форми детального перегляду локації з відгуками, рейтингом і кнопкою переходу до карти.

Середовищем розробки було обрано **Microsoft Visual Studio 2022**. Це інтегроване середовище розробки забезпечує зручне написання, редагування, налагодження та тестування програмного коду. Visual Studio також надає візуальний конструктор форм Windows Forms, що спрощує створення інтерфейсу користувача та дозволяє швидко налаштовувати властивості елементів керування. Крім того, середовище підтримує роботу з пакетами NuGet, що є важливим для підключення бібліотек, зокрема для роботи з SQLite.

Для збереження даних було використано локальну реляційну базу даних **SQLite**. Вибір SQLite обумовлений тим, що система не потребує окремого серверного середовища та працює як настільний застосунок. База даних зберігається у файлі `fishing.db`, що спрощує розгортання програми та забезпечує автономну роботу на комп'ютері користувача. У базі даних

зберігаються відомості про користувачів, риболовні локації, відгуки та оцінки.

Для взаємодії програми з базою даних використовується бібліотека **System.Data.SQLite**. Вона дозволяє створювати підключення до SQLite, виконувати SQL-запити, додавати нові записи, отримувати дані з таблиць і обчислювати середні значення рейтингу. У програмі ця бібліотека застосовується в сервісних класах, які відповідають за роботу з локаціями та відгуками. Завдяки цьому форми інтерфейсу не працюють із базою напряду, а звертаються до відповідних сервісів.

Окрему роль у системі відіграє сервіс **Google Maps**, який використовується для перегляду місцезнаходження вибраної риболовної локації. У застосунку не реалізовується повна інтеграція з Google Maps API, оскільки для поставлених завдань достатньо формування пошукового посилання за назвою локації та відкриття його у веббраузері користувача. Такий підхід спрощує реалізацію, але водночас забезпечує зручний перехід до карти.

Для організації структури програмного коду було використано поділ на окремі пакети: Forms, Models, Services та Data. Пакет Forms містить форми графічного інтерфейсу, пакет Models описує основні об'єкти системи, пакет Services реалізує логіку роботи з локаціями, відгуками та рейтингом, а пакет Data відповідає за підключення до бази даних. Така структура забезпечує розділення відповідальності між частинами програми та робить код більш зрозумілим для подальшого супроводу.

Використання обраного інструментарію дозволило реалізувати основні функціональні можливості системи: реєстрацію та авторизацію користувача, автоматичне завантаження каталогу риболовних локацій, пошук за назвою або описом, сортування локацій за середнім рейтингом, перегляд детальної інформації про водойму, додавання відгуків, виставлення оцінки від 1 до 5, формування рейтингу та відкриття локації у Google Maps.

## 4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ

### 4.1 Вимоги до апаратного та програмного забезпечення

Для коректної роботи розробленої системи збору відгуків та формування рейтингу риболовних локацій необхідно забезпечити відповідність комп'ютера користувача мінімальним апаратним і програмним вимогам. Оскільки програмний продукт реалізовано як настільний застосунок Windows Forms, його робота не потребує окремого вебсервера, серверної бази даних або постійного підключення до мережі Інтернет.

Застосунок призначений для запуску на персональному комп'ютері під керуванням операційної системи Windows. Основні функції програми, зокрема авторизація користувача, перегляд каталогу риболовних локацій, пошук, сортування за рейтингом, додавання відгуків і формування середнього рейтингу, виконуються локально. Дані зберігаються у файлі бази даних SQLite fishing.db.

Підключення до Інтернету не є обов'язковим для основної роботи програми. Воно потрібне лише для функції відкриття місцезнаходження вибраної риболовної локації у сервісі Google Maps. У разі відсутності Інтернету користувач може продовжувати працювати з каталогом, переглядати збережені локації, додавати відгуки та користуватися функціями рейтингу, однак карта у браузері може не завантажитися.

Таблиця 4.1 Мінімальні апаратні та програмні вимоги.

| Компонент          | Мінімальна вимога                           |
|--------------------|---------------------------------------------|
| Операційна система | Windows 10 або новіша версія                |
| Процесор           | Двоядерний процесор із частотою від 1,6 ГГц |

| Компонент             | Мінімальна вимога                                                   |
|-----------------------|---------------------------------------------------------------------|
| Оперативна пам'ять    | Не менше 4 ГБ                                                       |
| Вільне місце на диску | Не менше 100 МБ                                                     |
| Програмна платформа   | Встановлений .NET Runtime / .NET Desktop Runtime відповідної версії |
| База даних            | Локальний файл SQLite fishing.db                                    |
| Додаткові бібліотеки  | System.Data.SQLite.dll та пов'язані DLL-файли                       |
| Інтернет-з'єднання    | Потрібне лише для відкриття Google Maps                             |
| Пристрій введення     | Клавіатура та миша                                                  |
| Дисплей               | Роздільна здатність екрана не нижче 1366×768                        |

Наведені вимоги є достатніми для запуску та стабільної роботи програмного застосунку. Оскільки система використовує локальну базу даних SQLite, вона не потребує встановлення окремої серверної СУБД. Це спрощує розгортання програми та дозволяє використовувати її на звичайному персональному комп'ютері без додаткового адміністрування серверного середовища.

Для запуску застосунку на комп'ютері користувача має бути встановлено середовище виконання .NET. Якщо відповідна версія .NET Runtime або .NET Desktop Runtime відсутня, програма може не запуститися. У такому випадку користувачу необхідно встановити відповідне середовище виконання, після чого застосунок можна запускати повторно.

Робота з базою даних здійснюється через файл fishing.db, який повинен знаходитися поруч із виконуваним файлом програми або в іншому передбаченому програмою каталозі. У цьому файлі зберігаються дані про користувачів, риболовні локації, відгуки та оцінки. Для взаємодії з базою використовується бібліотека System.Data.SQLite, яка має бути включена до складу інсталяційного пакету.

Окремою вимогою є наявність веббраузера для відкриття локацій у Google Maps. Програма формує пошукове посилання на основі назви вибраної локації та відкриває його у браузері. Ця функція є допоміжною і не впливає на основні можливості застосунку, пов'язані з переглядом локацій, відгуками, пошуком і сортуванням.

Для наочного представлення середовища виконання програмного застосунку було розроблено діаграму розгортання. Вона показує, які програмні компоненти розміщуються на індивідуальному пристрої користувача, як вони взаємодіють між собою та яким чином здійснюється доступ до локальної бази даних і зовнішнього картографічного сервісу.

Оскільки розроблена система є настільним застосунком Windows Forms, усі основні компоненти програми виконуються локально на комп'ютері користувача. Для роботи застосунку необхідне програмне середовище .NET, у межах якого запускається виконуваний файл FishingLocationsApp.exe. Застосунок взаємодіє з окремими сервісними модулями, які відповідають за авторизацію, роботу з риболовними локаціями та обробку відгуків.

Збереження даних здійснюється у локальній базі даних SQLite fishing.db. До неї надсилаються SQL-запити для реєстрації користувачів, отримання списку локацій, збереження відгуків та обчислення рейтингових оцінок. Окремо на діаграмі показано взаємодію із сервісом Google Maps, яка реалізується через відкриття HTTPS-посилання на вибрану локацію у веббраузері користувача.

Розроблена діаграма розгортання представлена на рис. 4.1.

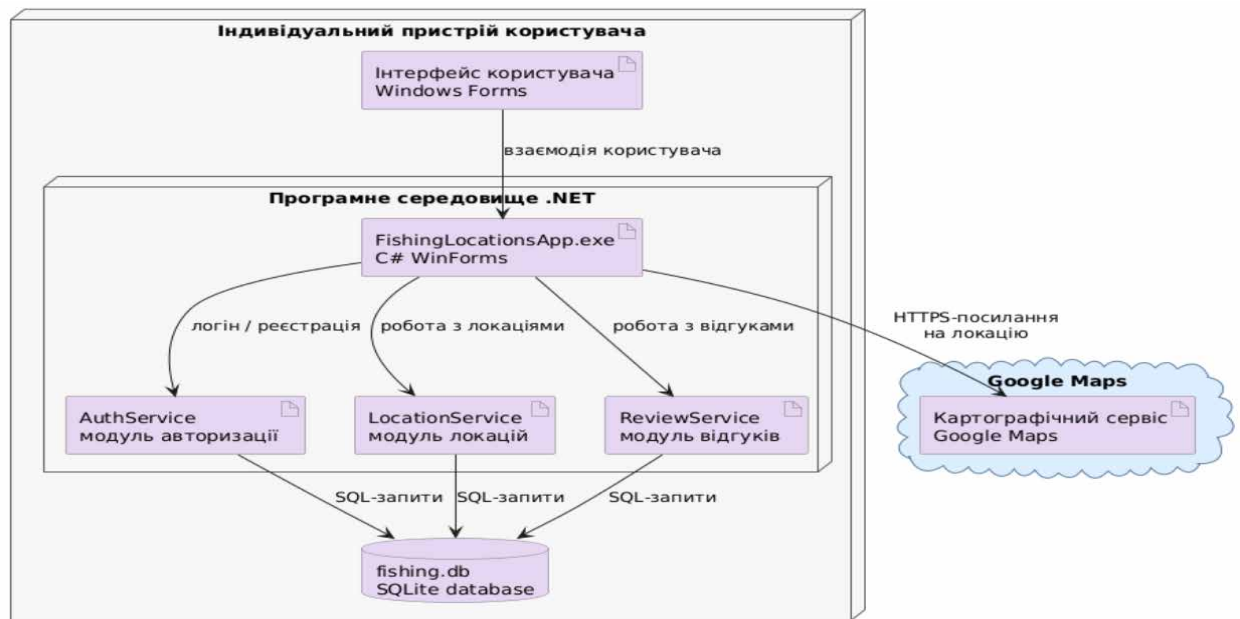


Рис. 4.1 Діаграма розгортання

На рис. 4.1 показано, що система розгортається на індивідуальному пристрої користувача та не потребує окремого серверного середовища. У межах програмного середовища .NET запускається основний виконуваний файл FishingLocationsApp.exe, який забезпечує роботу графічного інтерфейсу Windows Forms і взаємодію з внутрішніми модулями програми.

Модуль AuthService відповідає за процеси входу та реєстрації користувача.

Для запобігання дублюванню облікових записів у системі було реалізовано перевірку наявності користувача з таким самим email перед виконанням операції реєстрації. Перед додаванням нового запису до таблиці Users система виконує SQL-запит, який підраховує кількість користувачів із введеною електронною поштою. Якщо такий запис уже існує, реєстрація не виконується, а користувачу відображається повідомлення про помилку.

Такий підхід дозволяє уникнути створення кількох облікових записів з однаковою електронною поштою та забезпечує коректну роботу механізму авторизації.

Модуль LocationService забезпечує роботу з риболовними локаціями, зокрема отримання списку локацій із бази даних. Модуль ReviewService

використовується для додавання відгуків, отримання коментарів та формування середнього рейтингу локацій. Усі ці модулі звертаються до локальної бази даних `fishing.db` за допомогою SQL-запитів.

База даних SQLite зберігається локально на пристрої користувача, що спрощує встановлення та експлуатацію програми. Такий підхід дозволяє використовувати застосунок без налаштування окремого сервера або серверної СУБД. Основні функції програми, зокрема перегляд локацій, пошук, сортування за рейтингом, додавання відгуків і перегляд оцінок, виконуються локально.

Зовнішній сервіс Google Maps використовується лише для відкриття місцезнаходження вибраної риболовної локації. Застосунок формує HTTPS-посилання та передає його до веббраузера. За відсутності підключення до Інтернету основні функції програми залишаються доступними, однак карта може не завантажитися. Така структура розгортання відповідає призначенню настільного застосунку та забезпечує простоту його встановлення й використання.

## **4.2 Склад інсталяційного пакету та порядок запуску програми**

Для впровадження розробленого програмного забезпечення необхідно підготувати набір файлів, які забезпечують запуск і стабільну роботу настільного застосунку на комп'ютері користувача. Оскільки система збору відгуків та формування рейтингу риболовних локацій реалізована як Windows Forms застосунок, її розгортання не потребує окремого вебсервера або встановлення серверної системи керування базами даних.

Інсталяційний пакет має містити виконуваний файл програми, локальну базу даних SQLite, необхідні бібліотеки, конфігураційні файли та коротку інструкцію із запуску. Усі ці компоненти можуть бути розміщені в одній папці або передані користувачу у вигляді архіву.

Для зручності склад інсталяційного пакету наведено у табл. 4.2.

Таблиця 4.2 Склад інсталяційного пакету програмного застосунку

| Компонент                      | Назва / приклад файлу          | Призначення                                                                |
|--------------------------------|--------------------------------|----------------------------------------------------------------------------|
| Виконуваний файл програми      | FishingLocationsApp.exe        | Основний файл запуску настільного застосунку                               |
| Локальна база даних            | fishing.db                     | Зберігає дані про користувачів, риболовні локації, відгуки та оцінки       |
| Бібліотека для роботи з SQLite | System.Data.SQLite.dll         | Забезпечує взаємодію програми з базою даних SQLite                         |
| Додаткові DLL-файли            | супровідні бібліотеки .dll     | Забезпечують коректну роботу залежностей застосунку                        |
| Конфігураційний файл           | FishingLocationsApp.exe.config | Містить службові параметри запуску програми                                |
| Інструкція користувача         | README.txt                     | Описує порядок запуску та основні умови роботи програми                    |
| Архів із програмою             | FishingLocationsApp.zip        | Використовується для зручного перенесення або передачі готового застосунку |

Основним файлом інсталяційного пакету є FishingLocationsApp.exe. Саме цей файл запускає графічний інтерфейс програми та забезпечує доступ до основних функцій системи: авторизації, перегляду каталогу локацій, пошуку, сортування за рейтингом, перегляду детальної інформації, додавання відгуків і відкриття локації у Google Maps.

Файл fishing.db є локальною базою даних SQLite. У ньому зберігаються таблиці, необхідні для роботи програми. До них належать таблиці користувачів, риболовних локацій та відгуків. Наявність цього файлу є важливою умовою коректної роботи системи, оскільки саме з нього програма отримує збережені дані та до нього записує нову інформацію.

Бібліотека `System.Data.SQLite.dll` та пов'язані з нею DLL-файли потрібні для виконання SQL-запитів до бази даних SQLite. Якщо ці файли відсутні в папці з програмою, застосунок може не запуститися або не зможе виконувати операції читання та запису даних.

Конфігураційний файл `FishingLocationsApp.exe.config` використовується для збереження службової інформації, необхідної для запуску застосунку в середовищі .NET. У більшості випадків користувачу не потрібно редагувати цей файл вручну, однак він має бути включений до складу пакету разом із виконуваним файлом.

Для зручності користувача доцільно додати файл `README.txt`, у якому коротко описати порядок запуску програми, вимоги до середовища виконання та можливі дії у разі помилок. Це спрощує встановлення застосунку на іншому комп'ютері та зменшує кількість помилок під час першого запуску.

Порядок запуску програмного застосунку є таким:

1. Розпакувати архів `FishingLocationsApp.zip` у зручну папку на комп'ютері користувача.
2. Переконавшись, що в папці наявний файл `FishingLocationsApp.exe`.
3. Перевірити наявність файлу бази даних `fishing.db`.
4. Переконавшись, що в папці присутні необхідні DLL-бібліотеки, зокрема `System.Data.SQLite.dll`.
5. За потреби встановити .NET Runtime або .NET Desktop Runtime відповідної версії.
6. Запустити файл `FishingLocationsApp.exe`.
7. Виконати реєстрацію нового користувача або авторизацію з існуючими обліковими даними.
8. Після входу до системи перейти до каталогу риболовних локацій.

Якщо на комп'ютері користувача не встановлено необхідну версію .NET Runtime, програма може не запуститися. У такому випадку потрібно встановити відповідний **.NET Desktop Runtime**, оскільки застосунок

використовує графічний інтерфейс Windows Forms. Після встановлення середовища виконання додаткове налаштування програми не потрібне.

Під час запуску програма звертається до локальної бази даних fishing.db. Якщо файл бази даних відсутній або пошкоджений, застосунок може не мати доступу до збережених користувачів, локацій та відгуків. Тому під час перенесення програми на інший комп'ютер необхідно передавати не лише .exe файл, а й файл бази даних та супровідні бібліотеки.

Окреме встановлення SQLite на комп'ютер користувача не потрібне, оскільки робота з базою даних здійснюється через бібліотеку System.Data.SQLite, яка входить до складу інсталяційного пакету. Це спрощує розгортання системи та дозволяє запускати застосунок без налаштування серверного середовища.

Підключення до Інтернету не є обов'язковим для основної роботи програми. Користувач може переглядати каталог локацій, здійснювати пошук, сортувати записи за рейтингом, додавати відгуки та переглядати оцінки без доступу до мережі. Інтернет потрібен лише для відкриття вибраної локації у сервісі Google Maps.

Отже, інсталяційний пакет програмного застосунку має містити всі необхідні файли для автономного запуску системи на персональному комп'ютері. Такий спосіб розгортання є зручним для настільного застосунку, оскільки не потребує складного встановлення, налаштування сервера або підключення до зовнішньої бази даних.

### **4.3 Тестування системи**

Після завершення розробки програмного забезпечення було проведено тестування системи збору відгуків та формування рейтингу риболовних локацій. Метою тестування була перевірка працездатності основних функцій програми, зокрема авторизації користувача, реєстрації нового облікового

запису, автоматичного завантаження каталогу локацій, пошуку, сортування за рейтингом, перегляду детальної інформації про локацію, додавання відгуків, формування середнього рейтингу та переходу до місцезнаходження водойми через Google Maps.

Тестування проводилося шляхом послідовного виконання типових дій користувача у графічному інтерфейсі настільного застосунку. Окремо було перевірено як позитивні сценарії роботи системи, так і ситуації, пов'язані з некоректним введенням даних або відсутністю необхідних умов для виконання певної дії.

Після запуску програми користувач потрапляє на форму авторизації. На ній передбачено поля для введення електронної пошти та пароля, а також кнопки для входу в систему та реєстрації нового користувача. Під час тестування було введено тестові облікові дані, після чого система успішно виконала авторизацію та відкрила головне вікно програми.

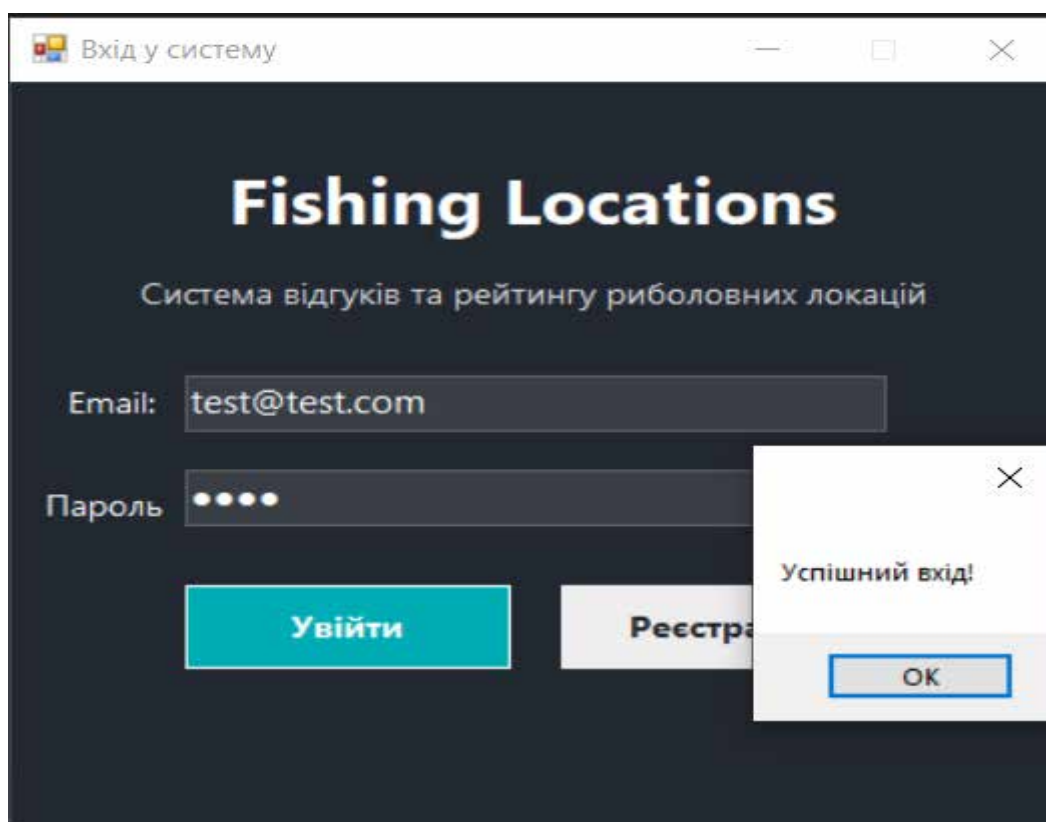


Рис. 4.2 Форма авторизації користувача

Окремо було перевірено процес реєстрації нового користувача. Для цього у відповідні поля було введено електронну пошту та пароль, після чого натиснуто кнопку реєстрації. У разі коректного заповнення даних система створює новий обліковий запис і виводить повідомлення про успішне виконання операції.

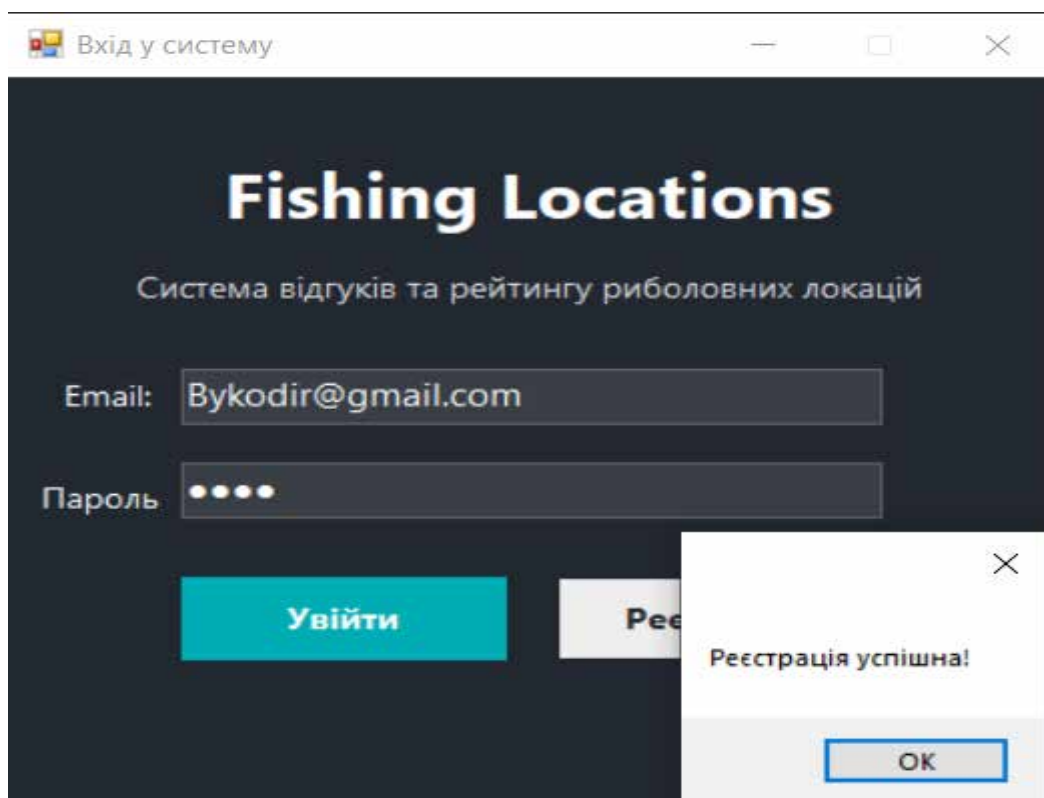


Рис. 4.3 Реєстрація нового користувача

Після успішної авторизації відкривається головне вікно програми з каталогом риболовних локацій. Список локацій завантажується автоматично, без необхідності натискання додаткової кнопки. У каталозі відображаються назви риболовних місць, а також середній рейтинг або повідомлення про відсутність оцінок. Це дозволяє користувачу одразу отримати загальне уявлення про доступні локації.

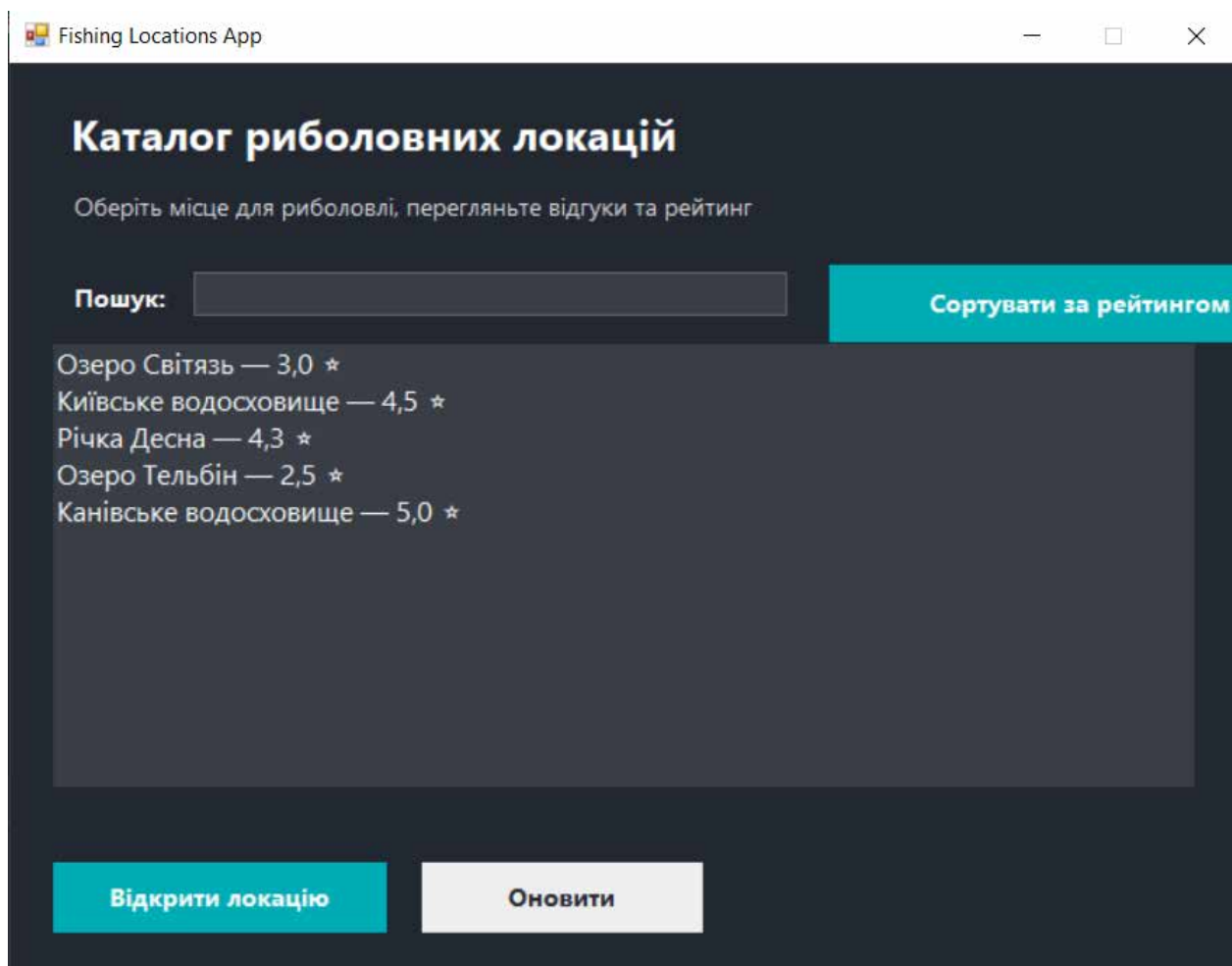


Рис. 4.4 Головне вікно з каталогом риболовних локацій

Далі було протестовано функцію пошуку риболовної локації. Для цього у поле пошуку було введено частину назви локації. Система виконала фільтрацію списку та залишила тільки ті записи, які відповідають введеному запиту. Це підтверджує коректну роботу пошуку за назвою або описом локації.

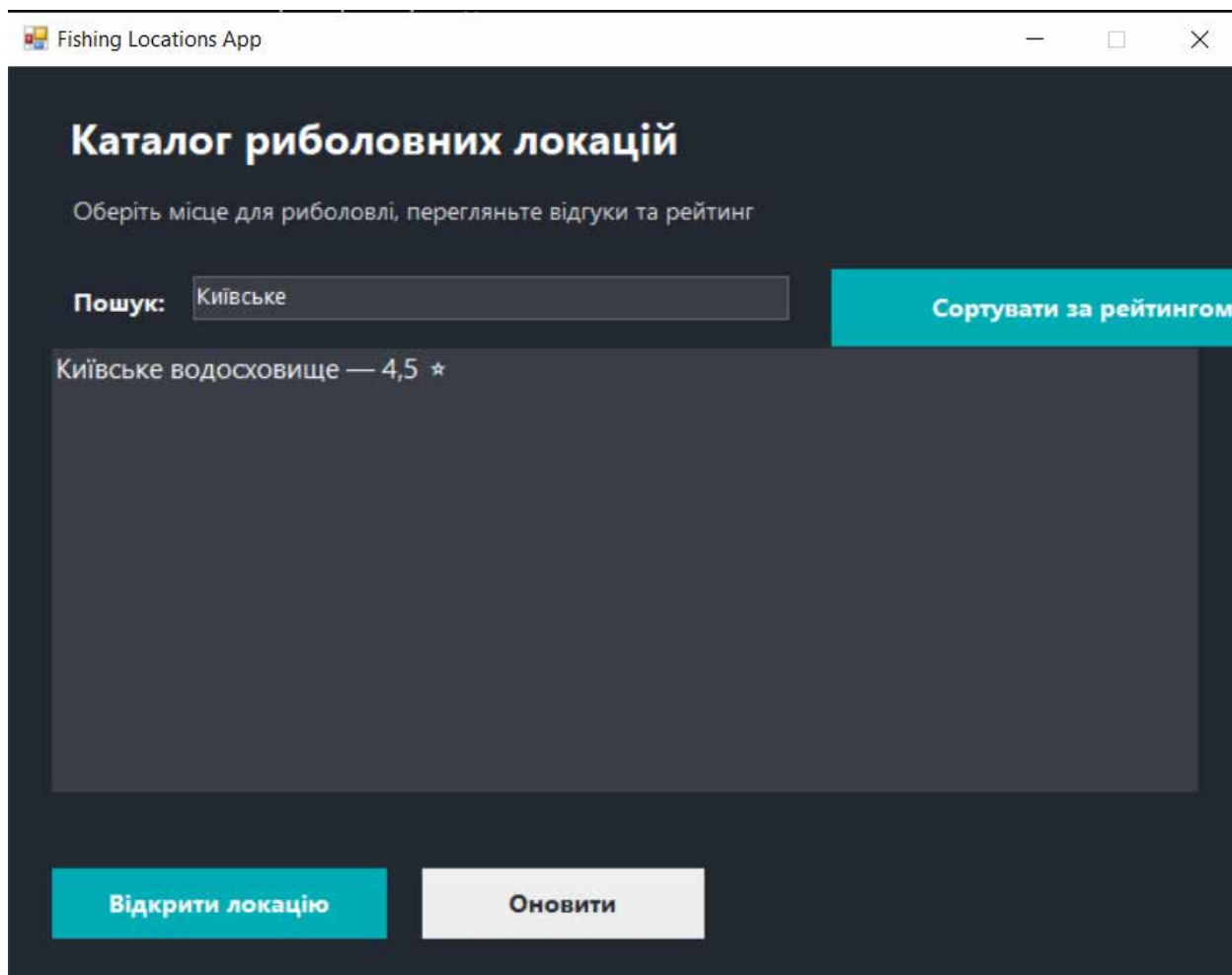


Рис. 4.5 Пошук риболовної локації за назвою

Також було перевірено ситуацію, коли користувач вводить у поле пошуку назву, якої немає в каталозі. У цьому випадку система не завершує роботу з помилкою, а виводить повідомлення про те, що локації не знайдено. Така поведінка є важливою для зручності користувача, оскільки система коректно реагує на відсутність результатів пошуку.

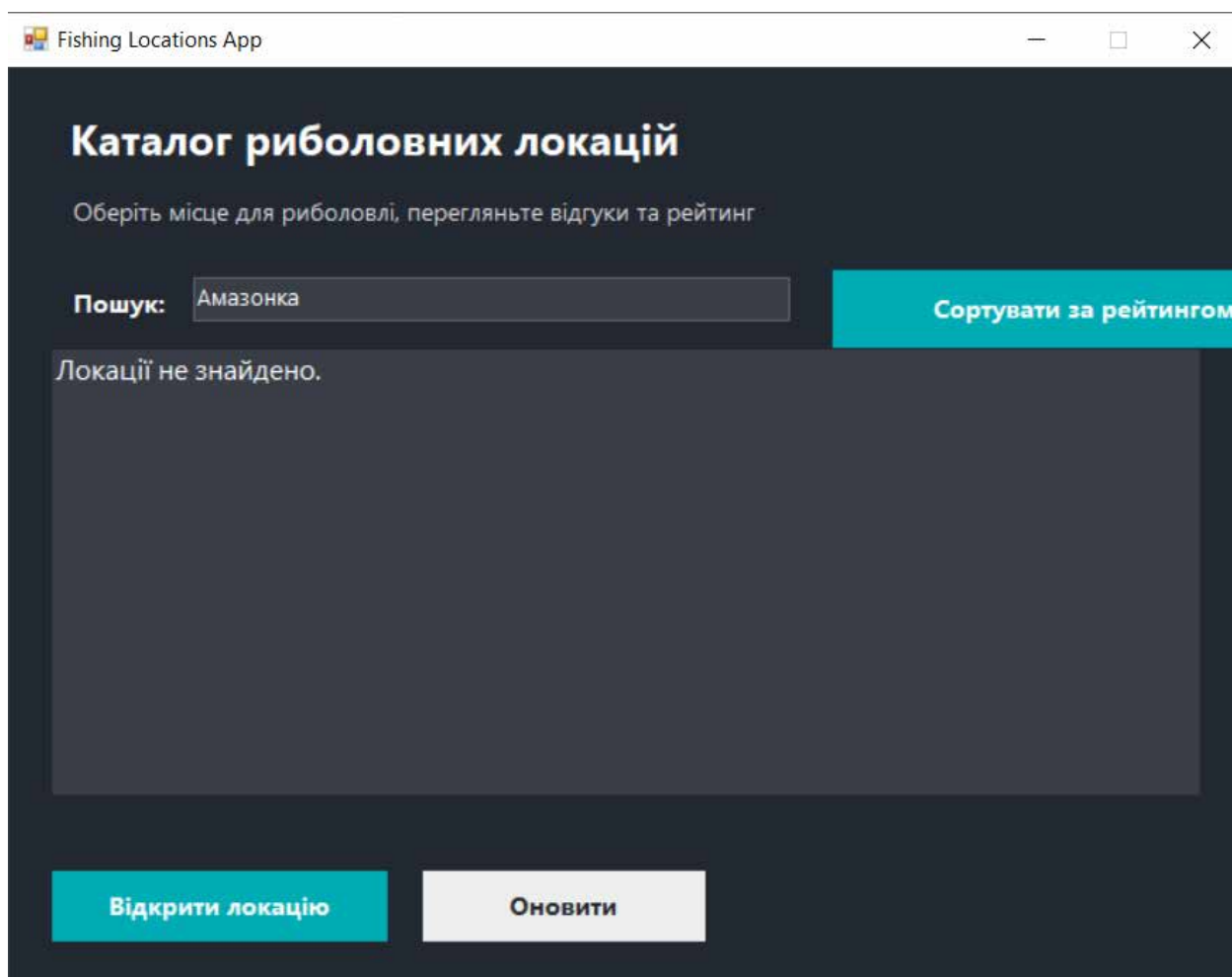


Рис. 4.6 Повідомлення про відсутність результатів пошуку

Наступним етапом було протестовано сортування локацій за рейтингом. Після натискання кнопки «Сортувати за рейтингом» система впорядковує список риболовних локацій за спаданням середнього рейтингу. Така функція дозволяє користувачу швидко визначити найбільш високо оцінені місця для риболовлі.

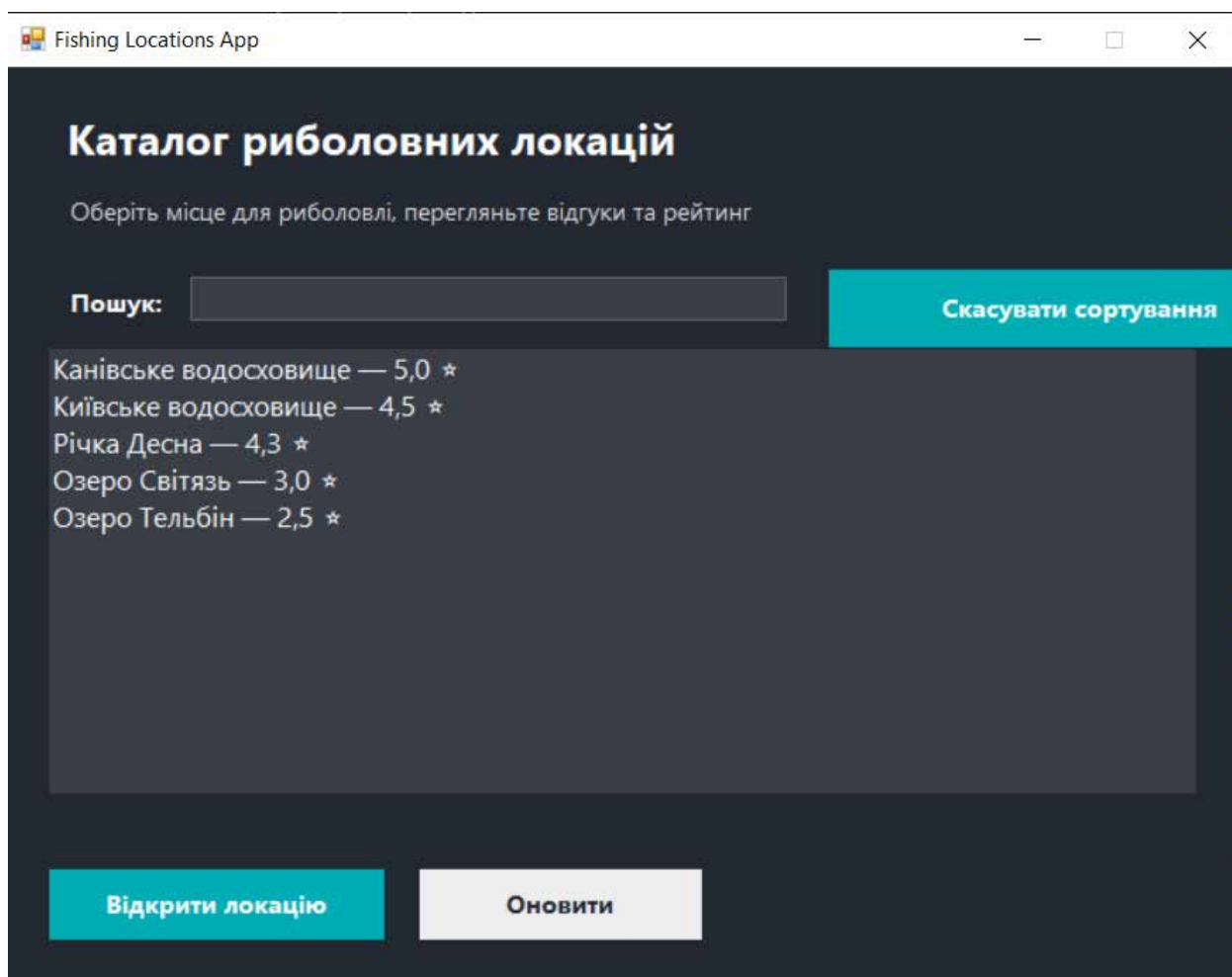


Рис. 4.7 Сортування локацій за середнім рейтингом

Після вибору конкретної локації користувач може відкрити форму детальної інформації. На цій формі відображається назва водойми, опис, середній рейтинг, список відгуків, поле для введення нового коментаря, елемент вибору оцінки та кнопка для переходу до Google Maps. Під час тестування було перевірено, що інформація про вибрану локацію відображається коректно.

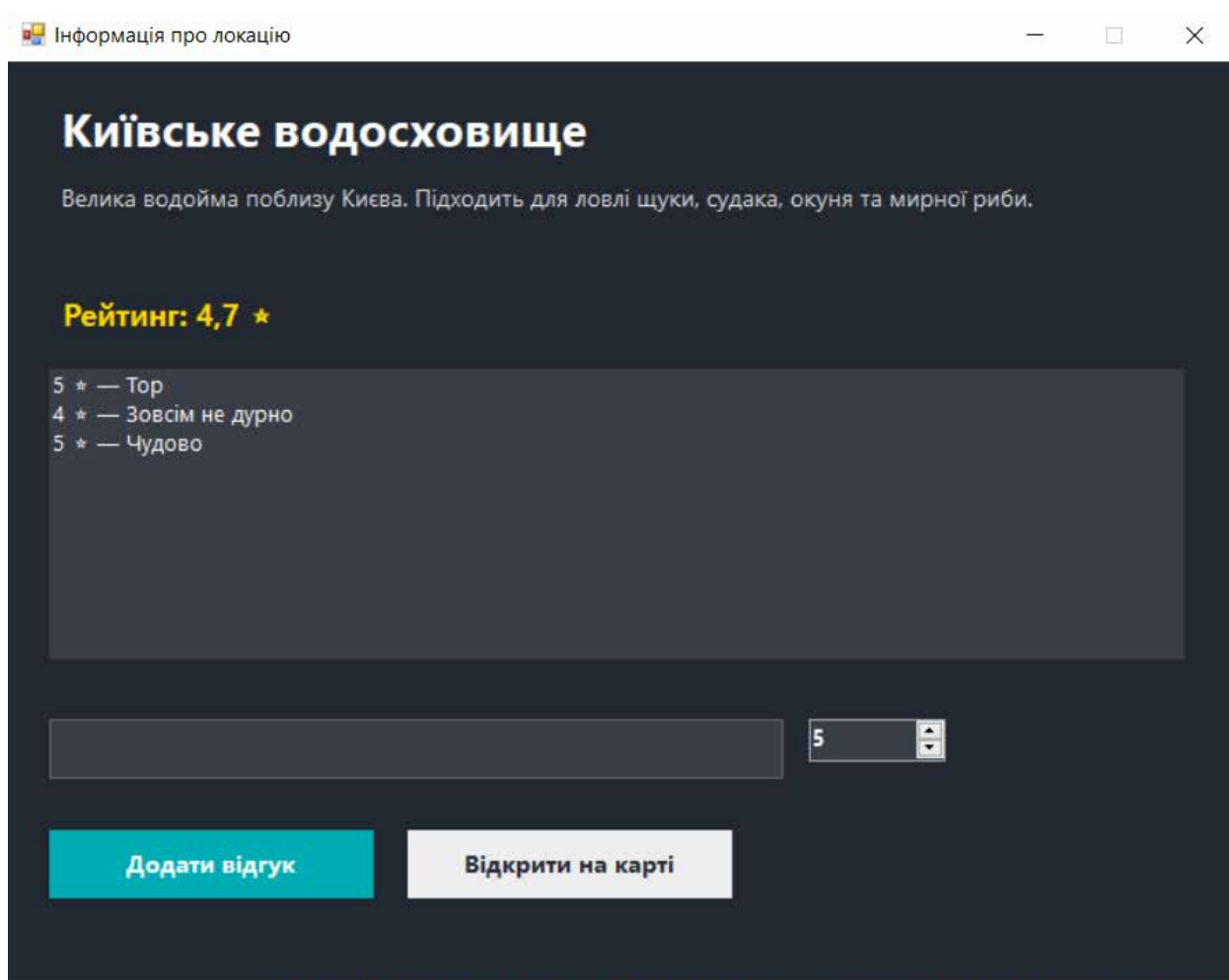


Рис. 4.8 Форма детальної інформації про риболовну локацію

Далі було протестовано функцію додавання відгуку. Користувач вводить текст коментаря, обирає оцінку за шкалою від 1 до 5 і натискає кнопку додавання відгуку. Після цього система зберігає коментар у базі даних, оновлює список відгуків і перераховує середній рейтинг локації. Після успішного додавання відгуку поле коментаря очищується, що підтверджує коректну обробку введених даних.

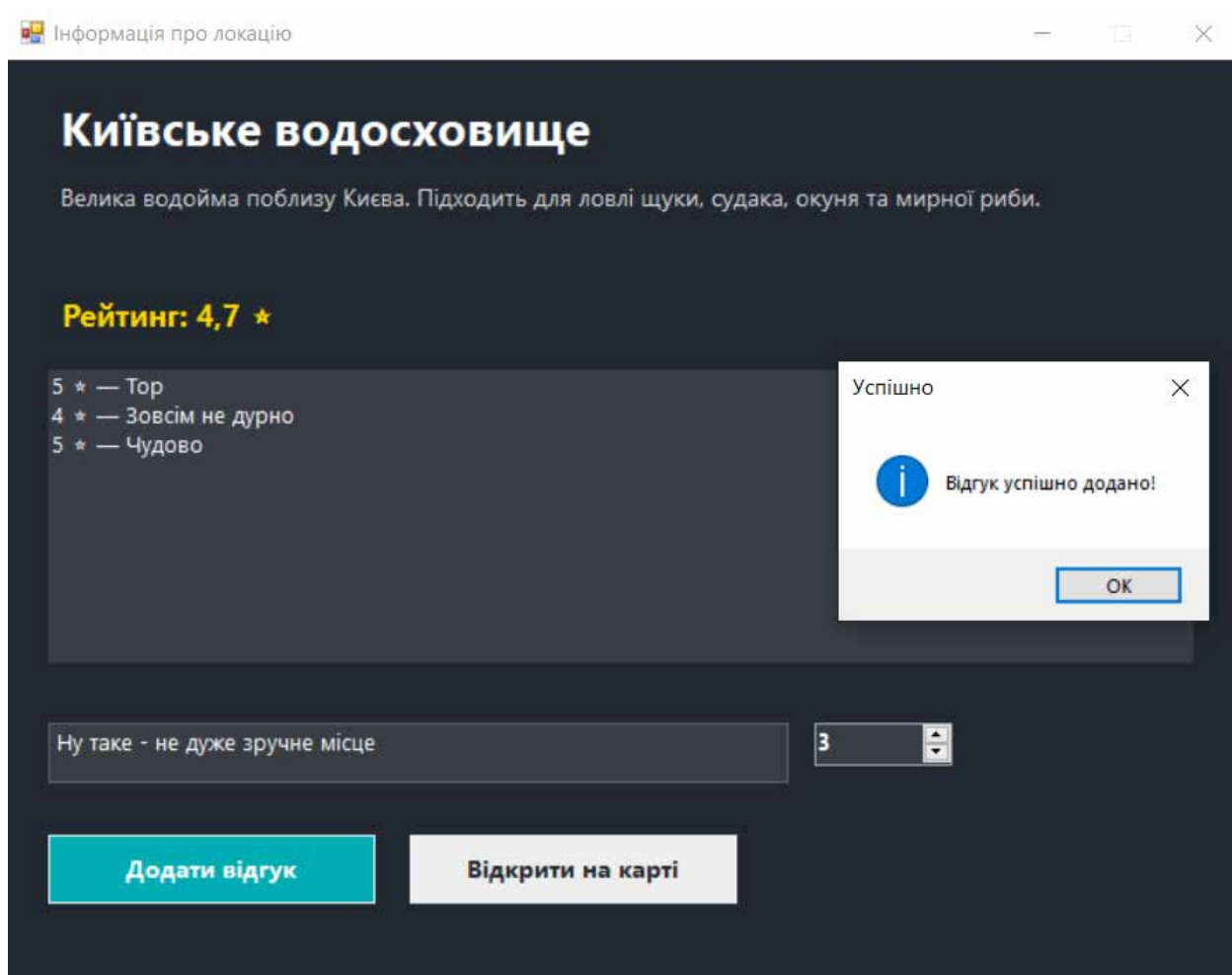


Рис. 4.9 Додавання відгуку та оновлення рейтингу локації

Окремо було перевірено обмеження для рейтингової оцінки. Поле оцінки реалізоване таким чином, що користувач може обрати значення лише в межах від 1 до 5. Це запобігає введенню некоректних рейтингових значень і забезпечує правильне формування середнього рейтингу.

Було також протестовано ситуацію додавання порожнього коментаря. Якщо користувач натискає кнопку «Додати відгук», не ввівши текст, система виводить попередження про необхідність заповнення поля коментаря. При цьому порожній запис не зберігається у базі даних.

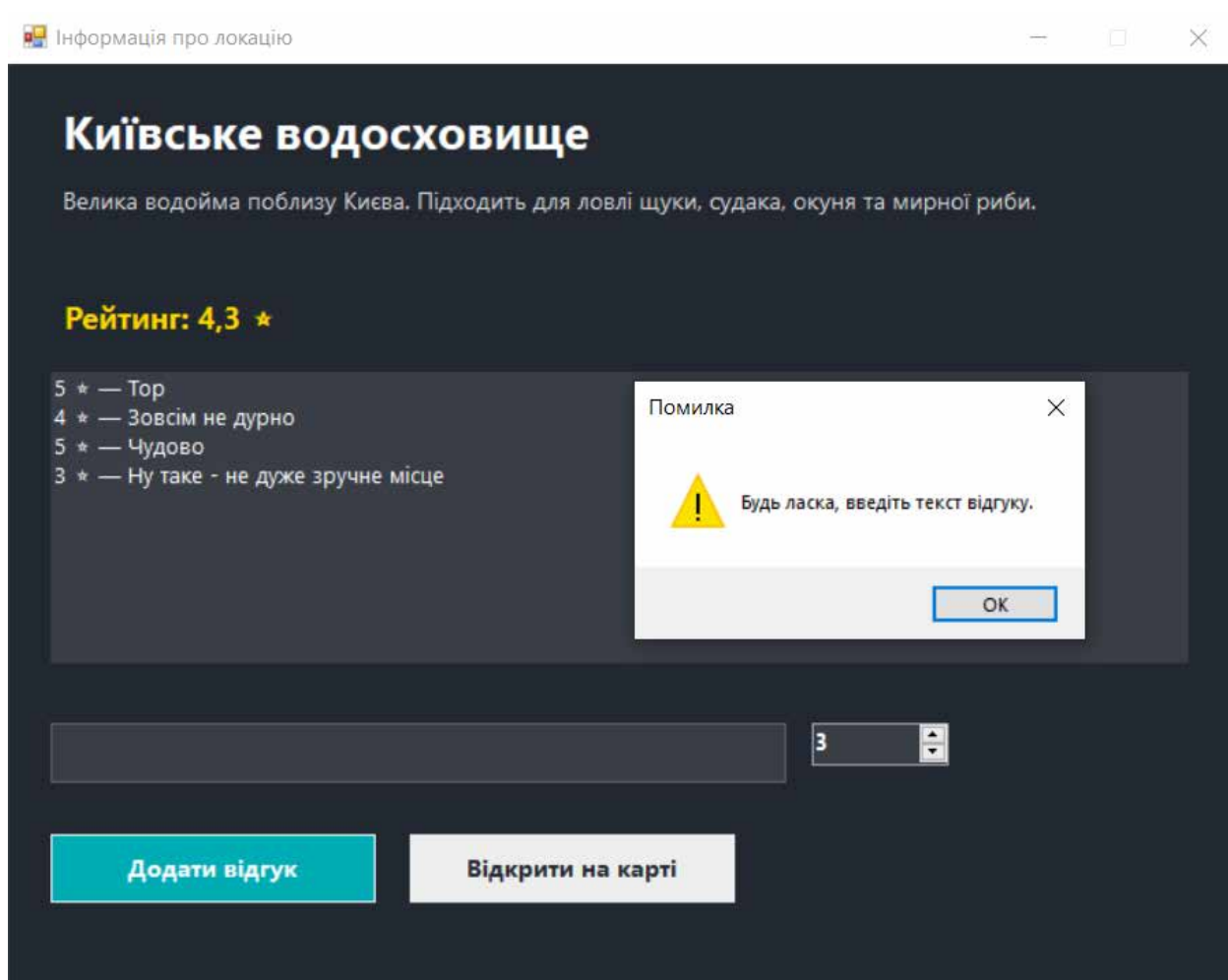


Рис. 4.10 Перевірка введення порожнього відгуку

Після додавання кількох відгуків було перевірено формування середнього рейтингу. Система обчислює середнє значення оцінок для конкретної локації та відображає його на формі детального перегляду. Після повернення до каталогу оновлений рейтинг також відображається у списку локацій, що дозволяє використовувати його для сортування.

Окремо було протестовано функцію відкриття локації у Google Maps. Після натискання кнопки «Відкрити на карті» програма формує пошукове посилання за назвою вибраної локації та відкриває його у веббраузері. У результаті користувач може переглянути місцезнаходження водойми на карті.

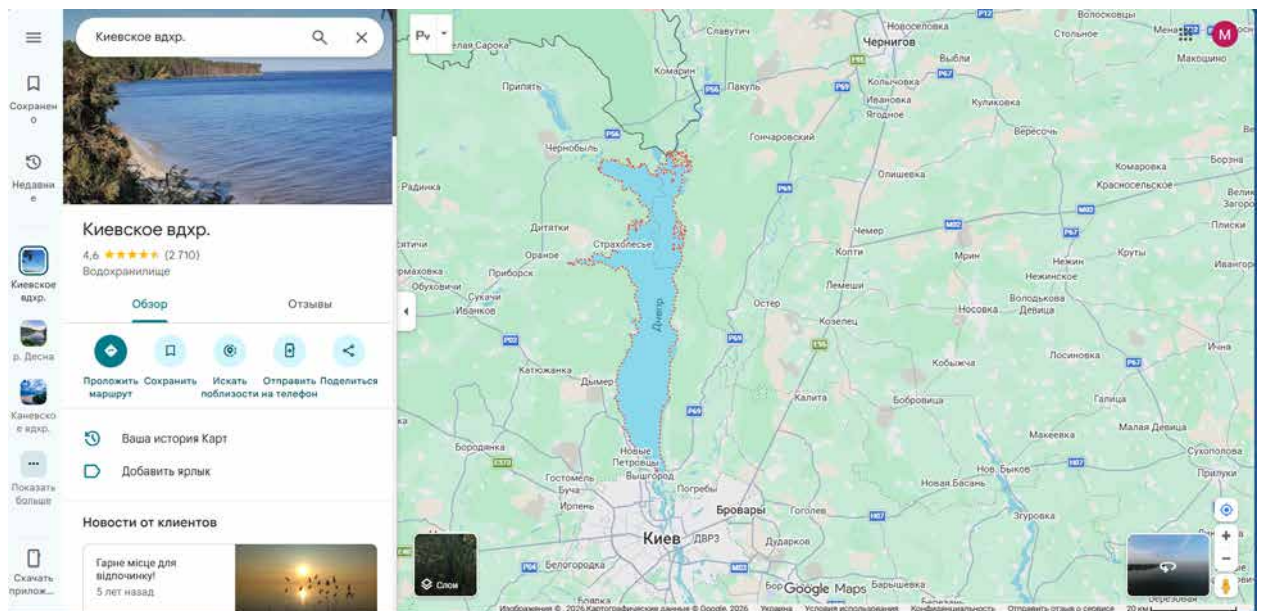


Рис. 4.11 Відкриття риболовної локації у Google Maps

Також було перевірено сценарій, коли користувач натискає кнопку відкриття локації, не вибравши жодного запису у списку. У цьому випадку система виводить повідомлення про необхідність вибрати локацію. Це запобігає виникненню помилки під час відкриття форми детальної інформації.

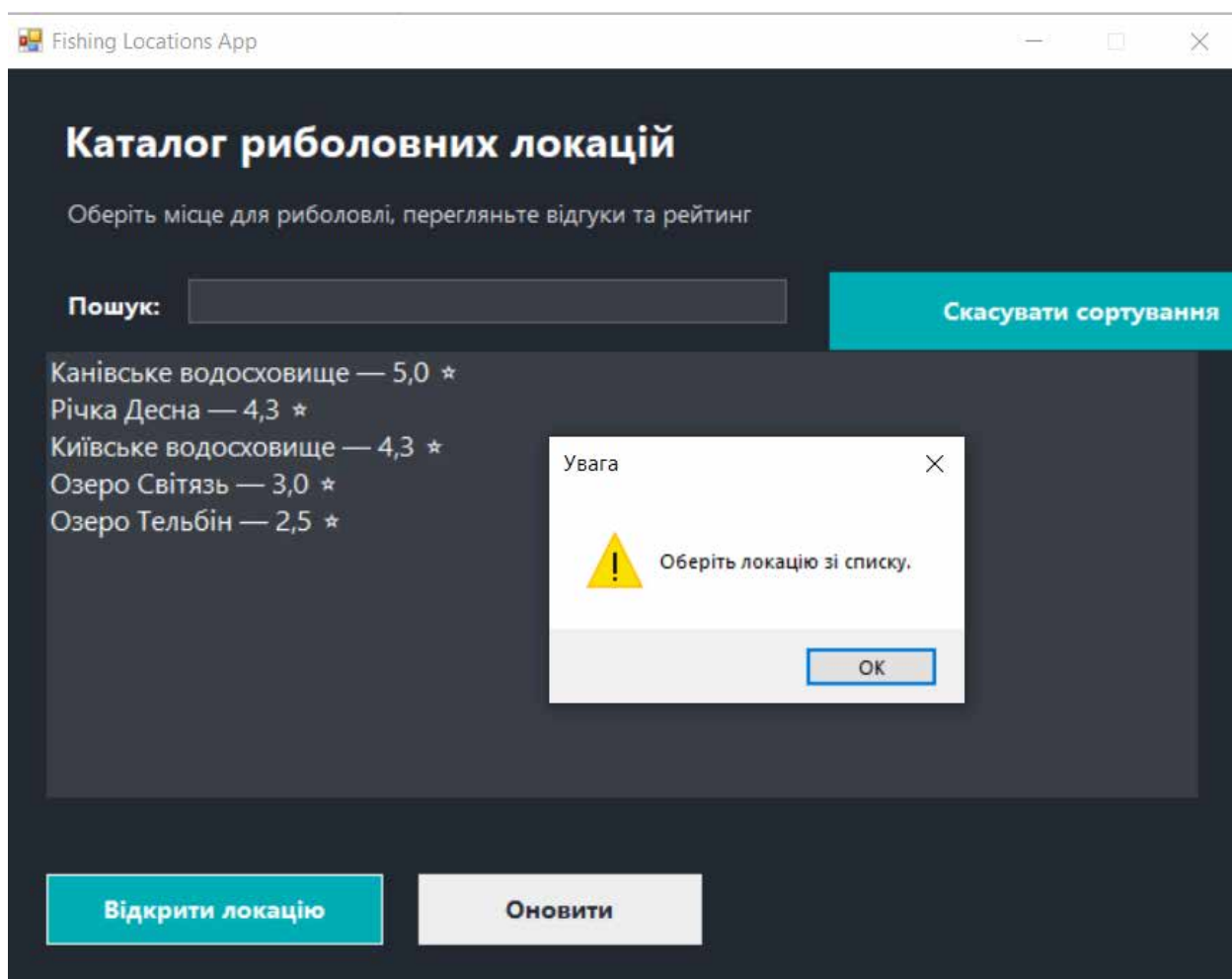


Рис. 4.12 Повідомлення про необхідність вибору локації

Для перевірки авторизації було виконано негативний сценарій із введенням неправильного пароля. Користувач ввів існуючу електронну пошту, але неправильний пароль. Система не виконала вхід і повідомила про помилку авторизації. Це підтверджує, що доступ до основного функціоналу надається лише після введення коректних облікових даних.

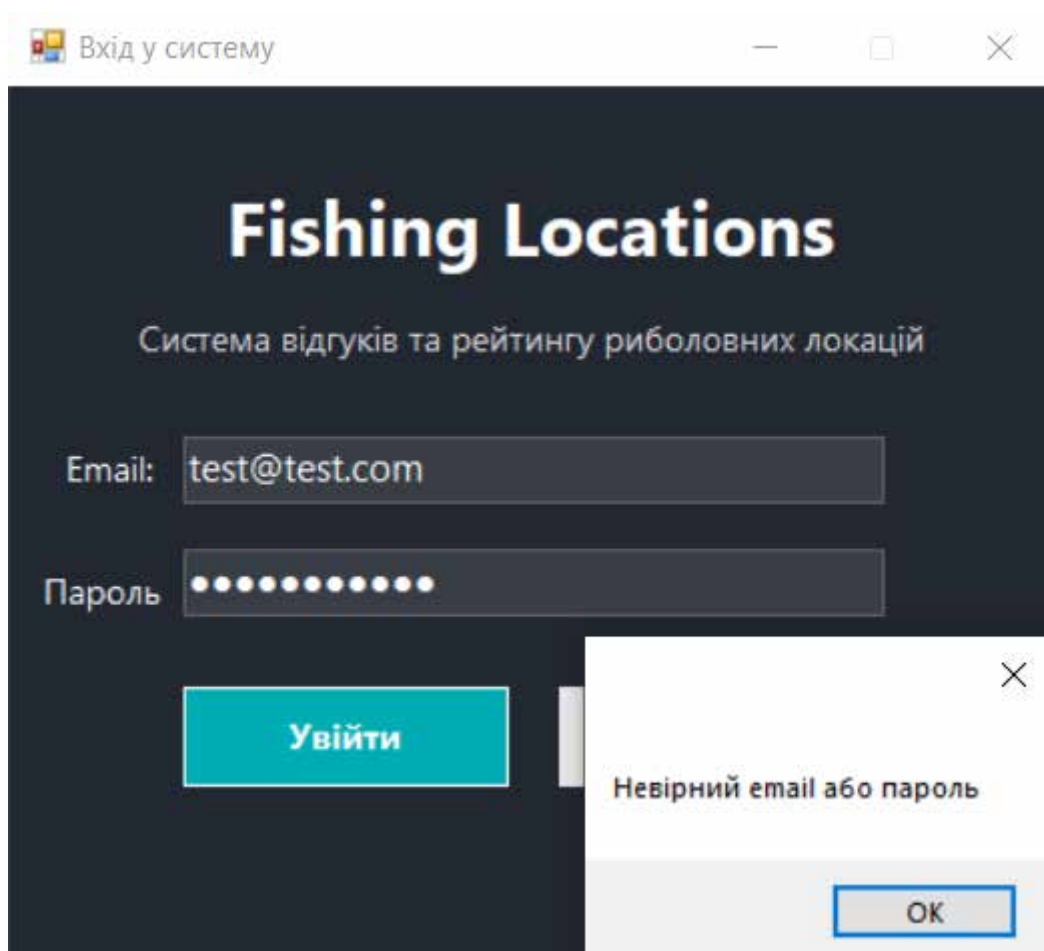


Рис. 4.13 Помилка авторизації при неправильному паролі

Окремо було перевірено роботу функції відкриття риболовної локації у Google Maps за відсутності підключення до Інтернету. Для цього користувач відкрив форму детальної інформації про риболовну локацію та натиснув кнопку «Відкрити на карті» без активного мережевого з'єднання.

У результаті настільний застосунок продовжив працювати у штатному режимі, однак веббраузер не зміг завантажити сторінку Google Maps або відобразив повідомлення про відсутність підключення. Це підтверджує, що основні функції системи, зокрема перегляд каталогу, відгуків і рейтингу, не залежать від Інтернету. Мережеве з'єднання необхідне лише для роботи зовнішнього картографічного сервісу.



Рис 4.14 Перевірка відкриття Google Maps без підключення до Інтернету

Окремо було перевірено сценарій повторної реєстрації користувача з електронною поштою, яка вже існує у базі даних. Для цього було виконано спробу створення нового облікового запису з email, який уже використовувався раніше.

У результаті система не створила дубльований обліковий запис і вивела повідомлення про те, що користувач із таким email уже існує. Це підтверджує коректність перевірки унікальності електронної пошти під час реєстрації та запобігає виникненню дубльованих записів у таблиці Users.

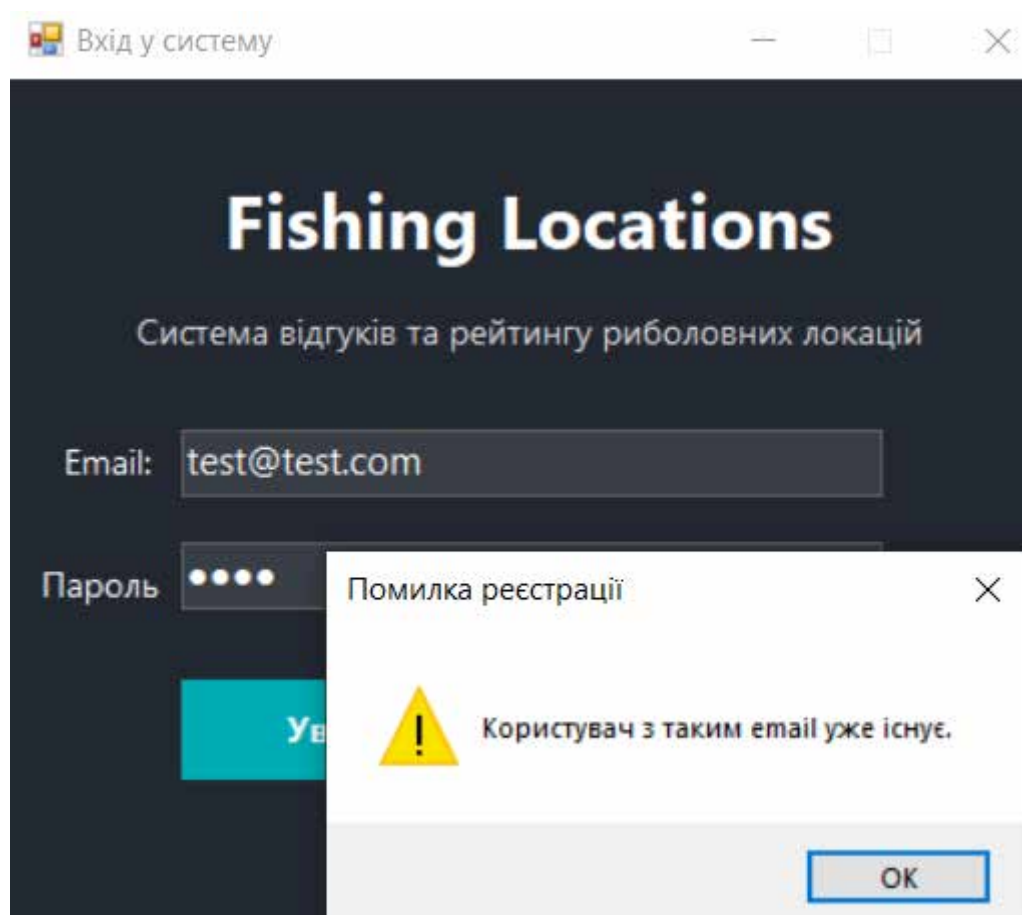


Рис 4.14 Перевірка повторної реєстрації з існуючим email

## Таблиця позитивних сценаріїв

Таблиця 4.3 – Результати тестування основних функцій системи

| №  | Тестова дія                        | Очікуваний результат                          | Фактичний результат                     | Статус  |
|----|------------------------------------|-----------------------------------------------|-----------------------------------------|---------|
| 1  | Введення коректних даних для входу | Система виконує авторизацію користувача       | Відкрито головне вікно програми         | Успішно |
| 2  | Реєстрація нового користувача      | Створюється новий обліковий запис             | Користувача зареєстровано               | Успішно |
| 3  | Відкриття головного вікна          | Відображається каталог риболовних локацій     | Каталог локацій завантажено автоматично | Успішно |
| 4  | Пошук локації за назвою            | У списку залишаються відповідні записи        | Список відфільтровано за запитом        | Успішно |
| 5  | Сортування за рейтингом            | Локації впорядковуються за середнім рейтингом | Список відсортовано за рейтингом        | Успішно |
| 6  | Вибір локації зі списку            | Відкривається форма детальної інформації      | Форма локації відкривається             | Успішно |
| 7  | Перегляд відгуків                  | Система відображає коментарі до локації       | Відгуки відображаються на формі         | Успішно |
| 8  | Додавання текстового відгуку       | Відгук зберігається у базі даних              | Відгук додано до списку                 | Успішно |
| 9  | Виставлення оцінки                 | Рейтинг локації перераховується               | Середній рейтинг оновлено               | Успішно |
| 10 | Відкриття Google Maps              | У браузері відкривається карта                | Google Maps відкрито                    | Успішно |
| 11 | Оновлення списку локацій           | Каталог повторно завантажується               | Список оновлено                         | Успішно |

## Таблиця негативних сценаріїв

Таблиця 4.4 – Результати перевірки негативних сценаріїв

| № | Негативний сценарій                                | Очікуваний результат                                                                                                       | Фактичний результат                                                                         | Статус  |
|---|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------|
| 1 | Введення неправильного пароля                      | Система не виконує вхід                                                                                                    | Виведено повідомлення про помилку авторизації                                               | Успішно |
| 2 | Порожні поля авторизації                           | Система просить заповнити email і пароль                                                                                   | Виведено попередження                                                                       | Успішно |
| 3 | Натискання «Відкрити локацію» без вибору           | Система просить обрати локацію                                                                                             | Виведено повідомлення користувачу                                                           | Успішно |
| 4 | Додавання порожнього відгуку                       | Порожній відгук не зберігається                                                                                            | Виведено повідомлення про необхідність введення тексту                                      | Успішно |
| 5 | Пошук неіснуючої локації                           | Система повідомляє про відсутність результатів                                                                             | Виведено «Локації не знайдено»                                                              | Успішно |
| 6 | Відкриття Google Maps без підключення до Інтернету | Основні функції програми продовжують працювати, карта не завантажується або браузер повідомляє про відсутність підключення | Застосунок не завершує роботу аварійно, браузер не завантажує Google Maps без Інтернету     | Успішно |
| 7 | Реєстрація користувача з email, який уже існує     | Система не створює дубльований обліковий запис і повідомляє про помилку                                                    | Виведено повідомлення «Користувач з таким email уже існує», повторна реєстрація не виконана | Успішно |

Результати тестування підтвердили працездатність основних функцій системи та відповідність розробленого програмного забезпечення поставленим вимогам. Застосунок коректно виконує авторизацію користувача, завантажує каталог риболовних локацій, підтримує пошук і сортування за рейтингом, дозволяє переглядати детальну інформацію про локацію, додавати відгуки, формувати середній рейтинг і відкривати локацію у Google Maps.

## ВИСНОВКИ

У бакалаврській кваліфікаційній роботі було розроблено програмне забезпечення для збору відгуків та формування рейтингу риболовних локацій. Результатом виконання роботи є настільний застосунок для Windows, створений мовою програмування C# із використанням технології Windows Forms та локальної бази даних SQLite. Розроблена система дозволяє користувачу переглядати каталог риболовних місць, здійснювати пошук за назвою або описом, сортувати локації за рейтингом, відкривати детальну інформацію про водойму, залишати відгуки, виставляти оцінки та переходити до розташування локації через Google Maps.

Поставлену мету роботи було досягнуто. У процесі розробки створено програмний продукт, який забезпечує збір користувацьких відгуків і формування рейтингу риболовних локацій на основі оцінок. Відповідно до визначених завдань було виконано аналіз предметної області, розглянуто існуючі програмні аналоги, сформульовано функціональні та нефункціональні вимоги, побудовано UML-моделі, спроектовано логічну модель даних, реалізовано програмну систему та проведено її тестування.

У ході аналізу аналогів було розглянуто такі рішення, як Fishbrain, FishAngler та Navionics Boating. Вони мають широкий функціонал, однак часто орієнтовані на мобільні платформи, вимагають підключення до онлайн-сервісів або містять надлишкові можливості для простого локального використання. Розроблена система відрізняється від цих аналогів тим, що реалізована як легкий настільний застосунок із локальним збереженням даних у SQLite. Це робить її простішою для встановлення, автономною в основних функціях і придатною для використання без складного серверного налаштування.

Отримані результати відповідають поставленому завданню. У системі реалізовано реєстрацію та авторизацію користувача, автоматичне

завантаження каталогу риболовних локацій, перегляд опису водойм, додавання відгуків, виставлення оцінок, обчислення середнього рейтингу, пошук локацій, сортування за рейтингом і відкриття локації у Google Maps. Додатково було реалізовано обробку типових помилкових ситуацій, зокрема неправильного пароля, порожнього коментаря, відсутності вибраної локації та пошуку без результатів.

Нове технічне рішення полягає у створенні локальної desktop-системи для збору відгуків і формування рейтингу риболовних локацій, яка поєднує графічний інтерфейс Windows Forms, локальну базу даних SQLite, механізм оцінювання за шкалою від 1 до 5, пошук і сортування локацій за середнім рейтингом, а також перехід до Google Maps через сформоване посилання. Такий підхід дозволяє реалізувати основні функції інформаційної системи без використання вебсервера або зовнішньої серверної бази даних.

Особистий внесок студента полягає у проєктуванні та реалізації структури настільного застосунку, створенні логічної моделі бази даних SQLite, розробці класів для роботи з користувачами, локаціями та відгуками, реалізації механізму обчислення середнього рейтингу, додаванні пошуку та сортування локацій, а також реалізації переходу до Google Maps для перегляду місцезнаходження водойми. Також було виконано моделювання системи за допомогою UML-діаграм, підготовлено інсталяційну структуру застосунку та проведено тестування основних і негативних сценаріїв роботи програми.

У третьому розділі було обґрунтовано структуру даних системи. Логічна модель охоплює користувачів, риболовні локації, відгуки та логічне представлення рейтингу. Було показано відповідність моделі реляційній структурі та нормальним формам. Рейтинг локації формується як середнє значення оцінок користувачів, що дозволяє уникнути дублювання даних і забезпечити актуальність рейтингової інформації.

Під час тестування було підтверджено працездатність основних функцій системи. Застосунок коректно виконує авторизацію, завантажує каталог локацій, здійснює пошук, сортує записи за рейтингом, відкриває детальну

інформацію про вибрану локацію, зберігає відгуки та оновлює середній рейтинг. Негативне тестування показало, що система обробляє типові помилкові дії користувача та не завершує роботу аварійно в основних перевірених сценаріях.

Також було перевірено обробку повторної реєстрації з уже існуючим email, що підтвердило коректність контролю дублювання облікових записів користувачів.

Практична цінність розробленої системи полягає у можливості використання її як локального інструменту для накопичення інформації про риболовні місця та оцінювання їх на основі досвіду користувачів. Програмний продукт може бути корисним для рибалок, які хочуть зберігати інформацію про водойми, переглядати відгуки та швидко визначати найбільш привабливі локації за рейтингом.

Подальший розвиток системи може передбачати розширення каталогу локацій, додавання фотографій водойм, фільтрацію за регіоном або типом риби, а також створення розширеної статистики популярності риболовних місць. Отриманий результат може бути використаний як основа для подальшого вдосконалення інформаційної системи оцінювання риболовних локацій.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. C# language documentation [Електронний ресурс]. Microsoft Learn. — Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 22.05.2026).
2. .NET documentation [Електронний ресурс]. Microsoft Learn. — Режим доступу: <https://learn.microsoft.com/en-us/dotnet/> (дата звернення: 22.05.2026).
3. Windows Forms for .NET documentation [Електронний ресурс]. Microsoft Learn. — Режим доступу: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/> (дата звернення: 22.05.2026).
4. Visual Studio documentation [Електронний ресурс]. Microsoft Learn. — Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/windows/> (дата звернення: 22.05.2026).
5. System.Data.SQLite package [Електронний ресурс]. NuGet Gallery. — Режим доступу: <https://www.nuget.org/packages/System.Data.SQLite/> (дата звернення: 22.05.2026).
6. SQLite Documentation [Електронний ресурс]. SQLite. — Режим доступу: <https://www.sqlite.org/docs.html> (дата звернення: 22.05.2026).
7. SQLite SQL Language Reference [Електронний ресурс]. SQLite. — Режим доступу: <https://www.sqlite.org/lang.html> (дата звернення: 22.05.2026).
8. SQLite Datatypes Documentation [Електронний ресурс]. SQLite. — Режим доступу: <https://www.sqlite.org/datatype3.html> (дата звернення: 22.05.2026).
9. Google Maps URLs [Електронний ресурс]. Google for Developers. — Режим доступу: <https://developers.google.com/maps/documentation/urls/get-started> (дата звернення: 22.05.2026).
10. Fishbrain: Fishing App and Fishing Tools [Електронний ресурс]. Fishbrain. — Режим доступу: [Fishbrain - Fishing App – Додатки в Google Play](#) (дата звернення: 22.05.2026).
11. FishAngler: Fishing App [Електронний ресурс]. FishAngler. — Режим доступу: <https://www.fishangler.com/> (дата звернення: 22.05.2026).
12. Navionics Boating App [Електронний ресурс]. Garmin. — Режим доступу: <https://www.garmin.com/en-US/garmin-technology/marine-technology/charts-and-maps/navionics-boating-app/> (дата звернення: 22.05.2026).
13. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. — 2nd ed. — Boston : Addison-Wesley, 2005. — 496 p.

14. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. — 3rd ed. — Boston : Addison-Wesley, 2003. — 208 p.
15. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3rd ed. — Upper Saddle River : Prentice Hall, 2004. — 736 p.
16. Sommerville I. Software Engineering. — 10th ed. — Boston : Pearson, 2015. — 816 p.
17. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. — 8th ed. — New York : McGraw-Hill Education, 2014. — 976 p.
18. Date C. J. An Introduction to Database Systems. — 8th ed. — Boston : Pearson, 2003. — 1024 p.
19. Elmasri R., Navathe S. B. Fundamentals of Database Systems. — 7th ed. — Boston : Pearson, 2016. — 1280 p.
20. ISO/IEC/IEEE 29148:2018. Systems and software engineering — Life cycle processes — Requirements engineering [Электронный ресурс]. — Режим доступа: <https://www.iso.org/standard/72089.html> (дата обращения: 22.05.2026).
21. Nielsen J. Usability Engineering. — San Francisco : Morgan Kaufmann, 1994. — 362 p.
22. Norman D. The Design of Everyday Things. — Revised and expanded edition. — New York : Basic Books, 2013. — 368 p.
23. Shneiderman B., Plaisant C., Cohen M., Jacobs S., Elmqvist N., Diakopoulos N. Designing the User Interface: Strategies for Effective Human-Computer Interaction. — 6th ed. — Boston : Pearson, 2016. — 624 p.
24. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. — Boston : Addison-Wesley, 1994. — 395 p.

**ДОДАТОК А****Фрагменти програмного коду. Підключення до бази даних SQLite**

```
using System.Data.SQLite;

namespace FishingLocationsApp.Data
{
    public class Database
    {
        private string connectionString =
            "Data Source=fishing.db;Version=3;Journal Mode=WAL;";

        public SQLiteConnection GetConnection()
        {
            return new SQLiteConnection(connectionString);
        }
    }
}
```

## ДОДАТОК Б

**Фрагменти програмного коду. Реєстрація та авторизація користувача**

```
using System.Data.SQLite;

namespace FishingLocationsApp.Services
{
    public class AuthService
    {
        public bool Login(string email, string password)
        {
            using (var conn = new Data.Database().GetConnection())
            {
                conn.Open();

                using (var cmd = new SQLiteCommand(
                    "SELECT * FROM Users WHERE Email=@e AND
Password=@p", conn))
                {
                    cmd.Parameters.AddWithValue("@e", email);
                    cmd.Parameters.AddWithValue("@p", password);

                    using (var reader = cmd.ExecuteReader())
                    {
                        return reader.HasRows;
                    }
                }
            }
        }

        public void Register(string username, string email, string password)
        {
            using (var conn = new Data.Database().GetConnection())
            {
                conn.Open();

                using (var cmd = new SQLiteCommand(
                    "INSERT INTO Users (Username, Email, Password) VALUES
(@u,@e,@p)", conn))
                {
                    cmd.Parameters.AddWithValue("@u", username);
```

## ДОДАТОК В

### Фрагменти програмного коду. Відкриття локації у Google Maps

```
private void btnOpenMap_Click(object sender, EventArgs e)
{
    string url =
$"https://www.google.com/maps/search/?api=1&query={Uri.EscapeDataString(cu
rrentLocation.Name)}";

    System.Diagnostics.Process.Start(new System.Diagnostics.ProcessStartInfo
    {
        FileName = url,
        UseShellExecute = true
    });
}
```

**ДОДАТОК Г**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**  
**Факультет інформаційних технологій Декларація про академічну  
добročесність та використання ШІ**

Я, Бикодір Максим Ярославович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Розробка системи збору відгуків та формування рейтингу риболовних локацій» є результатом моєї особистої праці.

2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.

3. Щодо використання інструментів ШІ зазначаю, що ШІ ChatGPT, Gemini, DeepL були використані для:

- перекладу іншомовних джерел;
- стилістичного редагування та перевірки граматики;
- допомоги у написанні та відлагодженні програмного коду;
- пошуку додаткових інструментів у розробці програмного продукту.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«22» Травня 2026 р. \_\_\_\_\_

Максим Бикодір