

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

(повна назва)
Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Програмна система для організації та управління груповими подорожами.
Мобільний застосунок
(тема)

Виконав:
здобувач 4 року навчання, групи ПЗП-21-3
Артем ФОМИЧОВ
(власне ім'я, прізвище)

Спеціальність 121 – Інженерія програмного
забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-професійна
Освітня програма Програмна інженерія
(повна назва освітньої програми)

Керівник доц. Дмитро КОЛЕСНИКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри

Кирило СМЕЛЯКОВ
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____

Кафедра _____ програмної інженерії _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 121 – Інженерія програмного забезпечення _____
(код і повна назва)

Тип програми _____ Освітньо-професійна _____

Освітня програма _____ Програмна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**здобувачеві _____ Фомичову Артему Сергійовичу _____
(прізвище, ім'я, по батькові)1. Тема роботи _____ Програмна система для організації та управління
груповими подорожами. Мобільний застосунок _____

Затверджена наказом університету від _____ 19.05. 2025р. № 397Ст _____

2. Термін подання здобувачем роботи до екзаменаційної комісії 10.06.2025 р.3. Вихідні дані до роботи Розробити застосунок для організації та управління
груповими подорожами за допомогою .NET MAUI.4. Перелік питань, що потрібно опрацювати у роботі
Вступ, аналіз предметної галузі, формування вимог до програмної системи,
архітектура та проектування програмного забезпечення, опис прийнятих
програмних рішень, тестування розробленого програмного забезпечення,
висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	10.04.2025	<i>виконано</i>
2	Формування вимог до ПЗ	15.04.2025	<i>виконано</i>
3	Проектування ПЗ	20.04.2025	<i>виконано</i>
4	Розробка ПЗ	20.05.2025	<i>виконано</i>
5	Тестування ПЗ	25.05.2025	<i>виконано</i>
6	Оформлення пояснювальної записки	02.06.2025	<i>виконано</i>
7	Перевірка роботи на антиплагіат та проходження нормоконтролю	03.06.2025	<i>виконано</i>
8	Оцінка роботи рецензентом, отримання відгуку від керівника кваліфікаційної роботи	08.06.2025	<i>виконано</i>
9	Попередній захист роботи	09.06.2025	<i>виконано</i>
10	Здача роботи до електронного архіву, допуск роботи до захисту завідувачем кафедри	10.06.2025	<i>виконано</i>
11	Захист кваліфікаційної роботи	13.06.2025	

Дата видачі завдання 07 квітня 2025 р.

Здобувач

_____ (підпис)

Артем ФОМИЧОВ

Керівник роботи

_____ (підпис)

доц. Дмитро КОЛЕСНИКОВ

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра, 89 стор., 32 рис., 4 табл., 18 джерел.

БИОМЕТРИЧНІ ДАНІ, БЕЗПЕКА, МОБІЛЬНИЙ ЗАСТОСУНОК, МОНИТОРИНГ, РЕАЛЬНИЙ ЧАС, СИСТЕМИ ГРУПОВИХ ПОДОРОЖЕЙ, ТРЕКІНГОВІ ДАНІ, ШИФРУВАННЯ, API, .NET MAUI, GPS-ТРЕКІНГ, SQLITE, XAML.

Об'єкт розробки – програмна система для організації та управління груповими подорожами.

Мета розробки – створення мобільного застосунку для ефективного планування, координації та управління груповими подорожами з використанням фреймворку .NET MAUI. Застосунок має надавати користувачам зручний, зрозумілий та доступний на різних платформах інструмент для колективного планування маршрутів, ведення обліку учасників і вирішення організаційних питань.

Метод рішення – середовище розробки Visual Studio, мова програмування C#, фреймворк .NET MAUI. Також використовуються патерн MVVM, SQLite для локального збереження даних і HTTP-запити для взаємодії з серверною частиною.

У результаті розробки було створено кросплатформний мобільний застосунок для організації та керування груповими подорожами, який надає користувачам інструменти для планування маршрутів, взаємодії між учасниками та обміну необхідною інформацією під час подорожі.

ABSTRACT

BIOMETRIC DATA, SECURITY, MOBILE APPLICATION, MONITORING, REAL-TIME, GROUP TRAVEL SYSTEMS, TRACKING DATA, ENCRYPTION, API, .NET MAUI, GPS TRACKING, SQLITE, XAML.

Object of development is a software system for organizing and managing group travel.

Purpose of development – to create a mobile application for effective planning, coordination, and management of group travel using the .NET MAUI framework. The application aims to provide users with a convenient, intuitive, and cross-platform tool for collaborative route planning, participant tracking, and handling organizational matters.

Solution method – the development environment is Visual Studio, the programming language is C#, and the framework is .NET MAUI. The MVVM pattern is applied, SQLite is used for local data storage, and HTTP requests are used for communication with the server-side.

As a result of the development, a cross-platform mobile application for organizing and managing group travel was created, providing users with tools for route planning, participant interaction, and sharing essential information during the trip.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	9
1.1 Загальний огляд предметної області.....	9
1.2 Виявлення та вирішення проблем.....	12
1.3 Постановка задачі.....	13
2 Формування вимог до програмної системи.....	14
2.1 Технологія та інструменти.....	14
2.2 Архітектура програми.....	15
2.3 Дизайн застосунку.....	16
2.4 Взаємодія з серверною частиною.....	17
3 Архітектура та проектування	19
3.1 UML проектування	19
3.2 Проектування архітектури	22
3.3 Проектування структури зберігання, управління даними.....	23
4 Опис прийнятих програмних рішень.....	27
4.1 Опис структури проєкту.....	27
4.2 Використані бібліотеки.....	29
4.3 Авторизація у програмному застосунку.....	30
4.4 Подорожі у програмному застосунку.....	31
4.5 Моніторинг та надзвичайні ситуації у програмному застосунку.....	32
4.6 Профіль користувача.....	34
5 Тестування програмного забезпечення.....	36
5.1 Тестування програмного застосунку.....	36
Висновки.....	42
Перелік джерел посилання.....	44
Додаток А Структура проєкту.....	46
Додаток Б Фрагменти коду.....	49
Додаток В Звіт з результатами перевірки на унікальність тексту в базу ХНУРЕ.....	52
Додаток Г Специфікація програмної системи.....	53
Додаток Д Слайди презентації.....	81

ВСТУП

У сучасному світі мобільні технології займають надзвичайно важливе місце в повсякденному житті людини, які забезпечують швидкий доступ до інформації, сервісів і засобів комунікації. Завдяки стрімкому розвитку інтернету та зростанню популярності мобільних пристроїв, мобільні застосунки стали незамінним інструментом у багатьох сферах людської діяльності, зокрема в освіті, медицині, бізнесі, логістиці та, звісно ж, у туризмі. Особливо актуальним є впровадження мобільних технологій у сферу організації подорожей, де ефективне планування, координація дій та обмін даними між учасниками мають вирішальне значення для успішного проведення заходів.

Організація групових подорожей зазвичай потребує значних зусиль — від розробки маршруту, бронювання транспорту та проживання до розподілу бюджету, координації дій учасників та реагування на непередбачувані обставини у реальному часі. Традиційні інструменти, такі як електронні таблиці, месенджери або неструктуровані списки, хоч і використовуються досить часто, не завжди здатні забезпечити необхідну гнучкість, централізований доступ до даних та швидке оновлення інформації. У таких умовах виникає нагальна потреба у створенні спеціалізованого цифрового рішення — мобільного застосунку, який забезпечить повний цикл управління подорожами в одному інтерфейсі.

У межах даної дипломної роботи було розроблено мобільний застосунок TravelMate, призначений для організації, планування та супроводу групових подорожей. Його головна мета — спростити координацію учасників, полегшити планування маршрутів, покращити комунікацію в межах групи а також аналізувати стан здоров'я учасників під час подорожі та сигналізувати у разі виникнення небезпечної ситуації. Створення подібного рішення дозволяє не лише автоматизувати процеси, які раніше потребували ручної обробки, а й підвищити загальну якість та безпеку подорожей.

Застосунок реалізовано за допомогою сучасного кросплатформного фреймворку .NET MAUI, що дозволяє створювати універсальні мобільні рішення, які працюють на різних операційних системах, включаючи Android, iOS та Windows. Такий підхід забезпечує зниження витрат на розробку, спрощує технічну підтримку продукту та розширює його потенційну аудиторію.

Основними функціональними можливостями застосунку є:

1. створення та редагування подорожей;
2. планування маршруту з інтеграцією Google Maps;
3. корегування учасників подорожі;
4. надсилання запитів на участь;
5. взаємодія користувачів через вбудовану систему повідомлень;
6. підтримка профілю користувача з можливістю редагування;
7. локальне збереження даних із синхронізацією через Web API;
8. використання Bluetooth для інтеграції з пристроями фітнес-трекінгу.

Дипломна робота охоплює повний цикл створення програмного забезпечення — від аналізу предметної області, формулювання вимог і проектування архітектури до реалізації, тестування та документування застосунку. Результатом стало функціональне та інтуїтивно зрозуміле рішення, орієнтоване на реальні потреби мандрівників, що дозволяє покращити процес організації групових подорожей, зробити його більш зручним, ефективним та доступним для широкого кола користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальний огляд предметної області

Сучасне суспільство зацікавлене у активному відпочинку, зокрема у формі одноденних подорожей, екскурсій, пішохідних прогулянок у горах, лісах та кемпінгу. Такі подорожі дають змогу швидко змінити оточення, відновити емоційний стан та зміцнити соціальні зв'язки в колі друзів, родини або колег. Особливістю цього формату є короткотривалість, мінімальні ресурси, гнучкий маршрут та акцент на взаємодію між учасниками.

Згідно з дослідженням, зростання популярності екотуризму прямо пропорційне зменшенню природних та традиційних культурних ландшафтів [1], що свідчить про підвищений інтерес до активного відпочинку на природі. Такий формат подорожей приваблює людей можливістю втекти від урбанізованого середовища, відновити емоційний стан та отримати нові враження в умовах живої природи. Водночас організація подібних подій, особливо у форматі короткотривалих виїздів — одноденних походів, екскурсій, пікніків чи кемпінгів, — зазвичай здійснюється вручну через месенджери, соціальні мережі або табличні редактори, що не забезпечує належного рівня координації, структурованості та зручності для всіх учасників.

У процесі підготовки часто виникають типові труднощі: хтось забуває місце збору або час виїзду, хтось не має доступу до актуальної версії маршруту, неясно, хто відповідає за спорядження, харчування чи транспорт, а важливі оновлення або зміни плану не доходять до всіх учасників вчасно. Крім того, складність зростає в умовах відсутності мобільного покриття або інтернету, що часто трапляється у віддалених природних локаціях.

Ці фактори підкреслюють актуальність створення спеціалізованого мобільного застосунку, який дозволить централізовано керувати усіма аспектами організації коротких групових подорожей. Такий інструмент має не лише спростити процес підготовки, але й підвищити ефективність комунікації,

забезпечити прозорий розподіл обов'язків і мінімізувати ризик непорозумінь у критичних ситуаціях.

Наразі на ринку практично відсутні мобільні рішення, орієнтовані саме на організацію короткотривалих групових виїздів, які б одночасно об'єднували:

1. планування маршруту з можливістю вибору точок на карті та додавання описів;
2. формування списку учасників з можливістю надсилання запрошень та відстеження статусу участі;
3. збереження ключової інформації: контактів, GPS-координат, нагадувань, чек-листів тощо;
4. офлайн-режим, що дозволяє зберігати доступ до даних навіть за відсутності інтернету;
5. інтегровану систему обміну повідомленнями, яка дозволяє надсилати оперативні зміни маршруту чи повідомлення групі;
6. редагування профілю, включаючи зміну імені, фото, базової інформації;
7. відстеження стану здоров'я користувачів через інтеграцію з фітнес-браслетами для зчитування пульсу.

На ринку групових подорожей помітно вирізняються два застосунки — *Group* і *Wanderlog*.

На ринку групових подорожей помітно вирізняються два застосунки — *Group* і *Wanderlog*, які пропонують корисний функціонал для планування поїздок, проте мають обмеження, що знижують їхню ефективність у контексті короткочасних виїздів на природу чи активного відпочинку без доступу до мережі.

Group, розроблений *JetBlue Travel Products*, фокусується на колективному ухваленні рішень. Основний функціонал охоплює створення опитувань, спільне голосування за дати, маршрути або місця, ведення централізованого чату та можливість підключення до сервісів бронювання житла або квитків. Застосунок

зручний для невеликих компаній друзів або родичів, які планують подорож у межах міської інфраструктури або популярних туристичних напрямків [2]. Втім, Google не адаптований для сценаріїв, де потрібне гнучке управління змінами маршруту або офлайн-доступ.

Wanderlog позиціонується як більш універсальний інструмент для індивідуальних і групових подорожей. Він дозволяє імпортувати бронювання з Gmail, будувати маршрути на основі точок інтересу, планувати бюджет, додавати нотатки та вести спільне редагування у реальному часі. Важливою перевагою є підтримка офлайн-доступу до маршрутів, що робить Wanderlog зручним для мандрівок у місцевості зі слабким інтернетом [3]. Проте застосунок орієнтований переважно на традиційні туристичні подорожі та не містить інструментів для внутрішньогрупової координації в реальному часі, таких як визначення місцезнаходження кожного учасника на карті, контроль фізичного стану або обробка екстрених ситуацій.

Головне питання мандрівників сьогодні не в тому, чи подорожувати, а куди саме подорожувати та яким чином. На це питання намагалися відповісти численні сервіси, такі як «Тріп Адвайзер», сайти туристичних агентстві навіть сервіси Google [4].

Розглянуті рішення не передбачають:

1. відстеження GPS-координат учасників у реальному часі, що критично важливо у походах чи кемпінгу;
2. інтеграції з фітнес-пристроями для моніторингу фізичного стану користувача, наприклад, пульсу;
3. екстреної сигналізації або повідомлень SOS, що є важливою функцією безпеки;
4. зручного режиму роботи без доступу до мережі, що дозволяє не тільки переглядати, але й оновлювати маршрут, дані учасників або створювати записи в офлайн-режимі.

Особливої уваги потребує простота та інтуїтивність інтерфейсу, адже учасниками подорожей часто є люди з різним рівнем цифрової підготовки. Також важливо забезпечити кросплатформність, щоб застосунок працював як на Android, так і на iOS-пристроях.

Таким чином, проблемна область охоплює потребу у спеціалізованому мобільному рішенні для організації коротких групових подорожей на природі, яке б поєднувало гнучкість, простоту, автономність і засоби взаємодії між учасниками.

1.2 Виявлення та вирішення проблем

Під час дослідження предметної області та аналізу аналогічних мобільних застосунків було виявлено низку обмежень, які не дозволяють повноцінно задовольнити потреби користувачів, що планують групові подорожі у форматі одноденних екскурсій, пішохідних походів або кемпінгів. Сучасні програмні продукти досить часто потребують підтримки роботи з даними у режимі офлайн. Однією з найважливіших вимог в даному випадку є відсутність конфліктів та блокувань для користувача, або їх мінімізація. [5] Серед ключових проблем:

1. відсутність централізованого інструменту координації учасників;
2. недостатня підтримка офлайн-режиму;
3. неможливість відстеження учасників у реальному часі за допомогою GPS;
4. брак механізмів екстреного сповіщення та реагування;
5. обмежена функціональність для комунікації між учасниками;
6. відсутність засобів призначення ролей та відповідальних осіб у групі.

Ці недоліки знижують ефективність організації подорожей, особливо у випадках, коли важлива синхронізація дій та оперативна взаємодія між учасниками. Для усунення зазначених проблем було запропоновано створити спеціалізований мобільний застосунок на основі фреймворку .NET MAUI [6], який забезпечуватиме:

1. зручне планування маршрутів з колективним внесенням змін;
2. інтеграцію з GPS та картографічними сервісами;
3. підтримку обміну повідомленнями та файлами;
4. локальне збереження важливої інформації з можливістю доступу без Інтернету;
5. систему сповіщень і нагадувань;
6. можливість поділу ролей серед учасників для кращої організації подорожі.

1.3 Постановка задачі

У процесі розробки мобільного застосунку для організації та управління груповими подорожами необхідно реалізувати функціональність, яка забезпечить ефективну взаємодію між учасниками групи, гнучке планування маршруту та підтримку під час поїздки в реальному часі. Відповідно до цього формуються наступні основні задачі:

1. Провести аналіз аналогічних програмних рішень, визначити їхні сильні та слабкі сторони.
2. Розробити структуру мобільного застосунку з урахуванням принципів зручності, доступності та безпеки.
3. Спроектувати базу даних для зберігання інформації про подорожі, учасників, маршрути та інші організаційні дані.
4. Реалізувати кросплатформний мобільний застосунок з використанням фреймворку .NET MAUI.
5. Забезпечити можливість локального збереження даних через SQLite.
6. Реалізувати обмін інформацією між користувачами в межах подорожі.
7. Додати можливість GPS-відстеження маршруту та учасників.
8. Провести тестування розробленої системи та оцінити її функціональність і зручність у користуванні.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

2.1 Технологія та інструменти

Для реалізації мобільного застосунку було обрано .NET MAUI (Multi-platform App UI) — сучасний фреймворк від Microsoft для розробки кросплатформних застосунків. Основними причинами вибору цієї технології є:

1. можливість створення застосунків під Android, iOS, Windows і macOS з єдиною базою коду;
2. підтримка архітектурного патерну MVVM (Model-View-ViewModel) [7], що забезпечує розділення логіки представлення та бізнес-логіки;
3. використання XAML для опису UI, що дозволяє швидко створювати адаптивні інтерфейси;
4. легка інтеграція з SQLite для локального зберігання даних користувача;
5. можливість реалізації HTTP-запитів до серверної частини для обміну даними з віддаленим сервером.

Інструменти, що використовуються для розробки мобільного застосунку:

1. Visual Studio 2022 – інтегроване середовище розробки з повною підтримкою .NET MAUI, C#, XAML та інструментів налагодження.
2. .NET SDK 8.0+ – платформа для створення кросплатформних застосунків із використанням .NET MAUI.
3. .NET MAUI (Multi-platform App UI) – кросплатформний фреймворк для створення мобільних і десктопних застосунків з єдиним кодом інтерфейсу.
4. SQLite-net – ORM-бібліотека для роботи з локальною базою даних SQLite у .NET MAUI-застосунках [8].
5. Plugin.BLE – кросплатформна бібліотека для роботи з Bluetooth-пристроями, зокрема для підключення до фітнес-браслета (наприклад, Mi Band) і зчитування пульсу [9].
6. Shell Navigation – механізм навігації у .NET MAUI, що дозволяє визначати маршрути сторінок і вкладок через AppShell.xaml.
7. Git – система контролю версій для відстеження змін у кодї та колаборації.
8. GitHub – хмарне сховище репозиторію проєкту та засіб командної роботи.

9. XAML – мова розмітки для побудови користувацького інтерфейсу в .NET MAUI.

10. C# – основна мова програмування для бізнес-логіки, роботи з Bluetooth, базою даних і обробки подій.

11. Фізичний Android-пристрій – для тестування працездатності застосунку в реальних умовах, зокрема перевірки підключення по Bluetooth, GPS та офлайн-режиму.

2.2 Архітектура програми

Для проєктування мобільних архітектур під ОС Android протягом останніх років найбільш поширені два підходи – MVP (Model-View-Presenter), та рекомендований компанією Google підхід MVVM (Model-View-ViewModel), який використовує двоспрямований потік даних [10].

Архітектура програмної системи побудована на основі патерну MVVM, який забезпечує чітке розділення відповідальностей між логікою представлення, бізнес-логікою та моделлю даних. MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), відомої як модель (можна також сказати, що це відокремлення представлення від моделі) [11]. Основними компонентами в даному підході є Model, View та ViewModel. Model відповідає за структуру даних і взаємодію з базою даних (SQLite) та зберігає інформацію про подорожі, маршрути, учасників, повідомлення тощо. View – це XAML-файли, що описують інтерфейс користувача, забезпечують інтуїтивну взаємодію з користувачем. ViewModel реалізовує логіку, яка пов'язує модель з уявленням, а також реалізує команди, властивості для прив'язки даних, обробку подій.

У структурі майбутнього застосунку TravelMate планується реалізувати архітектуру, яка базуватиметься на принципі розподілу компонентів Model - Service - ViewModel - View. Моделі (Model) будуть відповідати за опис основних сутностей системи, таких як користувачі, подорожі, точки маршруту, трекінгові дані та інші, забезпечуючи збереження та структурування інформації. Сервіси

(Service) передбачають інкапсуляцію логіки роботи з даними та зовнішніми джерелами, включаючи взаємодію з API, роботу з базою даних SQLite через репозиторії, обробку GPS-даних, керування локальними файлами та сповіщеннями. Вони будуть слугувати посередниками між моделями та ViewModel, реалізуючи бізнес-логіку застосунку. ViewModel планується використовувати для реалізації логіки представлення і взаємодії користувача із застосунком, забезпечуючи двонапрямний зв'язок між View та Model, а також обробку команд користувача, валідацію та підготовку даних для відображення. Візуальна частина (View) буде реалізована за допомогою XAML у .NET MAUI і виконуватиме роль інтерфейсу користувача, відповідального за відображення інформації і прийом введених даних. При цьому View не матиме прямого доступу до моделей чи сервісів, що забезпечить чітке розмежування відповідальностей і сприятиме зручності підтримки та масштабованості майбутнього застосунку.

Завдяки використанню патерну MVVM забезпечується легке оновлення інтерфейсу користувача при зміні даних та дозволяє мінімізувати дублювання коду. Крім того, така архітектура сприяє підвищенню тестованості окремих компонентів, полегшує підтримку та розширення застосунку в майбутньому, адже нові функції можна додавати, не порушуючи існуючий код.

2.3 Дизайн застосунку

Дизайн та інтерфейс є ключовим елементом будь-якого мобільного застосунку, особливо у випадку системи для організації та управління груповими подорожами. Оскільки користувачі застосунку можуть активно користуватись ним у динамічних умовах — у поїздках, під час активного відпочинку або пішого туризму — інтерфейс має бути інтуїтивно зрозумілим, простим та швидким у використанні. Правильно розроблений користувацький інтерфейс може значно підвищити задоволеність користувачів, знизити час на навігацію по

додатку, і таким чином, збільшити загальну ефективність використання програми [12].

У запропонованому мобільному застосунку, розробленому з використанням фреймворку .NET MAUI, особливу увагу приділено принципам мінімалізму та зручності. Основні елементи інтерфейсу створюються за допомогою мови XAML, що дозволяє ефективно розділити логіку представлення від бізнес-логіки завдяки патерну MVVM. Це забезпечує чітку структуру, покращує підтримку коду та спрощує масштабування інтерфейсу.

Для користувачів передбачено чітку ієрархію сторінок, зручну навігацію, великі елементи керування, що зручно використовувати на ходу, а також адаптивний дизайн, що гарантує коректне відображення на різних розмірах екранів. Інтерфейс орієнтований на швидкий доступ до основних функцій — перегляду маршруту, координації з іншими учасниками, перегляду завдань і повідомлень.

Дизайн передбачає використання зручної кольорової палітри, що не перевантажує очі, можливість перемикання між світлою та темною темами, прості та зрозумілі іконки й кнопки для забезпечення швидкого доступу до функцій та візуальні підказки та мінімізацію потреби в інструкціях.

Таким чином, дизайн застосунку спрямований не лише на зручність, але й на створення позитивного емоційного досвіду. Користувачі зможуть з легкістю взаємодіяти з усіма елементами інтерфейсу навіть у польових умовах, що робить застосунок надійним інструментом для організації та проведення групових подорожей.

2.4 Взаємодія з серверною частиною

Взаємодія мобільного застосунку з серверною частиною є критично важливою складовою програмної системи, що забезпечує обмін даними між користувачами та централізоване зберігання інформації. Комунікація між клієнтом і сервером відбувається через REST API з використанням HTTP-запитів

та передачі даних у форматі JSON. Для забезпечення безпеки авторизації та контролю доступу до функцій застосовується механізм аутентифікації на основі JWT (JSON Web Token) [13]. Використання JWT-токенів під час запитів до захищеного серверу дозволяє перевіряти дані користувачів під час кожного запиту. Але використання токенів з обмеженим терміном дії у REST API потребує від користувачів повторного вводу паролів та даних. Впровадження JWT-токенів з обмеженим терміном дії для підвищення захисту інформації дозволить оновлювати токен користувача з кожним запитом до серверу, уникати його копіювання та розповсюдження серед учасників, поліпшувати моделі даних, змінювати підходи до формування аналітики дій користувачів [14].

У системі передбачено механізм створення оголошень про подорожі: користувачі можуть додавати інформацію про маршрут, час, місце збору, умови участі тощо. Інші користувачі можуть переглядати ці оголошення й приєднуватися до обраних подорожей. Для кожної подорожі створюється окремий чат, у якому учасники можуть обговорювати деталі, узгоджувати плани, обмінюватися корисною інформацією чи файлами.

Уся комунікація між клієнтом і сервером відбувається через захищене HTTPS-з'єднання, що гарантує конфіденційність переданих даних. Сервер обробляє запити на створення, редагування, перегляд і приєднання до подорожей, веде облік активностей користувачів і забезпечує збереження повідомлень у чатах. Зі зростанням вимог до інтерактивності традиційна модель запит відповідь (HTTP RequestResponse) часто не відповідає сучасним вимогам до швидкості оновлення даних [15]. Тому було прийнято рішення про впровадження обміну повідомленнями в реальному часі за допомогою SignalR для покращення взаємодії між учасниками та для відправки повідомлення про надзвичайну ситуацію.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ

3.1 UML проектування

UML (Unified Modeling Language) є стандартизованою мовою моделювання, яка широко використовується для візуалізації, специфікації, конструювання та документування компонентів програмних систем. У контексті розробки програмної системи для організації та управління груповими подорожами UML дозволяє чітко структурувати логіку роботи застосунку, взаємодію користувачів із системою, а також внутрішні процеси між її складовими. Застосування UML-діаграм дає змогу ефективно формалізувати вимоги, налагодити комунікацію між членами команди та виявити потенційні проблеми ще до початку кодування. Візуальне представлення системи полегшує розуміння архітектури та функціоналу навіть для осіб без глибоких технічних знань. UML-діаграми істотно сприяють структуризації процесу розробки та покращують якість кінцевого продукту [16].

У процесі проектування було розроблено діаграми, що відображають ключові функціональні сценарії системи, взаємозв'язки між класами, а також логіку дій користувача в рамках одного з базових сценаріїв.

На Use-Case діаграмі представлені варіанти взаємодії користувача з основними функціями, а саме дії з профілем та подорожами (рис. 3.1). Оскільки мобільний застосунок створений лише до користувачів, то існує лише одна роль “Користувач”, проте для даної ролі доступний багатий набір можливостей, котрий включає як і створення подорожей, редагування даних та керування групою так і можливість приєднуватись до існуючих подорожей.

Діаграма активності відображає дії користувача при роботі з подорожами, від входу в систему до участі або створенні подорожі й обговорення її деталей (рис. 3.2). У разі якщо користувач немає аккаунту, то йому пропонується реєстрація. Після успішного входу у нього є вибір, обрати існуючу подорож або запропонувати свою.

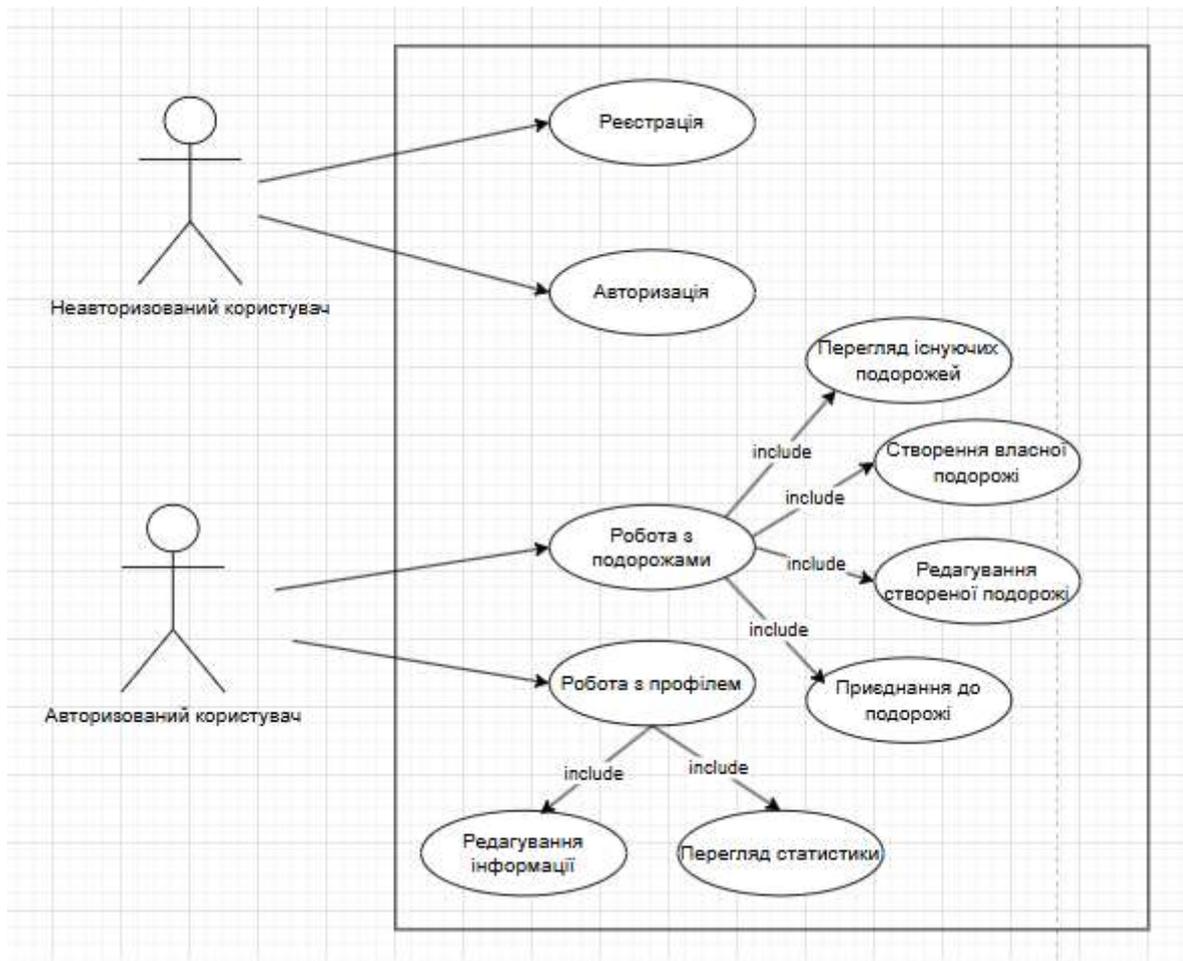


Рисунок 3.1 – Діаграма варіантів використання (Use-Case Diagram)
(рисунок виконано самостійно)

Діаграма послідовності (рис. 3.3) ілюструє взаємодію користувача з мобільним застосунком під час процесу приєднання до групової подорожі. Взаємодія починається з ініціації запиту на вхід у систему. Якщо обліковий запис відсутній, користувач проходить реєстрацію, після чого здійснює авторизацію. Далі користувач надсилає запит на отримання списку доступних подорожей, із якого обирає одну. Система перевіряє доступність місць у групі, після чого користувач подає заявку на участь. У разі підтвердження заявки користувач приєднується до подорожі та отримує доступ до внутрішнього чату, де може обговорювати деталі з іншими учасниками. Кожна з дій супроводжується відповідними запитами до серверної частини через API та отриманням відповідей у вигляді повідомлень або даних.

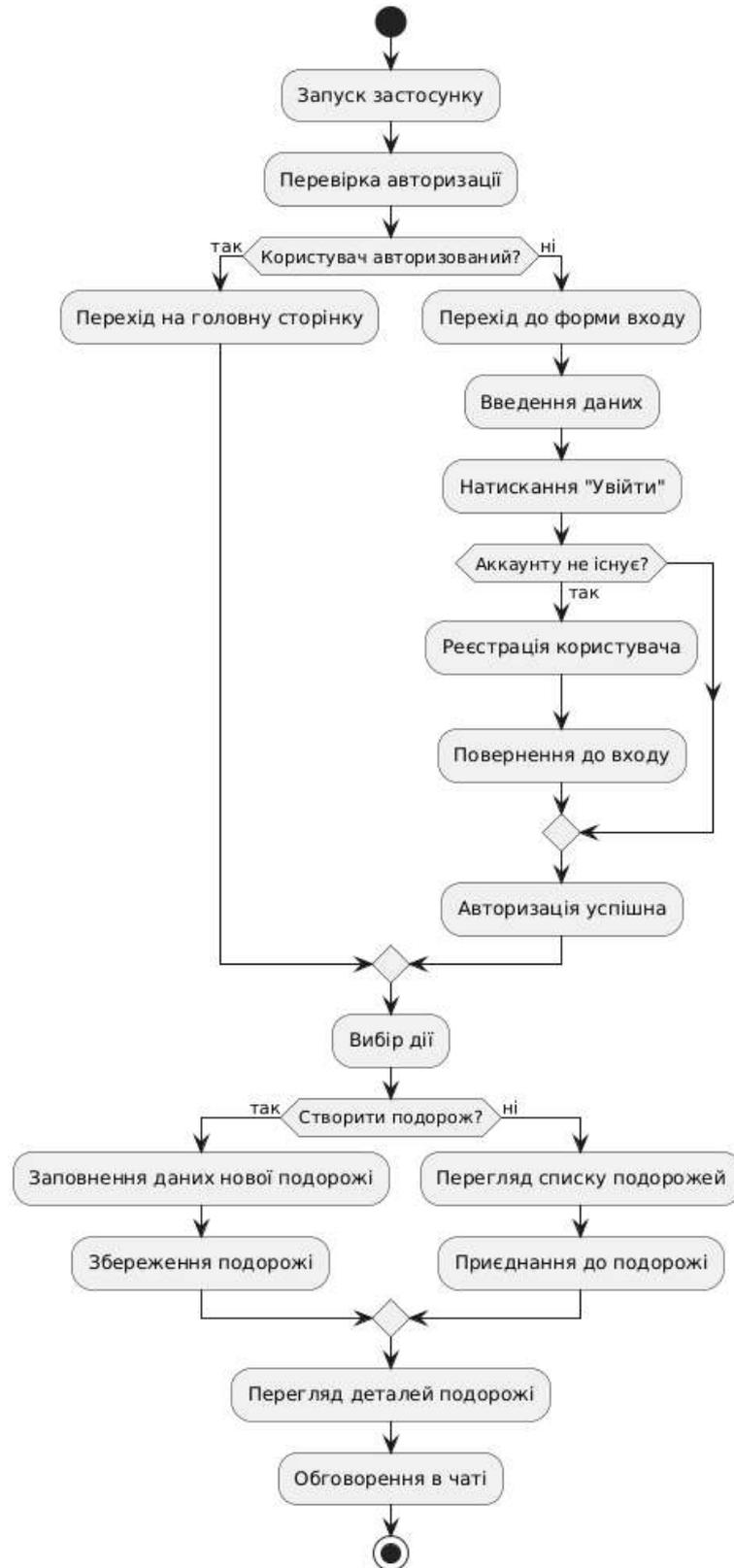


Рисунок 3.2 – Діаграма активності (Activity Diagram) (рисунок виконано самостійно)

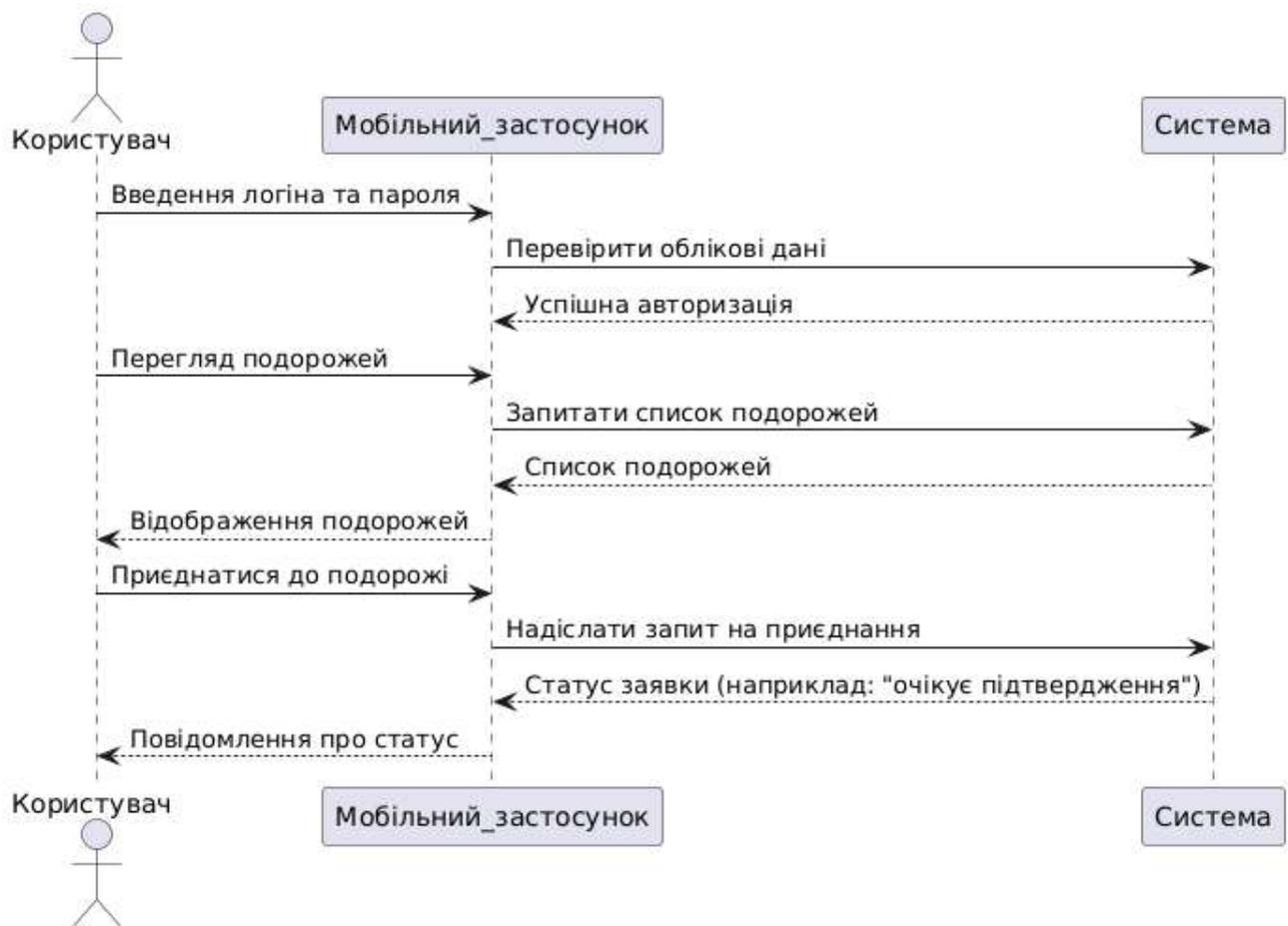


Рисунок 3.3 – Діаграма послідовності (Sequence Diagram) (рисунок виконано самостійно)

3.2 Проектування архітектури

Архітектура мобільного застосунку базується на шаблоні MVVM (Model–View–ViewModel), який є ефективним підходом для розділення логіки представлення, бізнес-логіки та даних у застосунках, розроблених за допомогою .NET MAUI.

У цій архітектурі Model відповідає за структуру даних і логіку доступу до них. Наприклад, сутності «Користувач», «Подорож», «Участь» тощо описують, як виглядають об'єкти в системі, та зберігають відповідну інформацію. ViewModel є посередником між Model та View. Вона містить усю логіку

керування станом інтерфейсу, обробкою подій, змінами даних тощо. Наприклад, ViewModel для екрана списку подорожей міститиме логіку вибору подорожі, фільтрації, а також обробку дій користувача. View – це інтерфейс користувача, який відображає дані, що надходять з ViewModel, та передає події, створені користувачем, назад у ViewModel. Всі екрани мобільного застосунку, такі як головна сторінка, екран подорожі, чат подорожі, екран профілю, реалізуються на рівні View.

Використання MVVM у мобільному застосунку забезпечує масштабованість, легкість у підтримці коду та можливість повторного використання логіки. Також цей шаблон дозволяє реалізовувати юніт-тести бізнес-логіки окремо від інтерфейсу, що покращує якість застосунку.

3.3 Проектування структури зберігання, управління даними

У мобільному застосунку особливу увагу приділено ефективному управлінню даними, враховуючи обмежені ресурси мобільних пристроїв та можливу відсутність постійного інтернет-з'єднання. Дані поділяються на дві основні категорії: локальні та ті, що потребують синхронізації з віддаленим сервером (у майбутньому або в офлайн-режимі).

Для зберігання структурованих даних на пристрої буде використано локальну базу даних SQLite, що добре інтегрується з .NET MAUI. Кожна сутність, визначена у моделі даних (наприклад, користувачі, подорожі, участь, повідомлення тощо), відповідає окремій таблиці у БД.

Сутність «Користувач» має такі атрибути:

- унікальний ідентифікатор користувача;
- повне ім'я;
- електронна пошта для авторизації;
- роль у системі (учасник або адміністратор);
- дата реєстрації;
- прапорець активності облікового запису.

Сутність «Групова подорож» має такі атрибути:

- унікальний ідентифікатор групової подорожі;
- назва подорожі;
- опис маршруту;
- дата початку;
- дата завершення;
- регіон або геолокація;
- максимальна кількість учасників.

Сутність «Участь» має такі атрибути:

- унікальний ідентифікатор участі;
- ідентифікатор користувача;
- ідентифікатор групової подорожі;
- чи є користувач організатором;
- статус заявки (очікує, підтверджено, відхилено);
- дата приєднання до групи.

Сутність «Дані трекінгу» має такі атрибути:

- унікальний ідентифікатор трекінгового запису;
- ідентифікатор участі;
- широта;
- довгота;
- частота серцебиття;
- час надходження даних;
- позначка аномального значення.

Сутність «Повідомлення» має такі атрибути:

- унікальний ідентифікатор повідомлення;
- ідентифікатор участі;
- вміст повідомлення;
- дата й час надсилання.

Сутність «Сигнал тривоги» має такі атрибути:

- унікальний ідентифікатор сигналу тривоги;
- ідентифікатор участі;
- координати сигналу (широта та довгота);
- статус обробки;
- дата створення.

Сутність «Журнал дій користувача» має такі атрибути:

- унікальний ідентифікатор дії користувача;
- ідентифікатор користувача;
- тип дії (вхід, перегляд, редагування);
- короткий опис дії;
- дата й час.

Сутність «Точка маршруту подорожі» має такі атрибути:

- унікальний ідентифікатор точки;
- ідентифікатор групової подорожі, до якої належить ця точка;
- широта;
- довгота;
- назва точки («старт», «оглядова вежа» тощо);
- опис;
- порядковий номер у маршруті.

Основні особливості структури:

1. SQLite виступає основною системою зберігання даних на пристрої. Вона дозволяє швидко виконувати операції читання/запису, підтримує реляційні зв'язки та транзакції.

2. Для об'єктно-реляційного доступу застосовується бібліотека SQLite-net, що дозволяє виконувати запити за допомогою LINQ-виразів, а також забезпечує зручну інтеграцію з MVVM-патерном.

3. З метою безпеки, чутливі дані (наприклад, токени авторизації) зберігаються у Secure Storage, а не в SQLite.

4. Управління даними реалізується через репозиторії та сервіси, які інкапсулюють логіку доступу до БД, спрощують тестування та підтримку коду.

Завдяки такому підходу застосунок зберігає гнучкість, забезпечує швидкий доступ до даних, дозволяє працювати в автономному режимі, а також забезпечує базову масштабованість для майбутньої інтеграції з серверною частиною.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Опис структури проекту

Після проведеного аналізу предметної області та етапу проектування було створено застосунок «TravelMate», призначений для організації та управління груповими подорожами. Для розробки програмного забезпечення було обрано кросплатформний фреймворк .NET MAUI, який поєднує високу продуктивність із можливістю реалізації сучасного та зручного інтерфейсу користувача. Розробка здійснювалася у середовищі Visual Studio, що забезпечує широкі можливості для написання, відлагодження та оптимізації коду, сприяючи створенню якісного, стабільного та підтримуваного програмного продукту. У додатку А відображено дерево архітектури мобільного застосунку.

Проект мобільного застосунку TravelMate містить п'ять основних папок, що були створені під час розробки програми. «Enums», «Models», «Services», «ViewModels» та «Views» – це папки, що містять відповідні аспекти функціональності та архітектури застосунку.

Папки «Enums» та «Models» містять відповідно описи переліків (enum) статусів, типів та інших фіксованих значень, а також моделі даних, які відображають основні сутності застосунку та використовуються для обміну даними між різними його компонентами.

Папка «Services» містить сервіси, відповідальні за взаємодію як із локальною базою даних, так і з зовнішнім Web API. Кожен сервіс реалізує логіку роботи з окремою сутністю застосунку. Наприклад, AuthService відповідає виключно за процеси автентифікації та реєстрації користувача. Аналогічно, інші сервіси відповідають за обробку даних профілю користувача, подорожей та інших сутностей — зокрема, за перегляд наявної інформації, створення нових записів, редагування та видалення існуючих. Такий підхід забезпечує чіткий поділ відповідальностей і сприяє масштабованості та зручності супроводу коду.

Папка «ViewModels» містить класи, що реалізують логіку представлення відповідно до шаблону MVVM (Model-View-ViewModel). ViewModel виступає посередником між моделями даних (Models) та інтерфейсом користувача (Views), забезпечуючи двостороннє зв'язування даних (data binding), обробку команд (commands) та іншу бізнес-логіку, необхідну для роботи сторінок застосунку. Кожна ViewModel відповідає окремій сторінці додатку та містить властивості, що відображаються у розмітці, а також методи для взаємодії з сервісами, перевірки даних, навігації тощо.

Папка «Views» містить XAML-розмітку всіх сторінок інтерфейсу користувача, що відображаються в додатку, а також відповідні програмні файли з логікою для обробки подій, взаємодії з елементами інтерфейсу та зв'язування з відповідними файлами ViewModel у рамках шаблону MVVM (Model-View-ViewModel).

Описана структура проєкту відповідає принципам багаторівневої архітектури та шаблону MVVM, що забезпечує чітке розділення відповідальностей між окремими компонентами системи. Такий підхід сприяє підвищенню читабельності, повторного використання коду та його простішого тестування. Поділ застосунку на логічні частини — Enums, Models, Services, ViewModels та Views — дозволяє кожному елементу виконувати строго визначену функцію, що мінімізує ризик помилок при зміні або розширенні функціоналу. Завдяки цьому проєкт є масштабованим, підтримуваним та легко адаптується до нових вимог. Крім того, така архітектура значно спрощує командну розробку, оскільки дозволяє паралельно працювати над різними модулями без конфліктів у коді.

4.2 Використані бібліотеки

У процесі розробки мобільного застосунку були використані наступні бібліотеки та пакети, що забезпечили реалізацію необхідного функціоналу, зручну архітектуру та інтеграцію з платформними можливостями .NET MAUI:

1. `CommunityToolkit.Mvvm` — бібліотека, що забезпечує зручну реалізацію шаблону MVVM (Model-View-ViewModel), дозволяючи суттєво скоротити обсяг шаблонного коду. Вона підтримує автоматичну генерацію властивостей, що реалізують `INotifyPropertyChanged`, команд (`RelayCommand`), а також інші корисні функції, що спрощують підтримку та розширення проєкту.

2. `SQLitePCLRaw.bundle_e_sqlite3` — пакет, що додає нативну реалізацію SQLite, необхідну для роботи з локальною базою даних за допомогою бібліотеки `sqlite-net-pcl`. Забезпечує повну сумісність з платформою .NET MAUI та дозволяє зберігати дані користувача у пристрої без підключення до мережі.

3. `System.IdentityModel.Tokens.Jwt` — бібліотека для роботи з JWT-токенами. Використовується для реалізації автентифікації та авторизації користувачів через Web API, що забезпечує безпечну взаємодію між мобільним клієнтом і серверною частиною.

4. `Onion.Maui.GoogleMaps` — бібліотека для інтеграції Google Maps у .NET MAUI-застосунки. Вона надає функціональність для відображення інтерактивної карти, роботи з маршрутами, маркерами, а також підтримку геолокації, що дозволяє визначати поточне розташування користувача та взаємодіяти з мапою в режимі реального часу [17].

5. `Plugin.BLE` — кросплатформна бібліотека для роботи з Bluetooth-пристроями у .NET MAUI. У межах проєкту використовується для підключення до фітнес-браслету (наприклад, `Mi Band`) та зчитування показників пульсу, що дозволяє інтегрувати дані про фізичну активність користувача в контексті подорожей.

Усі вищезазначені бібліотеки були обрані з огляду на їхню стабільність, сумісність із технологією .NET MAUI, а також відповідність вимогам проєкту щодо функціональності, надійності та зручності використання.

4.3 Авторизація у програмному застосунку

Для доступу до функцій застосунку користувач повинен пройти процедуру авторизації, що забезпечує захист персональних даних. При реєстрація необхідно вказати електронну пошту. Після заповнення відповідної форми користувачу надсилається лист із посиланням для підтвердження адреси.

Після успішної авторизації система видає токен доступу, який використовується для подальшої автентифікації під час взаємодії з сервером. Токен зберігається локально в `SecureStorage` та автоматично додається до кожного запиту (рис. 4.1), що дозволяє підтверджувати автентичність користувача без необхідності повторного входу.

```
var token = await SecureStorage.GetAsync("auth_token");
if (string.IsNullOrEmpty(token))
{
    return null;
}

var request = new HttpRequestMessage(HttpMethod.Get, "api/travel-groups");
request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", token);
```

Рисунок 4.1 – Додавання токена доступу до заголовку запиту (рисунок виконано самостійно)

Усі введені дані проходять перевірку на коректність: зокрема, перевіряється правильність формату електронної пошти, унікальність облікового запису та заповнення всіх обов'язкових полів. Нижче наведено код методу перевірки валідності даних під час реєстрації:

```
var addr = new System.Net.Mail.MailAddress(registrationDto.Email);

if (addr.Address == registrationDto.Email &&
    !string.IsNullOrEmpty(registrationDto.FirstName) &&
```

```

        registrationDto.FirstName.Length >= 2 &&
        Regex.IsMatch(registrationDto.FirstName, @"^[a-zA-Za-яА-
Яіїєєґґіі'\s\-\]+\$") &&
        !string.IsNullOrEmpty(registrationDto.LastName) &&
        registrationDto.LastName.Length >= 2 &&
        Regex.IsMatch(registrationDto.LastName, @"^[a-zA-Za-яА-
Яіїєєґґіі'\s\-\]+\$") &&
        !string.IsNullOrEmpty(registrationDto.Username) &&
        registrationDto.Username.Length >= 3 &&
        !string.IsNullOrEmpty(registrationDto.Password) &&
        registrationDto.Password.Length >= 6 &&
        Regex.IsMatch(registrationDto.Password, @"[A-Z]") &&
        Regex.IsMatch(registrationDto.Password, @"[a-z]") &&
        Regex.IsMatch(registrationDto.Password, @"[0-9]")

```

В даному блоку спочатку виконується перевірка чи є email у правильному форматі за допомогою `MailAddress` і чи співпадає з оригінальним рядком. Далі вона перевіряє, що ім'я та прізвище не порожні, містять принаймні по 2 символи і складаються лише з літер латиниці або кирилиці, апострофів, пробілів та дефісів. Також перевіряється, що логін не порожній і має мінімум 3 символи. Пароль перевіряється на мінімальну довжину 6 символів і на наявність принаймні однієї великої літери, однієї малої літери та однієї цифри. Якщо всі ці умови виконуються, дані вважаються валідними. На рисунку 4.2 зображено екран реєстрації та екран логіну.

4.4 Подорожі у програмному застосунку

Однією з ключових функціональних можливостей застосунку «TravelMate» є його здатність допомагати користувачам ефективно підбирати та керувати своїми подорожами. Користувачі можуть переглядати подорожі у клієнтській частині застосунку, отримуючи доступ до детальної інформації про кожну з них. Основні параметри, такі як маршрут подорожі, дата початку та завершення, а також учасники, відображаються для кожної подорожі. Місце призначення інтегроване через Google Maps і відображається як у списку подорожей, так і в детальному перегляді конкретної подорожі.

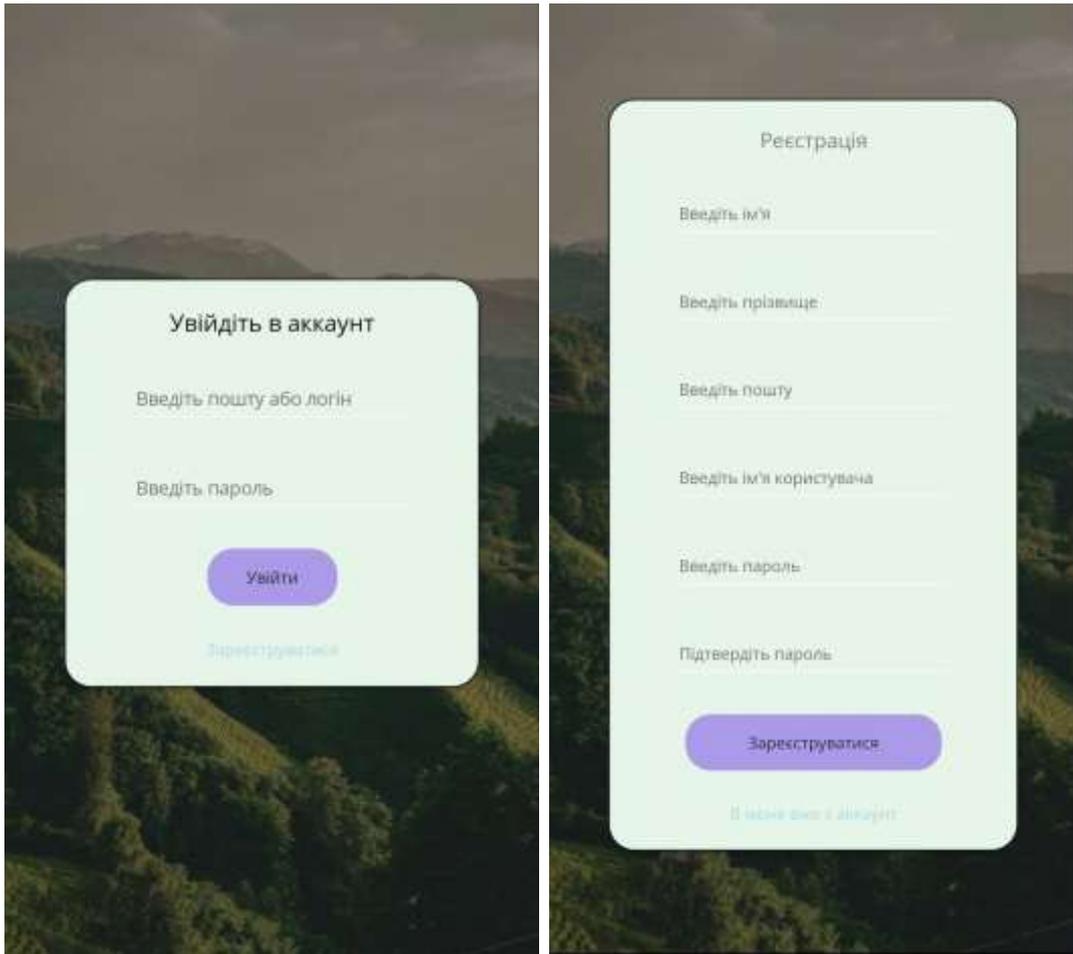


Рисунок 4.2 – Екран логіну та реєстрації (рисунок виконано самостійно)

У межах кожної подорожі користувач може переглянути детальніше інформацію про подорож та маршрут, а також перейти до чату з іншими учасниками. Власник подорожі може взаємодіяти з даними подорожі, зокрема редагувати маршрут, оновлювати подорожі та корегувати склад учасників. На рисунку 4.3 наведено приклад списку подорожей та детальної інформації.

4.5 Моніторинг та надзвичайні ситуації у програмному застосунку

Після початку подорожі запускається функція моніторингу даних користувача, а саме відслідковування місцезнаходження та сканування пульсу завдяки фітнес браслетам. Також передбачено робота в оффлайн режимі, а саме

трекінгові дані зберігаються в локальній базі даних SQLite і при з'єднанні з серверною частиною дані синхронізуються.

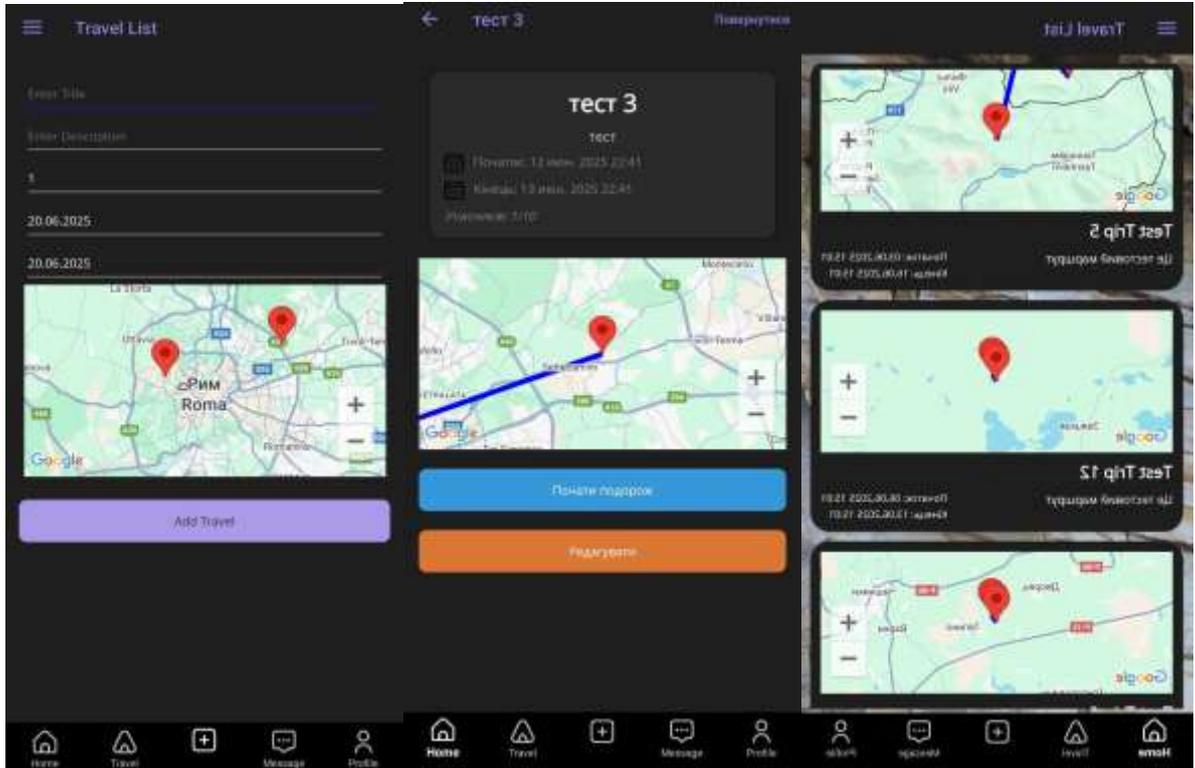


Рисунок 4.3 – Екрани роботи з подорожами (рисунок виконано самостійно)

Нижче наведено приклад коду для моніторингу стану здоров'я учасника подорожі та синхронізації даних з сервером:

```
private async Task TrackingLoopAsync(CancellationToken token)
{
    while (!token.IsCancellationRequested)
    {
        try
        {
            var location = await Geolocation.GetLocationAsync();
            int heartRate = await GetLatestHeartRateFromMiBand();
            var data = new LocalTrackingData
            {
                TravelId = _travelGroupId,
                Latitude = location?.Latitude ?? 0,
                Longitude = location?.Longitude ?? 0,
                HeartRate = heartRate,
                Timestamp = DateTime.UtcNow
            };
            bool sent = await TrySendTrackingDataAsync(data);
        }
    }
}
```

```

        if (!sent)
            _localDb.SaveTrackingData(data);
        else
            await SendCachedDataAsync();
            if (heartRate < 40 || heartRate > 180)
                await SendEmergencyCallAsync(data.Latitude,
data.Longitude, "HeartRateAnomaly");
        }
        catch (Exception ex)
        {
            Debug.WriteLine($"Unable to load tracking data:
{ex.Message}");
            return;
        }
        await Task.Delay(TimeSpan.FromSeconds(30), token);
    }
}

```

4.6 Профіль користувача

Після успішної реєстрації в застосунку «TravelMate» для користувача автоматично створюється персоналізований профіль, який містить основні особисті дані. Його мета — забезпечити індивідуальний підхід та зручність користування. Профіль включає фотографію за замовчуванням, яку користувач з легкістю може замінити на власне зображення, а також ім'я, унікальний тег та електронну адресу, вказані під час реєстрації.

Однією з важливих функцій профілю є можливість редагування персональної інформації. Користувач може змінювати своє ім'я та фото профілю, що дозволяє краще відобразити власну індивідуальність і створити комфортне середовище використання застосунку. Для редагування даних достатньо натиснути на іконку редагування у вікні профілю.

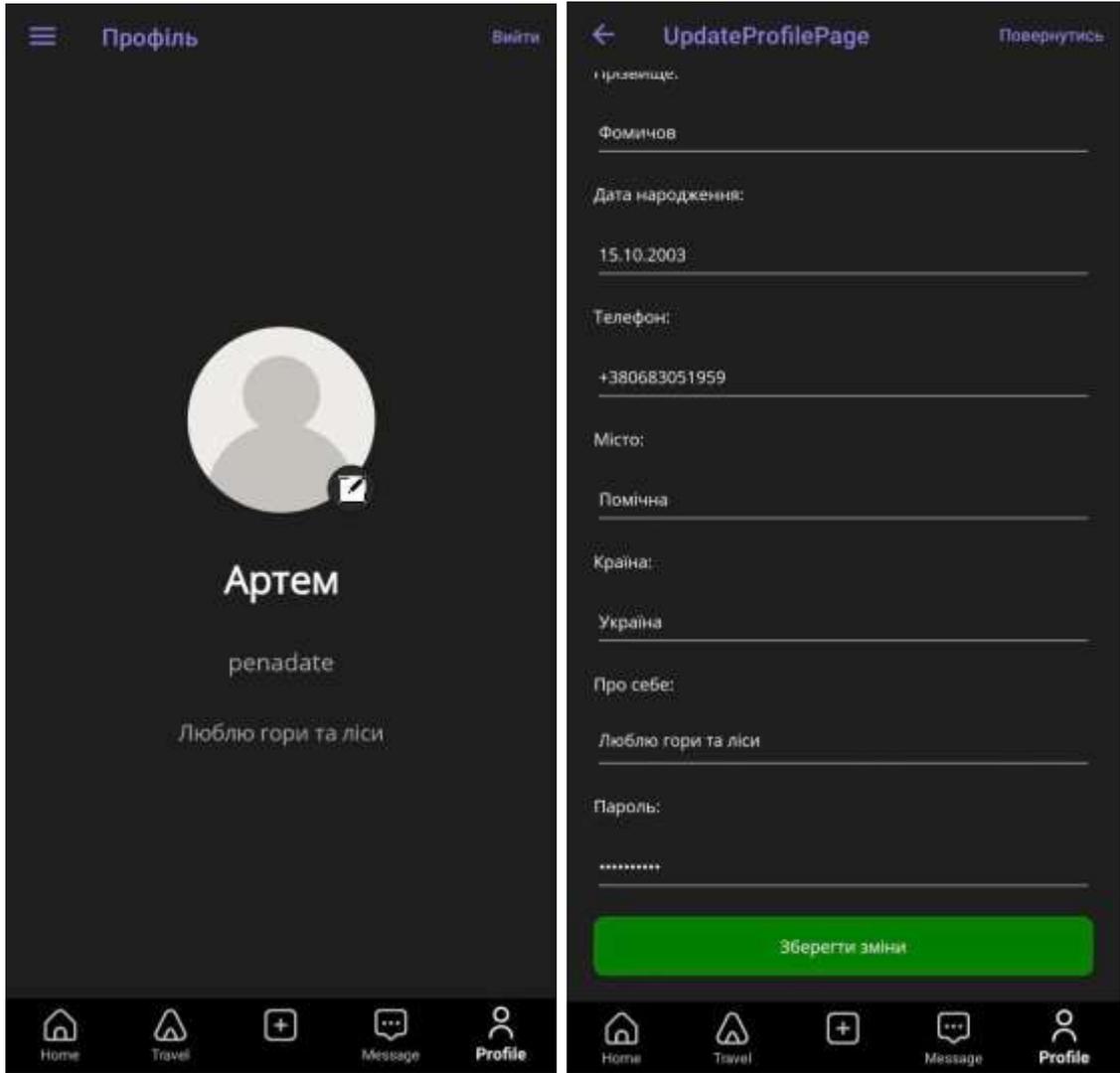


Рисунок 4.4 – Екрани профілю та редагування (рисунок виконано самостійно)

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Тестування програмного застосунку

Для перевірки працездатності та якості розробленого застосунку «TravelMate» застосовувалося мануальне тестування. Тестування охоплювало основні функціональні можливості застосунку з метою виявлення можливих помилок і забезпечення стабільної роботи системи. Зокрема, було протестовано процеси автентифікації та реєстрації користувачів — перевірявся повний цикл створення облікового запису, вхід до системи та обробка типових сценаріїв взаємодії з інтерфейсом.

Особливу увагу було приділено перевірці функціональності, пов'язаної з роботою з подорожами — переглядом наявних, створенням нових, а також надсиланням запитів на участь у вже створених подорожах. У процесі тестування створення нової подорожі перевірялась повна послідовність дій користувача: заповнення основної інформації про подорож (назва, опис, період), а також додавання маршрутних точок. Також було протестовано можливості редагування особистого профілю, зокрема зміну імені та профільного зображення. Перевірялися підтримка поширених форматів зображень (JPEG, PNG), обмеження на розмір файлу, а також коректність оновлення фотографії в інтерфейсі після збереження.

Крім того, виконувалася перевірка на правильність валідації введених користувачем даних і збереження змін без втрати інформації. Проведене тестування підтвердило коректність реалізованих функцій і дозволило виявити та усунути дрібні помилки до фінального запуску застосунку.

Тест-кейс з реєстрацією нового користувача, авторизацією, редагування профілю користувача та планування нової подорожі наведено у таблиці 5.1 – 5.4 відповідно.

Таблиця 5.1 – Тест-кейс №1(таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:		Тест-кейс №1	
Опис функції:		Реєстрація нового користувача	
Власник тесту:		Фомичов Артем Сергійович	
Дата створення:		28.05.2025	
Мета тесту:		Тестування процесу реєстрації користувача у застосунку	
№	Опис випадку	Очікуваний результат	Висновок
Передумова			
1	Відкрити завантажений застосунок	Користувач має доступ до застосунку	Успіх
Реєстрація нового користувача			
1	Відкрити застосунок «TravelMate»	Застосунок відкривається і відображається екран входу з кнопкою «Зареєструватися»	Успіх
2	Натиснути на кнопку «Зареєструватися»	Відкривається форма реєстрації з полями для введення даних.	Успіх
3	Заповнити обов'язкове поле «Ім'я»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
4	Заповнити обов'язкове поле «Прізвище»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
5	Заповнити обов'язкове поле «Пошта»	Поле заповнене та пройшло валідацію на коректність (правильний формат пошти).	Успіх
6	Заповнити обов'язкове поле «Ім'я користувача»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
7	Заповнити обов'язкове поле «Пароль»	Поле заповнене, пароль відповідає вимогам безпеки (не менше 8 символів, містить великі і маленькі літери та цифри)	Успіх
8	Заповнити обов'язкове поле «Підтвердити пароль»	Поле заповнене, введений пароль збігається з паролем у попередньому полі.	Успіх
9	Натиснути кнопку «Зареєструватися»	Застосунок відправляє дані на сервер для реєстрації нового користувача. Після успішної реєстрації, відправляється лист для підтвердження електронної пошти	Успіх
Результати тестування			
Тестувальник Фомичов А.С.		Дата проведення тестування 28.05.2025	Результат тесту (P/F/V): ПРОЙДЕНО (P)

Таблиця 5.2 - Тест-кейс №2 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:		Тест-кейс №2	
Опис функції:		Авторизація користувача	
Власник тесту:		Фомичов Артем Сергійович	
Дата створення:		28.05.2025	
Мета тесту:		Тестування процесу авторизації користувача у застосунку	
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити завантажений застосунок	Користувач має доступ до застосунку	Успіх
Авторизація користувача			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити застосунок «TravelMate»	Застосунок відкривається і відображається екран входу	Успіх
2	Заповнити обов'язкове поле «Пошта або Ім'я користувача»	Поле заповнене та пройшло валідацію на коректність (правильний формат пошти або ім'я користувача).	Успіх
3	Заповнити обов'язкове поле «Пароль»	Поле заповнене, пароль відповідає вимогам безпеки (не менше 8 символів, містить великі і маленькі літери та цифри)	Успіх
4	Натиснути кнопку «Увійти»	Застосунок отримує дані з серверу для входу користувача. Після успішної авторизації, користувач автоматично входить у систему та відображається головна сторінка та меню застосунку.	Успіх
5	Перезавантажити застосунок	Застосунок перевіряє наявність токена та його актуальність та автоматично авторизує користувача	Успіх
Результати тестування			
Тестувальник Фомичов А.С.		Дата проведення тестування 28.05.2025	Результат тесту (P/F/V): ПРОЙДЕНО (P)

Таблиця 5.3 – Тест-кейс №3 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:		Тест-кейс №1	
Опис функції:		Редагування профілю користувача	
Власник тесту:		Фомичов Артем Сергійович	
Дата створення:		28.05.2025	
Мета тесту:		Тестування процесу редагування профілю користувача у застосунку	
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Користувач зареєстрований у системі та увійшов у свій обліковий запис	Користувач увійшов у свій обліковий запис та перейшов на сторінку з меню	Успіх
2	Перейти до сторінки редагування профілю	Користувач знаходиться на сторінці редагування профілю з введеними поточними даними	Успіх
Редагування профілю користувача			
№	Опис випадку	Очікуваний результат	Висновок
1	Змінити поле «Ім'я»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
2	Змінити поле «Прізвище»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
3	Змінити поле «Дата народження»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
4	Змінити поле «Місто»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
5	Змінити поле «Країна»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
6	Змінити поле «Про себе»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
7	Змінити поле «Пароль»	Поле заповнене, пароль відповідає вимогам безпеки (не менше 8 символів, містить великі і маленькі літери та цифри)	Успіх
8	Натиснути кнопку «Зберегти зміни»	Застосунок відправляє дані на сервер для редагування профілю користувача.	Успіх
Результати тестування			
Тестувальник Фомичов А.С.		Дата проведення тестування 28.05.2025	Результат тесту (P/F/V): ПРОЙДЕНО (P)

Таблиця 5.4 – Тест-кейс №4 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:		Тест-кейс №1	
Опис функції:		Додавання нової подорожі	
Власник тесту:		Фомичов Артем Сергійович	
Дата створення:		28.05.2025	
Мета тесту:		Тестування додавання нової подорожі у застосунку	
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Користувач зареєстрований у системі та увійшов у свій обліковий запис	Користувач увійшов у свій обліковий запис та перейшов на сторінку з меню	Успіх
2	Перейти до додавання нової подорожі	Користувач знаходиться на сторінці додавання подорожі	Успіх
Редагування профілю користувача			
№	Опис випадку	Очікуваний результат	Висновок
1	Заповнити обов'язкове поле «Назва»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
2	Заповнити обов'язкове поле «Опис»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
3	Заповнити обов'язкове поле «Кількість учасників»	Поле заповнене, кількість правильна, значення збережене.	Успіх
4	Заповнити обов'язкове поле «Дата початку»	Поле заповнене, формати дати правильний, значення збережене.	Успіх
5	Заповнити обов'язкове поле «Дата кінця»	Поле заповнене, формати дати правильний, значення збережене.	Успіх
6	Обрати точку маршруту	Відкрилось діалогове вікно	Успіх
7	Заповнити обов'язкове поле «Назва точки»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
8	Заповнити обов'язкове поле «Опис точки»	Поле заповнене, кількість символів правильна, значення збережене.	Успіх
9	Натиснути «ОК»	Діалогове вікно закрилось, на мапі додалась точка маршруту.	Успіх
10	Натиснути кнопку «Додати подорож»	Застосунок успішно відправив запит серверу та відкрив список подорожей	Успіх
Результати тестування			
Тестувальник Фомичов А.С.		Дата проведення тестування 28.05.2025	Результат тесту (P/F/B): ПРОЙДЕНО (P)

Отже, проведене тестування застосунку «TravelMate» підтвердило його стабільність і відповідність функціональним вимогам. Мануальне тестування дало змогу ретельно перевірити ключові аспекти роботи, зокрема процеси авторизації, реєстрації, редагування профілю та взаємодії з подорожами. Усі тест-кейси було успішно виконано, що свідчить про коректну реалізацію функцій і надійну роботу застосунку. Перевірені процеси виконуються відповідно до очікувань, без збоїв або помилок. Такий підхід до тестування дозволяє впевнено стверджувати, що застосунок забезпечує якісний користувацький досвід і відповідає вимогам кінцевих користувачів.

ВИСНОВКИ

У процесі виконання дипломного проєкту було розроблено мобільний застосунок «TravelMate», призначений для організації та управління груповими подорожами. На початковому етапі було проведено аналіз предметної області, визначено вимоги до функціональності системи та спроектовано її архітектуру з урахуванням принципів масштабованості, модульності та зручності використання.

У результаті розробки створено повноцінний мобільний застосунок, який дозволяє користувачам реєструватися, редагувати персональний профіль, створювати власні подорожі, переглядати доступні маршрути, приєднуватися до груп, взаємодіяти з іншими учасниками за допомогою повідомлень, переглядати маршрутні точки на карті за допомогою інтеграції з Google Maps, а також слідкувати за статистикою активності.

У ході реалізації проєкту:

1. Було сформульовано повні функціональні та нефункціональні вимоги до системи;
2. Спроектовано структуру бази даних і взаємозв'язки між основними сутностями (користувач, подорож, маршрутна точка, повідомлення, трекінг тощо);
3. Розроблено UML-діаграми (діаграми варіантів використання, послідовності, активності), які описують логіку взаємодії користувача з системою;
4. Імплементовано інтерфейс користувача з урахуванням принципів UX-дизайну та підтримкою навігації за допомогою Shell;
5. Реалізовано двостороннє зв'язування даних відповідно до патерну MVVM з використанням бібліотеки CommunityToolkit.Mvvm;
6. Створено сервіси для взаємодії з Web API та локальною базою даних SQLite;

7. Проведено ручне тестування основної функціональності, що підтвердило стабільність роботи застосунку та відповідність заданим вимогам.

Розробка мобільного застосунку «TravelMate» здійснювалася з урахуванням сучасних тенденцій у програмуванні та дизайні інтерфейсів. Це дозволило створити високоякісний продукт, який не лише відповідає функціональним вимогам, але й забезпечує приємний користувацький досвід. Завдяки використанню перевірених архітектурних рішень, таких як трирівнева архітектура, та сучасних технологій управління станом, застосунок має гнучку архітектуру та легко піддається розширенню, що дозволяє у майбутньому додати підтримку нових функцій.

Таким чином, проєкт реалізував усі заплановані цілі та продемонстрував практичне втілення сучасних підходів до розробки кросплатформених мобільних застосунків з використанням .NET MAUI. Отримані результати можуть бути використані як основа для подальшого розвитку повноцінного сервісу для планування активного відпочинку та групових подорожей. Врахування актуальних потреб користувачів і аналіз ринку дають підстави вважати, що «TravelMate» має всі шанси стати конкурентоспроможним продуктом, який буде корисним широкому колу мандрівників. Проєкт має потенціал до подальшого розвитку та масштабування, що відкриває нові перспективи його використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Москвяк Я. Є. (2022). Сучасний стан та перспективи розвитку екологічного туризму в карпатському регіоні України. *Herald of khmelnytskyi national university. Economic sciences*, 304(2(1), 184-188. [https://doi.org/10.31891/2307-5740-2022-304-2\(1\)-26](https://doi.org/10.31891/2307-5740-2022-304-2(1)-26)
2. Troupe – Group Travel Planning App by JetBlue [Електронний ресурс]. – URL: <https://troupe.com>. (дата звернення 14.04.2025).
3. Wanderlog – Trip Planner with Itineraries & Maps [Електронний ресурс]. – URL: <https://wanderlog.com>. – Загол. з екрану. (дата звернення 14.04.2025).
4. Наукові праці Вінницького національного технічного університету, (2), 2024. DOI: 10.31649/2307-5376-2024-2-17-27.
5. Полторацький А. О. Моделі синхронізації безконфліктних реплікованих типів даних в мобільних офлайн застосунках / А. О. Полторацький // Радіоелектроніка та молодь у XXI столітті : матеріали 24 Міжнар. молодіж. форуму, 7–9 квіт. 2020 р. – Харків : ХНУРЕ, 2020. – Т. 6. – С. 165–166.
6. .NET MAUI – Multi-platform App UI [Електронний ресурс]. – URL: <https://learn.microsoft.com/en-us/dotnet/maui/> (дата звернення 10.05.2025).
7. Model-View-ViewModel (MVVM) Pattern [Електронний ресурс]. – URL: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm> (дата звернення 10.05.2025).
8. SQLite-net: Simple, powerful, cross-platform SQLite client [Електронний ресурс]. – URL: <https://github.com/praeclarum/sqlite-net> (дата звернення 13.05.2025).
9. Plugin.BLE – Bluetooth Low Energy plugin for Xamarin and .NET MAUI [Електронний ресурс]. – URL: <https://github.com/xabre/xamarin-bluetooth-le> (дата звернення 13.05.2025).
10. Каук В. І. Дослідження можливостей застосування односпрямованного підходу до проектування мобільних архітектур на прикладі Redux / В. І. Каук, С.

О. Губар // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тез. доп. дванадцятій міжнародній науково-технічній конференції, 27–28 квітня 2022 р. – Т. 2. – Баку–Харків–Жиліна, 2022. – С. 140.

11. Новоселов С. П. Основи розробки кросплатформного програмного забезпечення на Avalonia : навч. посіб. / С. П. Новоселов, О. В. Сичова ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : Вид-во Іванченка І. С., 2024. – 267 с. – ISBN 978-617-8332-57-0.

12. "Herald of Khmelnytskyi National University. Technical Sciences, 333(2), 2024. -с.291-296. DOI: 10.31891/2307-5732-2024-333-2-46 "

13. Introduction to JSON Web Tokens [Електронний ресурс]. – URL: <https://jwt.io/introduction> (дата звернення 13.05.2025).

14. Калінін І. М. Використання сучасного транспорту даних у захищених серверах / І. М. Калінін, А. В. Власов // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: тези доповідей одинадцятій міжнародній науково-технічній конференції, 8–9 квітня 2021р. – ВА ЗС АР; НТУ "ХПІ"; НАУ, ДП "ПДПРОНДІАВІАПРОМ"; УМЖ, 2021. – Т. 2, секції 3-5. – С. 56.

15. Черних І. А. Особливості обміну повідомленнями в режимі реального часу / І. А. Черних, О. Ф. Лановий // Радіоелектроніка та молодь у ХХІ столітті : матеріали 29-го Міжнар. молодіж. форуму, 16–19 квітня 2025 р. – Харків : ХНУРЕ, 2025. – Т. 6 – С. 349-351.

16. Коч, Хатіце, Ердоган, Алі Мерт, Баржаклі, Юсеф, Пекер, Серхат. UML-діаграми в дослідженнях з інженерії програмного забезпечення: систематичний огляд літератури // Proceedings. – 2021. – Т. 74. – С. 13. – DOI: 10.3390/proceedings2021074013.

17. Onion.Maui.GoogleMaps – Google Maps for .NET MAUI [Електронний ресурс]. – URL: <https://github.com/onionyoo/Onion.Maui.GoogleMaps> (дата звернення 20.05.2025).

18. Посилання на архів проекту
https://github.com/NureFomychovArtem/2025_B_PI_PZPI-21-1_Fomychov_A_S