Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Застосунок для мобільного пристрою з використанням .NET MAUI

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІз-21-1
Тетяна ПЕРЕСІЧАНСЬКА
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник: ст. викл. Владислав ДЯЧЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Пересічанській Тетяні Миколаївні
(прізвище, ім'я, по батькові)

1. Тема роботи Застосунок для мобільного пристрою з використанням .NET MAUI

затверджена наказом по університету від "05" травня 2025 р. № 72Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи
1) мова програмування C#; 2) мова розмітки XAML;
3) середовище розробки Visual Studio; 4) шаблон проектування MVVM

4. Перелік питань, що потрібно опрацювати у роботі
1) аналіз предметної області та постановка завдання
2) вибір програмних засобів
3) програмна реалізація
4) тестування
5) аналіз отриманих результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд-презентація - 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	05.05.2025-09.05.2025	
2	Аналіз проблеми та огляд існуючих рішень	10.05.2025-17.05.2025	
3	Вибір програмних засобів	18.05.2025-20.05.2025	
4	Побудова логічної структури застосунку	21.05.2025-26.05.2025	
5	Реалізація та тестуваня	27.05.2025-10.06.2025	
6	Аналіз результатів	11.06.2025-13.06.2025	
7	Оформлення ПЗ	14.06.2025-15.06.2025	

Дата видачі завдання " 05 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. Дяченко В.О.
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 19 рис., 1 табл., 2 дод., 18 джерел.

.NET MAUI, СОЦІАЛЬНА МЕРЕЖА, МОБІЛЬНИЙ ЗАСТОСУНОК, БАЗА ДАНИХ, КРОСПЛАТФОРМЕННА РОЗРОБКА.

Метою кваліфікаційної роботи є розробка мобільного застосунку з використанням .NET MAUI.

У ході виконання кваліфікаційної роботи розглянуто принципи та методи побудови кросплатформеного застосуноку, принципи архітектурних рішень, які використовуються при побудові масштабованих мобільних застосунків. У першому розділі проаналізовано сферу застосування, актуальність та теоретичні основи побудови мобільних застосунків, розглянуто існуючі мобільні застосунки. У другому розділі проведено вибір програмних засобів та побудова логічної структури застосунку. У третьому розділі розроблено програмне забезпечення для реєстрації та авторизації користувачів, публікації повідомлень, обмін повідомленнями, і проведено тестування застосунку. В процесі розробки використано мову програмування C#, мову розмітки XAML та базу даних MySQL для зберігання інформації про користувачів. Розроблений застосунок є перспективним як застосунок для спілкування між студентами в освітньому середовищі.

ABSTRACT

Bachelor’s thesis: 78 pages, 19 figures, 1 tables, 2 appendices, 18 sources.

.NET MAUI, SOCIAL NETWORK, MOBILE APPLICATION, DATABASE, CROSS-PLATFORM DEVELOPMENT.

The major goal of this thesis is to develop a mobile application using .NET MAUI.

In order to qualification work, the principles and methods of building a cross-platform application, the principles of architectural solutions used in building scalable mobile applications are considered. The first chapter analyzes the scope, relevance and theoretical foundations of building mobile applications, and reviews existing mobile applications. The second section describes the selection of software tools and the construction of the logical structure of the application. In the third section, the software for registration and authorization of users, publishing messages, messaging, and testing of the application is developed. The development process used the C# programming language, the XAML markup language, and the MySQL database for storing information about users. The developed application is promising as an application for communication between students in an educational environment.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД - база даних

ОС - операційна система

ШІ - штучний інтелект

API - програмний інтерфейс програми (англ., Application Programming Interface)

IDE - Інтегроване Середовище Розробк (англ., Integrated Drive Electronics)

ВСТУП

З кожним роком зростає кількість користувачів мобільних пристроїв, і разом із цим зростає попит на якісні, зручні та функціональні мобільні застосунки. Через збільшенний попит на мобільні застосунки є необхідність у створенні багатоплатформених додатків які забезпечують швидку розробку, просте обслуговування та високу продуктивність. Важким у реалізації є розробка програмного забезпечення для кількох платформ одночасно, таких як Android, iOS, при умові якщо використовувати під кожну платформу окремий фреймворк. Внаслідок цього виникає проблема коли розробник більш якісно створює тільки для однієї платформи, або оновлення застосунку та вирішення проблем в першу чергу буде на одній з платформ. Тому проблемою є створення гнучких кросплатформенних застосунків, які включають сучасний інтерфейс, зручну навігацію, багатий функціонал і водночас ефективну підтримку обох популярних мобільних платформ Android та iOS.

Зараз існують ряд рішень для кросплатформової розробки, серед яких React Native, Flutter, Xamarin тощо. Водночас компанія Microsoft представила новий фреймворк .NET MAUI, яка надає розробникам змогу створювати нативні мобільні та десктопні застосунки з єдиною базою коду, використовуючи мову C# та XAML. Завдяки цьому, знижується вартість розробки, спрощується супровід застосунків і скорочується час виходу продукту на ринок.

В Україні та у світі ще відносно мало реальних практичних реалізацій проєктів з використанням .NET MAUI. Через те, що більшість існуючих прикладів обмежуються навчальними чи демонстраційними застосунками, це дає можливість дослідити .NET MAUI та розробляти нові застосунки за допомогою технології цього фреймворку. Застосунок з використання .NET MAUI використовується в різних стартапах, корпоративних та

індивідуальних рішеннях.

Метою роботи є розробка мобільного багатоплатформного застосунку з застосуванням .NET MAUI, перевірити переваги та недоліки фреймворку в контексті продуктивності, зручності підтримки та масштабування. Проаналізувати застосунки які розроблені за допомогою .NET MAUI, розробити алгоритм роботи застосунку, розробка та тестування програмного забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Сфера застосування та актуальність мобільного застосунку

.NET MAUI зручний для розробки широкого спектру крос платформних мобільних додатків, наприклад: застосунки для ігор, соціальних мереж або бізнес застосунки, навчальні застосунки або для електроної комерції. Фреймворк зручний як для звичайних користувачів, так і стартап-компаній або корпорацій.

Інтерфейс .NET MAUI розроблений для підтримки різноманітних типів застосунків і варіантів використання, вбудована підтримка елементів керування дозволяє створювати застосунки, оптимізовані для інтерфейсу користувача та взаємодії з користувачем кожної платформи. Це допомогає розробникам швидше створювати програми, які надалі можна буде підтримувати набагато швидше та ефективніше.

.NET MAUI підтримує ряд функцій пристроїв, таких як камера, GPS і push-сповіщення, що дозволяє створювати застосунки, які використовують ці можливості. Це дає змогу розробникам реалізувати проекти від простого застосунку для підвіщення продуктивності до складного рішення корпоративного рівня. Також є інструменти та можливості, щоб створювати унікальні крос-платформні мобільні застосунки.[1]

Актуальністю .NET MAUI є те, що за допомогою цього фреймворку можливо розробляти багатоплатформні застосунки в рамках одного проекту, це дає змогу не витрачати ресурси та час на декілька проєктів, створюючи один і той самий застосунок для кожної платформи. При цьому забеспечується гнучкість розробки специфічного для платформи коду і ресурсів там, де це необхідно, через що можливо створити застосунок з більшими можливостями. За його допомогою розробники можуть централізувати більшу частину логіки програми та дизайну інтерфейсу в

єдиній кодовій базі, що спрощує процес розробки.[2]

Таким чином, за допомогою .NET MAUI можливо розробити будь який застосунок, який буде корисним для користувачів. При створені високопродуктивних програм головною перевагою .NET MAUI є його здатність скорочувати час і витрати на розробку, що дозволяє розробникам більш поглиблено та структуровано створювати програмне забезпечення та підтримку декількох програм.

1.2 Теоретичні основи та концепції побудови мобільних застосунків

Для розробки мобільних застосунків потрібно враховувати різні аспекти: короткий життєвий цикл розробки, який впливає на якість застосунку; можливості мобільних пристроїв, оскільки пристрої мають обмежені ресурси, процесорну потужність, пам’ять, батарею. Слід враховувати під час розробки мобільність та зручність для користувачів, специфікації мобільних пристроїв, такі як розмір екрану, дизайн інтерфейсу користувача і навігації, безпека і конфіденційність користувачів, а саме: застосунки повинні відповідати сучасним вимогам до захисту даних користувачів. Також застосунки потребують маркетингу, щоб більше людей були в них зацікавлені.

Життєвий цикл розробки мобільного застосунку складається з наступних етапів:

- Аналіз ідеї застосунку. Для цього потрібно визначити цільову аудиторію, функціонал та бізнес-цілі проекту.

- Дизайн інтерфейсу користувача включає в себе розробку логічної структури, навігації, стилів, адаптивного дизайну.

- Розробка застосунку з використанням інструментів та мов програмування цільової платформи.

- Тестування застосунку на різних пристроях, операційних системах, виявлення багів.

- Публікація застосунку в магазині цільової платформи, нові функції або оновлення застосунку випускаються в послідовних версіях застосунку в магазині цільової платформи.

Щоб розробити мобільний застосунок для багатьох платформ, вищеперечислений життєвий цикл розробки повторюється для кожної платформи, за винятком першого етапу аналізу вимог до застосунку. Тому, щоб не витрачати багато ресурсів і часу, варто використовувати розробку кросплатформеного застосунку. [3]

Однією з найважливіших проблем є необхідність забезпечення доступності застосунку на різних мобільних операційних системах, таких як Android та iOS. Ось чому існує потреба в кросплатформенній розробці - методі, який дозволяє створювати застосунки, які працюють на різних платформах з єдиною кодовою базою.

Перевагою використання .NET MAUI є те, що структура застосунку зазвичай побудована з використанням патерну MVVM, який дозволяє відокремити логіку застосунку від інтерфейсу користувача. Такий підхід покращує виявлення помилок в коді, модульність і повторне використання компонентів. .NET MAUI надає набір вбудованих можливостей, які суттєво спрощують розробку сучасних мобільних застосунків

Розробка застосунків з використанням .NET MAUI може поєднувати створення інтерфейсу користувача з програмним кодом , що забезпечує функціональність інтерфейсу. Спочатку такий підхід може здатися зручним, але в міру зростання складності проекту, розширення його функціональності і масштабування архітектури, можуть виникнути серйозні труднощі з супроводом і модифікацією коду. Основною проблемою є тісний взаємозв'язок між візуальними компонентами інтерфейсу і логікою додатку. Це ускладнює внесення змін до інтерфейсу, оскільки будь-яка зміна може вимагати перегляду логіки додатку, що, в свою чергу, збільшує складність підтримки і знижує гнучкість розробки. Це також створює перешкоди для модульного тестування, тому що логіка і представлення занадто тісно

переплетені.

Для вирішення цієї проблеми використовуємо патерн проектування MVVM, який дозволяє чітко розмежувати логіку інтерфейсу, бізнес-логіку та логіку взаємодії між ними. Завдяки такій структурі, код стає більш організованим, що значно полегшує тестування окремих компонентів, підвищує зручність внесення змін і сприяє більш легкому обслуговуванню в довгостроковій перспективі. Використання MVVM сприяє повторному використанню коду і дозволяє різним учасникам команди, наприклад, розробникам і UI-дизайнерам, працювати над окремими частинами програми паралельно, не створюючи конфліктів. Такий підхід особливо важливий для великих і складних проектів, де стабільність, масштабованість і супроводжуваність є критично важливими.

У патерні MVVM є три основні компоненти (рисунок 1.1): модель, представлення та модель представлення.

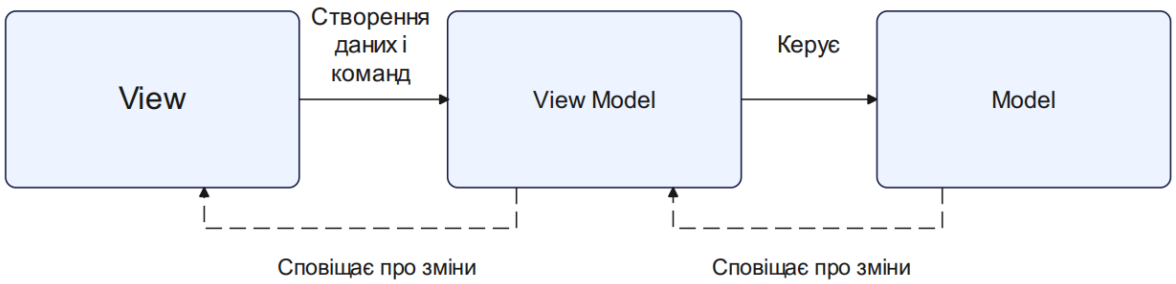

Рисунок 1.1 - Діаграма взаємодії між компонентами шаблону MVVM

Окрім чіткого розуміння ролі кожного компонента в патерні MVVM, не менш важливо розуміти, як вони взаємодіють. На концептуальному рівні View має доступ до ViewModel і може взаємодіяти з нею так само як ViewModel може опосередковано реагувати на зміни у View. Проте, існує чітке розділення між моделлю та іншими компонентами: Model не має уявлення про існування ViewModel або View, а ViewModel не має прямої залежності від View. Такий підхід дозволяє ViewModel діяти як посередник,

який повністю ізолює інтерфейс користувача від бізнес-логіки. Це забезпечує можливість змінювати або вдосконалювати Model без необхідності змінювати інтерфейс користувача, що підвищує гнучкість архітектури та полегшує супровід додатку.[4]

Однією з ключових переваг використання патерну MVVM є чіткий розподіл обов'язків між різними частинами програми. Така структура дозволяє легше орієнтуватися в коді, швидко знаходити потрібну частину функціоналу та вносити в неї зміни без ризику порушити роботу інших компонентів системи. Завдяки модульності та цілеспрямованості кожного елемента, внесення змін або доопрацювань не вимагає глибокого аналізу всієї програми, що значно спрощує процес обслуговування. Архітектура MVVM сприяє створенню ізольованих, добре структурованих блоків коду. При правильному підході зовнішні залежності, наприклад, сервіси або джерела даних, і логіка взаємодії з користувачем відокремлені від основної бізнес-логіки. Це дозволяє легко тестувати окремі компоненти незалежно один від одного. Зокрема, модульні тести можуть бути зосереджені на ViewModel або Model без необхідності взаємодії з інтерфейсом користувача. Ця властивість значно підвищує якість програмного забезпечення, дозволяючи виявляти і виправляти помилки на етапі розробки, а також гарантує стабільність роботи програми навіть при внесенні наступних змін. Також забезпечує високу розширюваність системи. Завдяки чітко визначеним межам між компонентами та структурованості коду, кожна частина програми може бути відносно легко замінена, розширена або повторно використана в іншому контексті. Це відкриває широкі можливості для масштабування: додавати нові функції, компоненти або інтеграції, не порушуючи існуючої логіки. [5]

Розробка кросплатформних мобільних застосунків за допомогою .NET MAUI дозволяє ефективно розробляти застосунки для Android, iOS використовуючи єдину кодову базу на C#. Такий підхід значно спрощує розробку та підтримку програмного забезпечення. В архітектурі .NET MAUI лежить патерн MVVM, який забезпечує чітке розділення логіки інтерфейсу

та бізнес-логіки, сприяючи масштабованості та повторному використанню коду.

1.3 Аналіз існуючих мобільних застосунків

Перед початком розробки застосунку потрібно провести аналіз існуючих застосунків, які з них користуються більшим попитом або звернути увагу на недоліки інших застосунків, щоб в майбутньому уникнути вирішення цих проблем. За допомогою такого аналізу розробнику буде легше створити конкурентоспроможний застосунок.

Приклади застосунків, розроблених з використанням .NET MAUI:

Financing Tracker App - це безкоштовний застосунок який використовується у вільному доступі. Має функцію допомагати відслідковувати витрати і доходи та ефективно управляти особистими фінансами без зайвих витрат часу. Є зрозумілий інтерфейс, який дозволяє додавати транзакції дуже швидко, автоматично складає поточний баланс і створює графічну діаграму, щоб користувачі застосунку бачили схему витрат. За допомогою готових шаблонів користувач створює власні категорії, вибирає будь-які кольори та обирає для них відповідні категорії. Що дозволяє зручно орієнтуватися у власних шаблонах. Застосунок формує звіти за періодами та має функцію нагадування про регулярні платежі, щоб не забувати про важливі фінансові зобов'язання(рисунок 1.2) .[6]

.NET MAUI - Weather '21 - демонстраційний застосунок, що показує поточну погоду, використовуючи зовнішнє API. Застосунок містить 3-денний прогноз, який доступний під поточними даними про погоду на головному екрані, сторінку обраного, яка дозволяє користувачам налаштувати до 8 обраних місць для швидкого доступу до них. Через те , що застосунок є демонстраційним, він більш підходить для навчання .NET MAUI. [7]

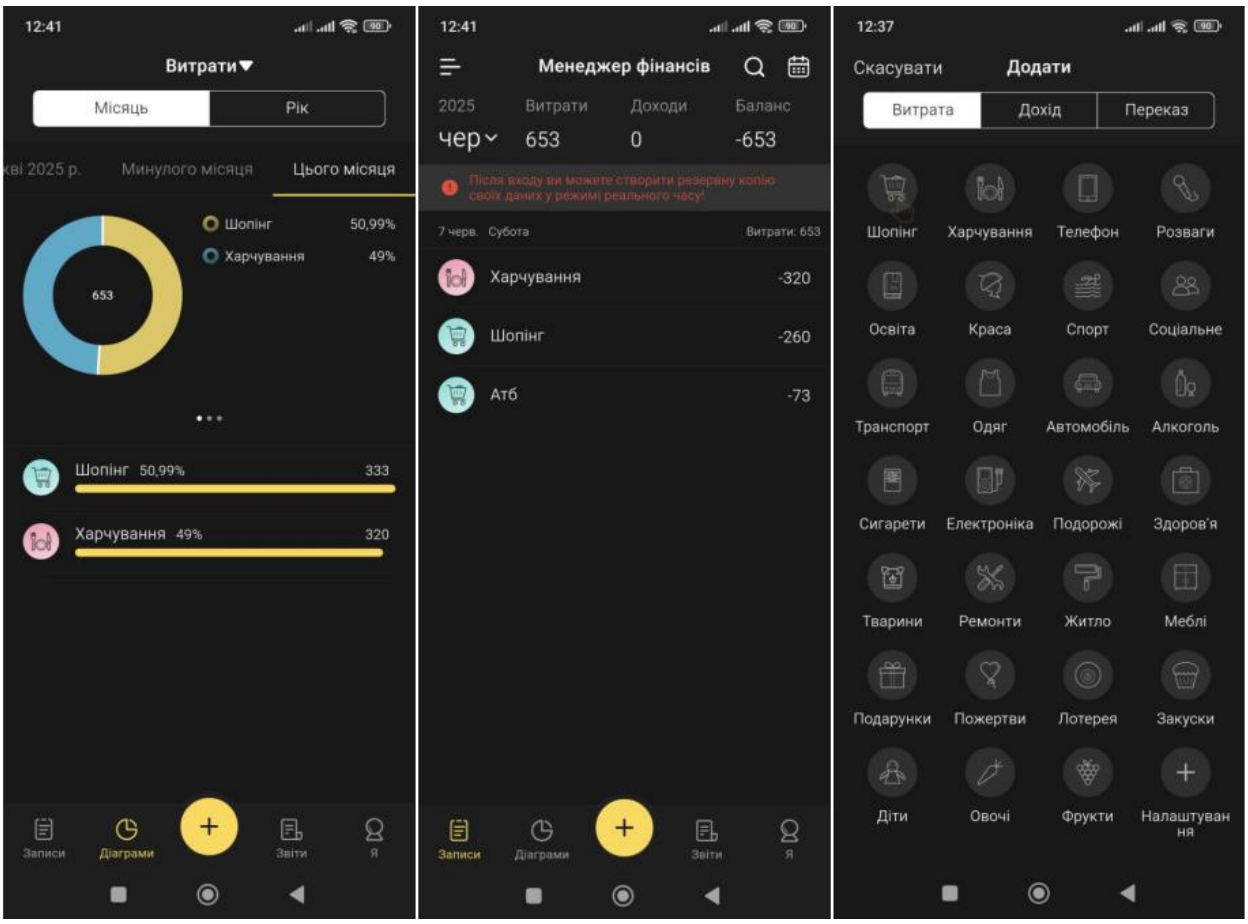

Рисунок 1.2 - Застосунок “Financing Tracker App”

Seeing AI - безкоштовна програма, яка допомогає спільноті сліпих і людей зі слабким зором відкрити візуальний світ, описуючи людей, текст і об’єкти поблизу. Поки що проект є дослідницьким з використанням ШІ. Seeing AI розрізняє текст та озвучує цей текст. Описує фотографії та відповідає на поставленні запитання про зроблене фото. Розпізнає продукти, людей, валюту або яскравість світла чи кольори, що дає змогу людям з порушенням зору більше можливостей (рисунок 1.3).[8]

FinTrack - аналог сервісу Financing Tracker App, застосунок для фінансового відстеження. Інтерфейс застосунку є доволі зручним, FinTrack дозволяє відстежувати витрати, створювати бюджети та планувати витрати, контролювати борги та планувати їх погашення. Але також має досить багато недоліків, тому що при тестуванні застосунку постійно виникають помилки, що робить неможливим його використання (рисунок 1.4).[9]

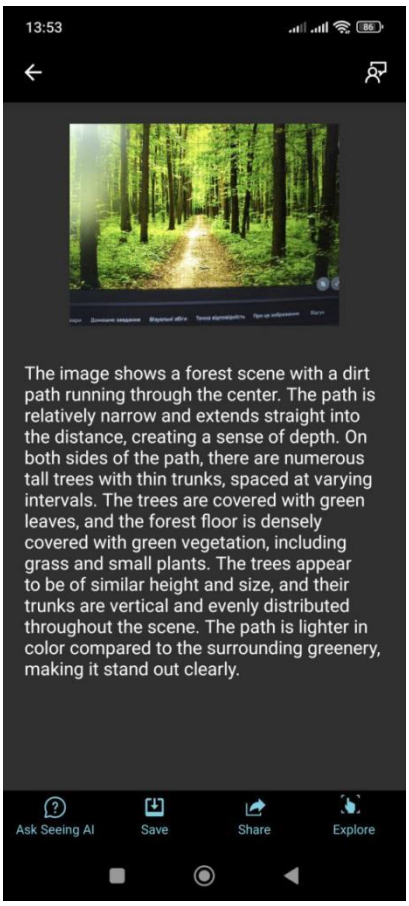

Рисунок 1.3 - Застосунок “Seeing AI”

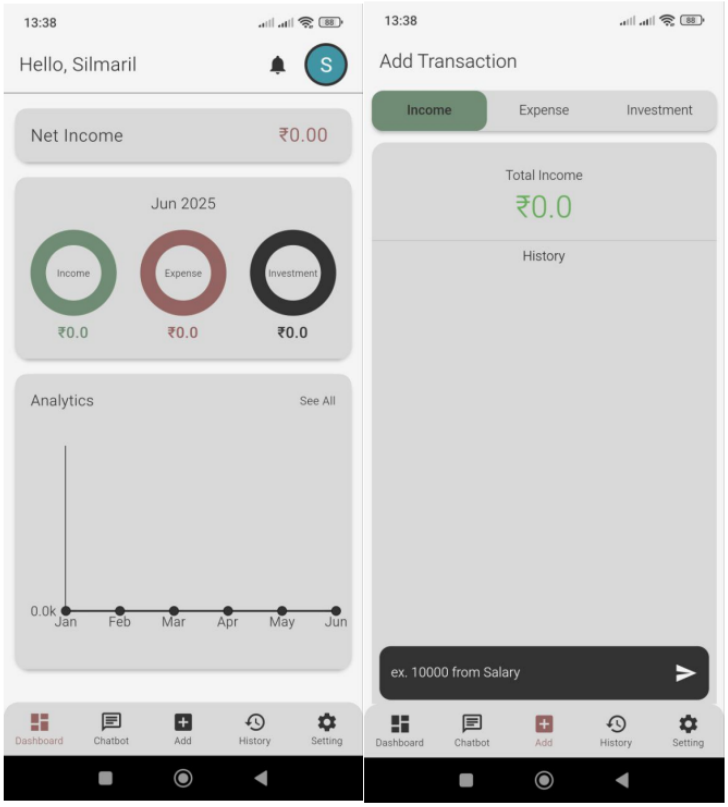

Рисунок 1.4 - Застосунок “FinTrack”

Таблиця 1.1 - Порівняльна таблиця застосунків на .NET MAUI

Застосунок	Financing Tracker App	Weather '21 (MAUI Weather)	Seeing AI	FinTrack
1	2	3	4	5
Призначення	Трекінг доходів і витрат	Показ погоди в реальному часі	Допомога людям із порушенням зору	Персональний фінансовий контроль
Функціональність	Додавання витрат, доходів, категорій, графіки	Пошук погоди за містом, GPS-локація, API інтеграція	Розпізнавання тексту, осіб, об’єктів, банкнот, кольорів	Кошторис, аналітика витрат, планування бюджету
Архітектура	MVVM, Dependency Injection	MVVM, API-запити	MVVM з використанням служб Microsoft Cognitive Services	MVVM, SQLite
Локальне збереження	Так (SQLite)	Ні	Часткове - для кешування результатів	Так (SQLite)
Інтеграція з API	Ні	Так (OpenWeather Map API)	Microsoft Azure AI (Computer Vision API, Face API)	Так (можлива інтеграція)

Продовження таблиці 1.1

1	2	3	4	5
Кросплатформність	Android, iOS, Windows	Android, iOS, Windows	Android, iOS,	Android, iOS
UI/UX	Простий, функціональний	Мінімалістичний, адаптивний	Простий інтерфейс, оптимізований для доступності	Дружній інтерфейс з графіками
Використані бібліотеки	SQLite-net, CommunityToolkit	HttpClient, JSON.NET	Microsoft.Maui, Azure.AI SDK, CommunityToolkit.Mvvm	SQLite, SkiaSharp
Складність реалізації	Середня	Низька-середня	Висока, через ШІ-функції та UX для незрячих	Середня

В таблиці 1.1 всі чотири застосунки демонструють можливості .NET MAUI як кросплатформного фреймворку, проте кожен з них має свою специфіку. Наприклад Seeing AI є дуже складним у виконанні через використання технології розпізнавання об’єктів, в свою чергу Financing Tracker App має базову фінансову функціональність і добре ілюструє реалізацію категорій витрат та їх візуалізацію. Weather '21 демонструє роботу з API, геолокацією та мережевими запитами, хоча не підтримує офлайн-режим. FinTrack підтримує роботу з локальною базою даних, планування, аналіз та візуалізацію фінансових даних.

Ці приклади можна використовувати як етапи зростання складності при вивченні розробки застосунків на .NET MAUI: від простого з визначенням місця розташування до високофункціонального застосунку з розпізнаванням тексту, людей , об’єктів і т.ін. Всі вони базуються на архітектурі MVVM та

використовують мову програмування C# із розміткою XAML, що робить їх корисними навчальними зразками.

Незважаючи на популярність фреймворку, зараз існує обмежена кількість масових застосунків, які повністю реалізовані на .NET MAUI через його відносну новизну. Більшість великих компаній використовують нативну розробку або фреймворки на кшталт Flutter чи React Native.

.NET MAUI активно розробляється як інструмент для створення повноцінних крос-платформних мобільних додатків. .NET MAUI демонструє високий потенціал для розробки нових додатків, зокрема, у сфері обміну візуальним контентом. Наявність прикладів функціональних MVP-застосунків доводить, що фреймворк є зручним для комерційного використання та подальшого масштабування.

1.4 Постановка завдання та визначення функціональних вимог

Мета цього проекту полягає в розробці крос-платформного мобільного застосунку соціальної мережі з використанням фреймворку .NET MAUI. Розроблений застосунок повинен надавати користувачам основні функції для соціальної взаємодії, включаючи реєстрацію, створення профілю, публікацію контенту та взаємодію з іншими користувачами. В рамках проекту передбачається вивчення інструментів для розробки інтерфейсу користувачів, роботи з базами даних, обміну повідомленнями через мережу та забезпечення безпеки даних користувачів. Застосунок має бути зручним у використанні, адаптивним до різних платформ та забезпечувати продуктивну взаємодію з користувачами.

Робота над додатком передбачає ознайомлення з архітектурою .NET MAUI, принципами крос-платформної розробки, організацією взаємодії між інтерфейсом користувача та бізнес-логікою, реалізацією локального або віддаленого зберігання даних.

Основні завдання проекту:

- Розробити кросплатформний застосунок на основі .NET MAUI.
- Реалізувати автентифікацію та авторизацію користувачів (реєстрація, вхід, вихід).
- Надати можливість завантаження та перегляду світлин.
- Реалізувати функціонал стрічки публікацій із підтримкою лайків.
- Забезпечити обмін повідомленнями між користувачами в чаті.
- Побудувати архітектуру застосунку за принципами MVVM.
- Зберігати дані в базі даних.

Для досягнення поставлених цілей необхідно визначити перелік функцій, які повинен надавати застосунок. Основні функціональні вимоги перераховані нижче, вони охоплюють ключові аспекти взаємодії користувача з системою, структуру інтерфейсу та обробку даних.

Функціональні вимоги:

а) Реєстрація та вхід

1) Форма реєстрації з введенням логіну, пароля.

2) Збереження облікового запису в базі даних.

3) Вхід з перевіркою облікових даних.

б)Профіль користувача

1) Перегляд власних публікацій.

2) Видалення або редагування власних фото.

3) Стрічка публікацій

в) Відображення фотографій інших користувачів у хронологічному порядку.

1) Можливість ставити лайк та дизлайк.

2) Перехід до профілю користувача за його публікацією.

г)Завантаження фото

1) Додавання фото з галереї або камери пристрою.

2) Додавання опису до фотографії.

3) Відображення завантажених фото у загальній стрічці та в профілі користувача.

г) Обмін повідомленнями

1) Чат між користувачами в реальному часі.

2) Відображення списку діалогів.

д) Адаптивність та продуктивність

1) Підтримка Android, iOS.

2) Адаптивний дизайн інтерфейсу для різних типів екранів.

3) Оптимізація обробки фото (стиснення, кешування).

Також можливо включити додаткові нефункціональні вимоги, які допомагають забезпечити якість і зручність використання застосунку. Інтерфейс програми має бути адаптивний, щоб коректно відображатися на різних мобільних пристроях та екранах. Продуктивність досягається за рахунок оптимізації завантаження фотографій та впровадження механізмів кешування, які зменшують час очікування. Масштабованість передбачає можливість інтеграції з хмарними платформами для зберігання даних. Зручність використання забезпечується інтуїтивно зрозумілим дизайном і логічною структурою навігації.

Використання .NET MAUI дозволить реалізувати один код для декількох платформ, знизивши витрати на розробку і підтримку, а також забезпечити високий рівень інтеграції з інструментами Microsoft, що є особливо актуальним для швидкої реалізації MVP та подальшого масштабування продукту.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПОБУДОВА ЛОГІЧНОЇ СТРУКТУРИ

2.1 Аналіз та вибір інструментальних засобів розробки

Для реалізації застосунку було обрано .NET Multi-platform App UI, оскільки .NET MAUI - крос-платформенний фреймворк для створення нативних мобільних і десктопних застосунків за допомогою C# і XAML. За допомогою .NET MAUI, можна створювати застосунки, які працюватимуть на Android, iOS, macOS та Windows використовуючи той самий код. Що дозволяє витрачати менше часу та ресурсів на розробку застосунку для кожної платформи. Схематично робота .NET MAUI (рисунок 2.1). [10]

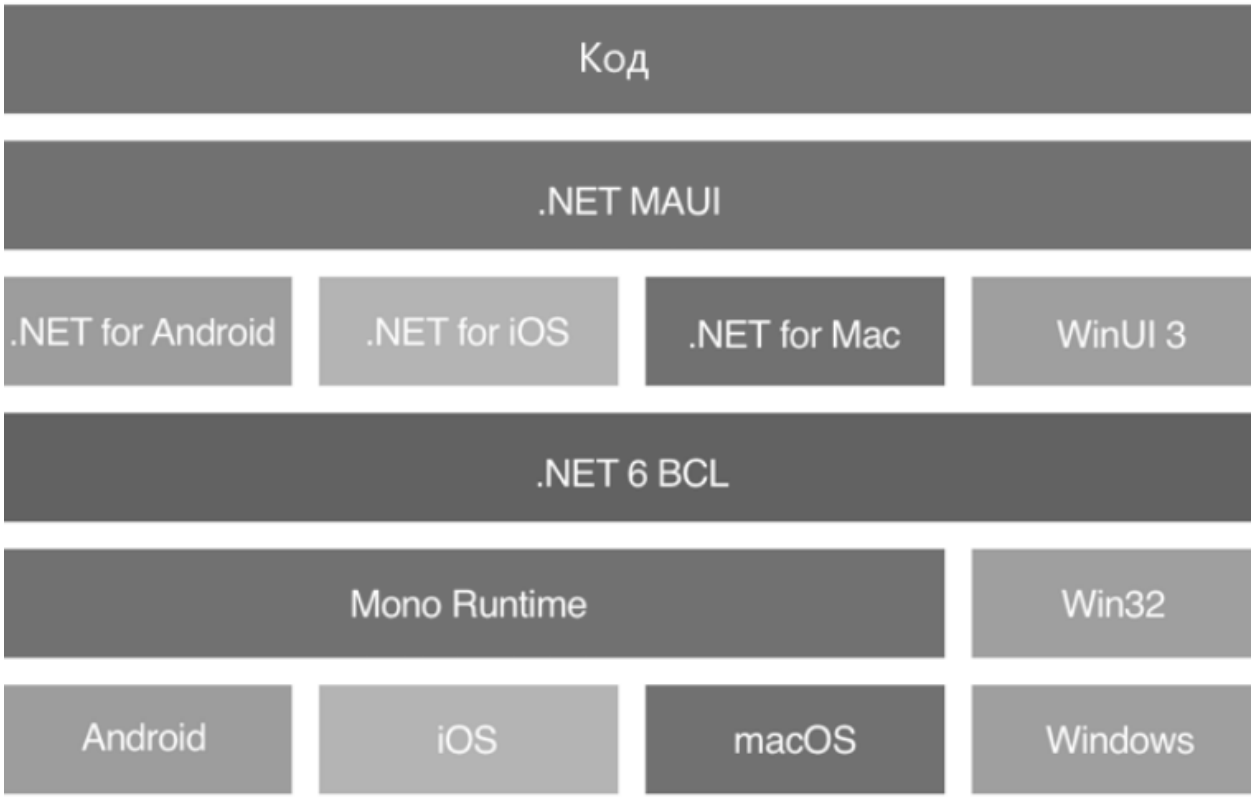

Рисунок 2.1 - Робота .NET MAUI

Розробка застосунків за допомогою .NET MAUI потребує правильного вибору інструментів, які впливатимуть на ефективність процесу розробки, продуктивність додатків, простоту обслуговування та підтримку майбутніх оновлень.

Для розробки застосунку за допомогою .NET MAUI потрібні такі основні інструменти: IDE:Visual Studio [11] , .NET SDK[12] та Android SDK[13], UI-бібліотеки, база даних MySQL[14].

Visual Studio - інтегроване середовище розробки IDE, створене компанією Microsoft для забезпечення розробникам можливості створювати, редагувати, налагоджувати, збирати та розгортати застосунки для різноманітних платформ. Це одне з найпотужніших і багатофункціональних IDE, яке підтримує широкий спектр мов програмування, включаючи: C#, VB.NET, C++, JavaScript, Python та інші. Visual Studio використовують в екосистемі .NET, зокрема, при розробці крос-платформних застосунків з використанням .NET MAUI. [11]

Visual Studio вбудований .NET Framework, який включає в себе велику бібліотеку і підтримує кілька мов програмування, які забезпечують функціональну сумісність мов . Бібліотека .NET доступна для всіх мов програмування. Програми, написані для .NET Framework, виконуються в програмному середовищі Common Language Runtime, це віртуальна машина додатків, яка надає тікі послуги як безпека, управління пам'яттю та обробка виключень. [12]

.NET Standard - стандартна специфікація інтерфейсів .NET API, які мають бути доступними у всіх реалізаціях .NET.[13]

Android SDK Platform-Tools - компоненти для Android SDK. Містить інструменти для взаємодії з платформою Android, насамперед adb та fastboot. Незважаючи на те, що adb необхідний для розробки Android-додатків, розробники зазвичай просто використовують інсталяції Copy Studio. Завантаження корисне, якщо використовувати adb безпосередньо з командного рядка і не має встановленої Studio. fastboot потрібен, якщо

розблокувати завантажувач пристрою і перепрошити його новим образом системи.[14]

MySQL - найпопулярніша система управління базами даних з відкритим вихідним кодом, яка широко використовується у веб-розробці, корпоративних додатках, фінансових системах, аналітиці. Багато користувачів вибирає MySQL через те, що БД може працювати на не потужному обладненні та не навантажує системні ресурси, також зберігає гнучкість і швидкість обробки запитів. Перевагою є швидке налаштування і MySQL працює так, як цього бажає більшість користувачів. [15]

При виборі інструментів для розробки застосунків з використанням .NET MAUI варто враховувати мету проекту, підтримувані платформи та вимоги до продуктивності, тому Visual Studio в поєднанні з офіційними SDK та підтримкою пакетів NuGet створює потужне середовище для швидкої та стабільної розробки.

2.2 Обґрунтування вибору мови програмування та середовища розробки

Під час розробки кросплатформених застосунків потрібно правильно обрати технології: мови програмування, середовища розробки та інтерфейсної розмітки. У даному випадку платформа .NET MAUI та мови програмування C#, XAML і Visual Studio утворюють масштабовану та ефективну інфраструктуру для створення мобільних та десктопних застосунків для Android, iOS, Windows та macOS з єдиною кодовою базою.

Мова C# - високорівнева, типізована, об’єктно-орієнтована мова програмування, розроблена компанією Microsoft спеціально для платформи .NET. Це дуже гнучка та потужна мова програмування і має схожість з мовами програмування C++ та Java, маючи запозичені або покращені функції, що надаються цими мовами. C# відсутні деякі засоби безпеки, які надаються іншими мовами програмування але це робить мову набагато гнучкішою. Мова C# має повну інтеграція з .NET та є основною мовою

програмування для .NET MAUI, що дає прямий доступ до API платформи, інтерфейсів, сервісів та шаблонів проєктів. [16]

XAML - мова розмітки, яка використовується для створення інтерфейсу користувачів у додатках на основі .NET MAUI, WPF та Xamarin. XAML розділяє логіку та інтерфейс. Структура інтерфейсу описується окремо від логіки, це дає можливість підвищити читабельність та модульність коду. Підтримка патерну MVVM, гнучкість стилів та ресурсів, а саме використання стилів, шаблонів, ресурсів для повторного використання коду інтерфейсу. Також інтеграція з Visual Studio Designer дає можливість візуального редагування XAML інтерфейсу.[17]

Використання C# та XAML у поєднанні Visual Studio є зручним рішенням для створення сучасних, продуктивних, кросплатформних застосунків. Така комбінація забезпечує високу швидкість розробки, повторне використання коду, чисту архітектуру, гнучкий інтерфейс та широку інтеграцію з хмарними та локальними сервісами.

2.3 Переваги та особливості використання платформи .NET MAUI

В .NET MAUI є декілька переваг в порівняні з іншими фреймворк-платформами. Ключові переваги інтерфейсу .NET MAUI:

- Єдина кодова база, яка дозволяє писати один код на C# та XAML для всіх платформ: Android, iOS, Windows, macOS, а також скорочує час і вартість розробки, спрощує підтримку та оновлення проекту.

- Спільна бізнес-логіка та UI дає можливість використовувати специфічні для платформи функції через ін'єкцію залежностей або умови компіляції.

- Незалежність від платформи в .NET MAUI використовує нативні інтерфейси кожної платформи, через “Обробники” замість “Рендери”, як в Xamarin. Користувачі бачать інтерфейс, який повністю відповідає їхній ОС.

Вбудована підтримка шаблонів архітектури підтримується MVVM

з прив'язкою даних. Сумісний з .NET Community Toolkit, Dependency Injection, INotifyPropertyChanged, ICommand тощо.

- Hot Reload та Live Preview - швидке оновлення інтерфейсу та логіки без повної перекомпіляції, дозволяє змінювати XAML та бачити результат в реальному часі.

- Підтримка сучасних API, таких як геолокація, камера, Bluetooth, Push-повідомлення, безпечне зберігання, локальні повідомлення, все це реалізовано за допомогою .NET MAUI Essentials.

- Повна підтримка .NET MAUI в Visual Studio 2022 і вище, візуальні інструменти, налагодження, профілювання, Live Preview.

Особливості використання .NET MAUI в розробці полягають у тому, що фреймворк має гнучку архітектуру, може підтримувати патерни MVVM, впровадження залежностей та Clean Architecture, а також платформоспецифічний код за допомогою умовної компіляції та розширень. .NET MAUI підходить для створення як простих мобільних додатків, так і складних корпоративних рішень, оскільки має легку інтеграцію з хмарними сервісами (Azure, REST API, SignalR). Ще однією перевагою є те, що основним середовищем є Visual Studio 2022+, яка надає всі необхідні інструменти, такі як шаблони проектів MAUI, емулятори Android/iOS, налагодження, профілювання та XAML-дизайнер.[18]

.NET MAUI можна використовувати для створення кросплатформних мобільних застосунків: бізнес-застосунки, маркетинг, електронна комерція, або для розробки десктопних застосунків з однаковим інтерфейсом для Windows і macOS. Тому для розробки мобільного застосунку було використано .NET MAUI, в платформі поєднуються повторне використання коду, висока продуктивність, гнучкий інтерфейс через XAML, повний доступ до API платформи та потужна екосистема .NET.

2.4 Архітектура застосунку та побудова логічної і функціональної моделей

Для розробки мобільного застосунку використано архітектурний шаблон MVVM, який є рекомендованим для .NET MAUI. За допомогою MVVM-архітектури можливо розділити логіку інтерфейсу та бізнес-логіки, зручно тестувати окремі компоненти, повторно використовувати ViewModel між платформами та ефективно підтримувати зв’язування даних між UI та логікою.[5]

Логічна структура застосунку включає основні функціональні блоки:

а) Авторизація та реєстрація має компоненти для створення акаунту, логіну, зберігання токенів

б) Рекомендації відображають пости інших користувачів та лайк постів.

в) Додавання світлин включає в себе можливість вибрати фото з галереї, додавання допису, завантаження на сервер.

г) Повідомлення та чат реалізує обмін повідомленнями для реального часу

ґ) Служби даних відповідають за HTTP-запити до API, обмін фотографіями та текстом

Логічна модель застосунку складається з таких основних сутностей (рисунок 2.2):

- User (Користувач): id, Username, PasswordHash.
- Post (Публікація): id, username, text , image_path, created_at, like_count, dislike_count.
- Message (Повідомлення): id, sender, receiver, content, timestamp.
- Votes (Вподобання): id, post_id, username, vote_type.

Між цими сутностями існують такі зв’язки: один користувач може мати багато постів, може надсилати та отримувати багато повідомлень та бачити пости інших користувачів. Один пост може мати лайк або дизлайк.

Функціональна модель (рисунок 2.3) складається з таких блоків:

- Запуск застосунку, завантаження усіх компонентів.

- Система перевіряє чи авторизований користувач.

- Якщо користувач не авторизований, відображається кнопка увійти та перенаправляє на сторінку реєстрації/входу, де користувач може зареєструватися або увійти.

- Головна сторінка, де відображаються пости користувача в профілі з датою створення та дописом до посту, які він опублікував і функцією видалення посту.

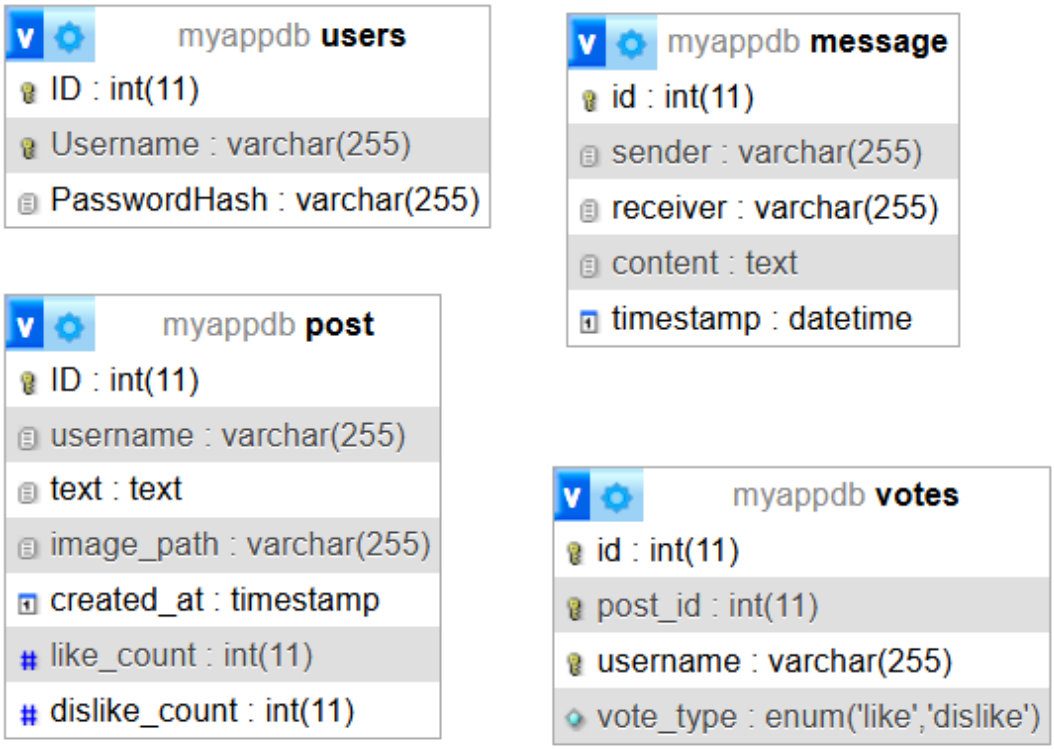

Рисунок 2.2 - Сутності

- При натисненні користувачем на кнопку “Додати пост”, користувача перенаправляє до галереї.

- Користувач може додати фото з галереї, також додати опис до фотографії.

- Користувачеві пропонується перейти на сторінку рекомендації.

- Сторінка рекомендації, де відображаються пости інших користувачів;

користувач може ставити лайк або дизлайк під кожним постом.

- Користувач може натиснути на кнопку з іменем іншого користувача і перейти на його сторінку.

- Сторінка іншого користувачв відображає його пости, з можливістю ставити вподабання до фото.

- Користувач може натиснути на кнопку ”Повідомлення” та написати власнику поточної сторінки.

- Відображається чат з вибраним користувачем.

- Список інших користувачів з якими авторизований користувач може спілкуватися.

- Вибір перейти на сторінку налаштування.

- Якщо користувач обере вийти, то завершується сесія користувача і у нього відображається сторінка реєстрації.

Побудова архітектури застосунку, а також створення логічної та функціональної моделей є критично важливим етапом життєвого циклу програмного забезпечення. Вони забезпечують не лише структурну ясність, але й дозволяють ефективно управляти розробкою, тестуванням, супроводом та масштабуванням системи.

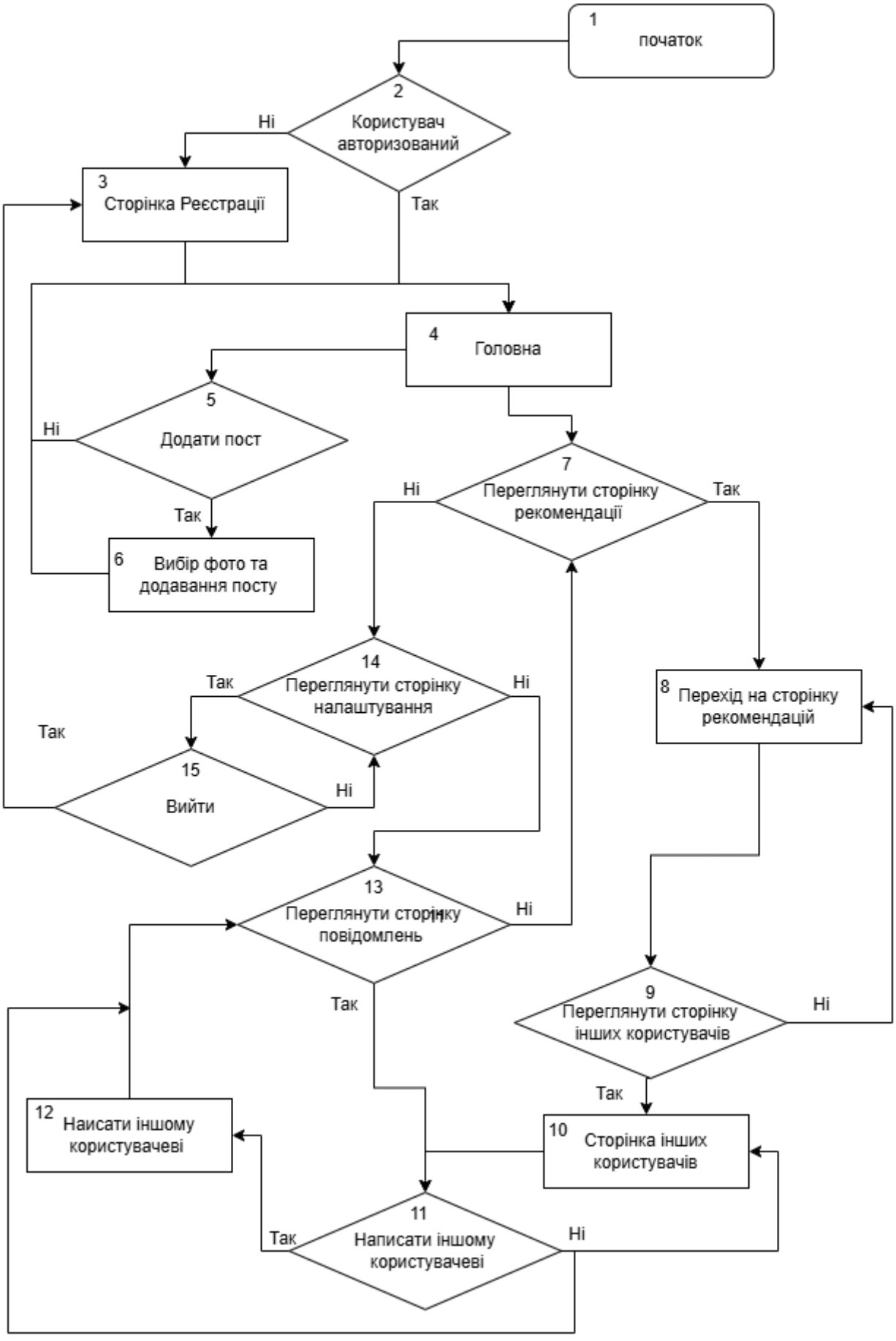

Рисунок 2.3 - Функціональна модель застосунку

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Програмна реалізація застосунку за допомогою .NET MAUI

Застосунок було створено з використанням архітектурного шаблону MVVM, що дало можливість розділити код на програмну частину та інтерфейс користувача. Для збереження даних користувачів та інформацію про пости було використано базу даних MySQL. В ході виконання проекту була створена структура проекту (рисунок 3.1)

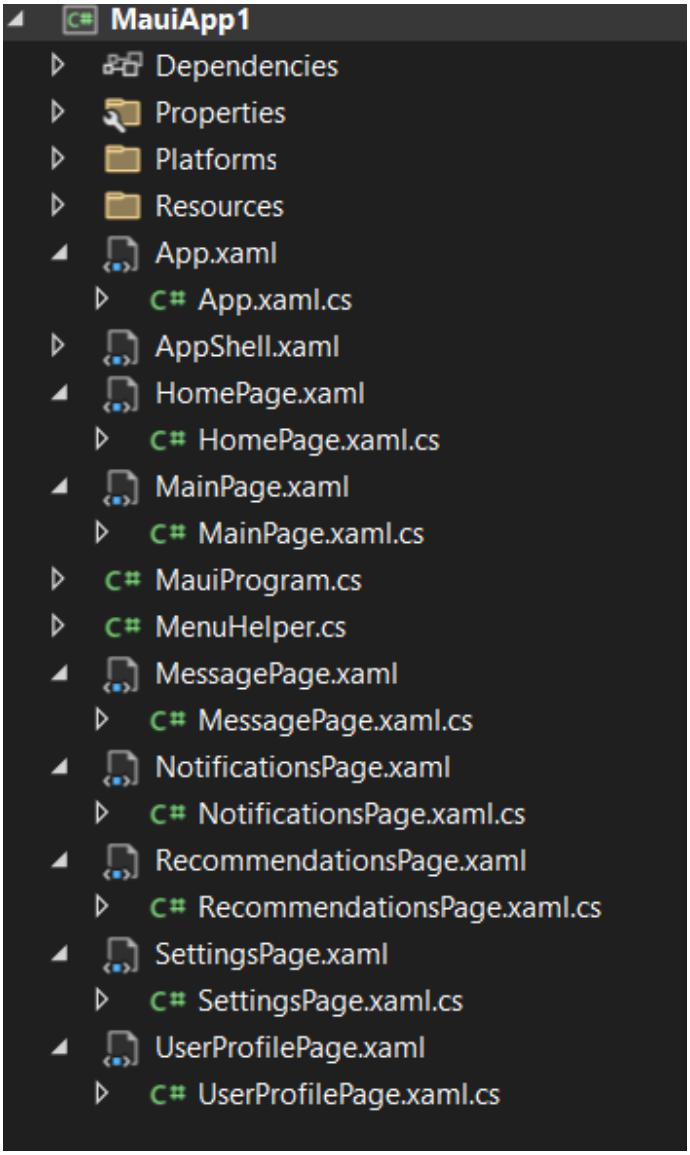

Рисунок 3.1 - Структура проекту

Для реалізації реєстрації користувача (рисунок 3.2) спочатку відбувається зчитування ім’я користувача та паролю з текстових полів, потім відбувається хешування паролю, що дає змогу захистити пароль від зловмисників. За допомогою бібліотеки BCrypt.Net, яка генерує хеш із сольовим значенням автоматично, це допомагає створити додатковий рівень безпеки, а алгоритм BCrypt дозволяє забезпечити високу стійкість до атак. Коли новий користувач реєструється, відбувається з’єднання до бази даних і додавання нового користувача. Якщо користувач вже існує тоді відображається повідомлення про помилку.

```
private async void OnRegisterClicked(object sender, EventArgs e)
{
    var username = usernameEntry.Text;
    var password = passwordEntry.Text;
    var passwordHash = BCrypt.Net.BCrypt.HashPassword(password);

    using (var connection = new MySqlConnection(connectionString))
    {
        await connection.OpenAsync();
        var query = "INSERT INTO users (Username, PasswordHash) VALUES (@username, @passwordHash)";
        using (var command = new MySqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@username", username);
            command.Parameters.AddWithValue("@passwordHash", passwordHash);
            try
            {
                await command.ExecuteNonQueryAsync();
                await DisplayAlert("Успіх", "Реєстрація пройшла успішно!", "ОК");
            }
            catch (MySqlException ex) when (ex.Number == 1062)
            {
                await DisplayAlert("Помилка", "Користувач із таким ім'ям уже існує.", "ОК");
            }
        }
    }
}
```

Рисунок 3.2 - Реалізації методу реєстрації

Після того як користувач зареєструвався в системі, наступним кроком є авторизація користувача. З поля вводу зчитуєтуються дані логіну і паролю, при підключенні до бази даних виконується запит, щоб отримати збережений хеш пароля. За допомогою BCrypt.Verify, перевіряється відповідність введеного пароля до хешу. Якщо дані всі правильні, відбувається перехід на головну сторінку. Зображено реалізації методу авторизації (рисунок 3.3).

Для додавання посту реалізована функція, яка перенаправляє

користувача до галереї, де користувач вибирає зображення з пристрою. Після вибору фотографії, користувачу пропонується зробити допис до фото. Дані про фото та допис зберігаються в базу даних. Після збереження посту відбувається оновлення списку постів (рисунок 3.4).

```
private async void OnLoginClicked(object sender, EventArgs e)
{
    var username = usernameEntry.Text;
    var password = passwordEntry.Text;

    using (var connection = new MySqlConnection(connectionString))
    {
        await connection.OpenAsync();
        var query = "SELECT PasswordHash FROM users WHERE Username = @username";
        using (var command = new MySqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@username", username);
            var passwordHash = (string)await command.ExecuteScalarAsync();
            if (passwordHash != null && BCrypt.Net.BCrypt.Verify(password, passwordHash))
            {
                if (!string.IsNullOrEmpty(username))
                {
                    await Navigation.PushAsync(new HomePage(username));
                }
            }
            else
            {
                await DisplayAlert("Помилка", "Неправильний логін або пароль", "ОК");
            }
        }
    }
}
```

Рисунок 3.3 - Реалізації методу авторизації

```
1 reference
private async Task AddPostAsync()
{
    var result = await MediaPicker.PickPhotoAsync();
    if (result != null)
    {
        string selectedImagePath = result.FullPath;
        string text = await DisplayPromptAsync("Новий пост", "Введіть текст посту:");

        if (!string.IsNullOrEmpty(text))
        {
            SavePostToDatabase(text, selectedImagePath);
            await DisplayAlert("Успіх", "Ваш пост було додано!", "ОК");
            PostsListView.ItemsSource = GetPosts();
        }
        else
        {
            await DisplayAlert("Помилка", "Текст посту не може бути порожнім.", "ОК");
        }
    }
    else
    {
        await DisplayAlert("Помилка", "Зображення не вибрано.", "ОК");
    }
}
```

Рисунок 3.4 - Завантаження фото

Щоб зберегти нове повідомлення між поточним користувачем і іншим співрозмовником з бази даних MySQL (рисунок 3.5). Спершу відбувається з'єднання з MySQL, потім зберігається нове повідомлення в таблицю message та автоматично додає поточний час.

```
private void SaveMessage(string content)
{
    try
    {
        using var conn = new MySqlConnection(connectionString);
        conn.Open();

        string query = "INSERT INTO message (sender, receiver, content, timestamp) VALUES (@sender, @receiver, @content, NOW())";
        using var cmd = new MySqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@sender", CurrentUser);
        cmd.Parameters.AddWithValue("@receiver", OtherUser);
        cmd.Parameters.AddWithValue("@content", content);
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Помилка при збереженні повідомлення: {ex.Message}");
    }
}
```

Рисунок 3.5 - Збереження повідомлень в базу данних

3.2 - Взаємодія користувача з інтерфейсом застосунку

Важливим для застосунку є те щоб для користувача був зручний та зрозумілий інтерфейс застосунку. Нижче наведений загальний сценарій взаємодії користувача з інтерфейсом застосунку.

Сторінка авторизації та реєстрації складається з поля воду логіна, паролю, і кнопок "Реєстраця", Вхід". Якщо користувач вже зареєстрований, то буде відображатися повідомлення, що користувач вже зареєстрований. При вводі неправильного паролю буде відображатися повідомлення, що пароль некоректний (рисунок 3.6).

Якщо користувач натисне на кнопку "Додати пост", буде відображена галерея пристрою, де потрібно вибрати фото за бажанням (рисунок 3.7). Після вибору фото користувачеві буде запропоновано додати допис до фото. При умові не вибору фото або не додавання допису до фото відобразиться повідомлення, що пост не буде додано.

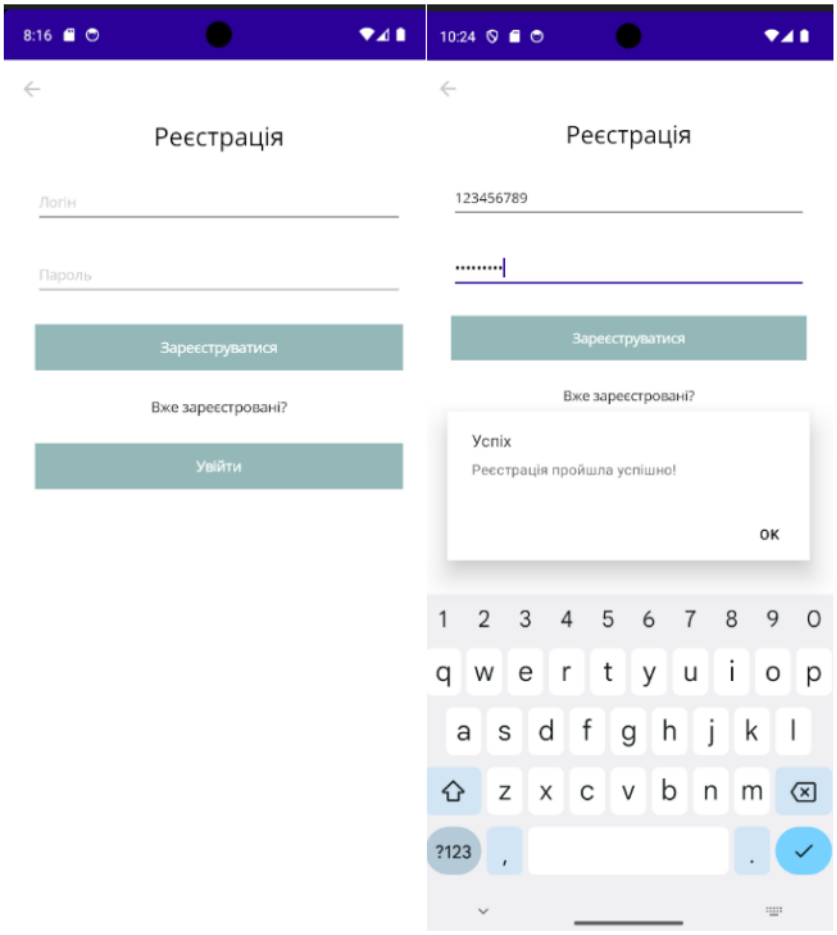

Рисунок 3.6 - Реєстрація користуача

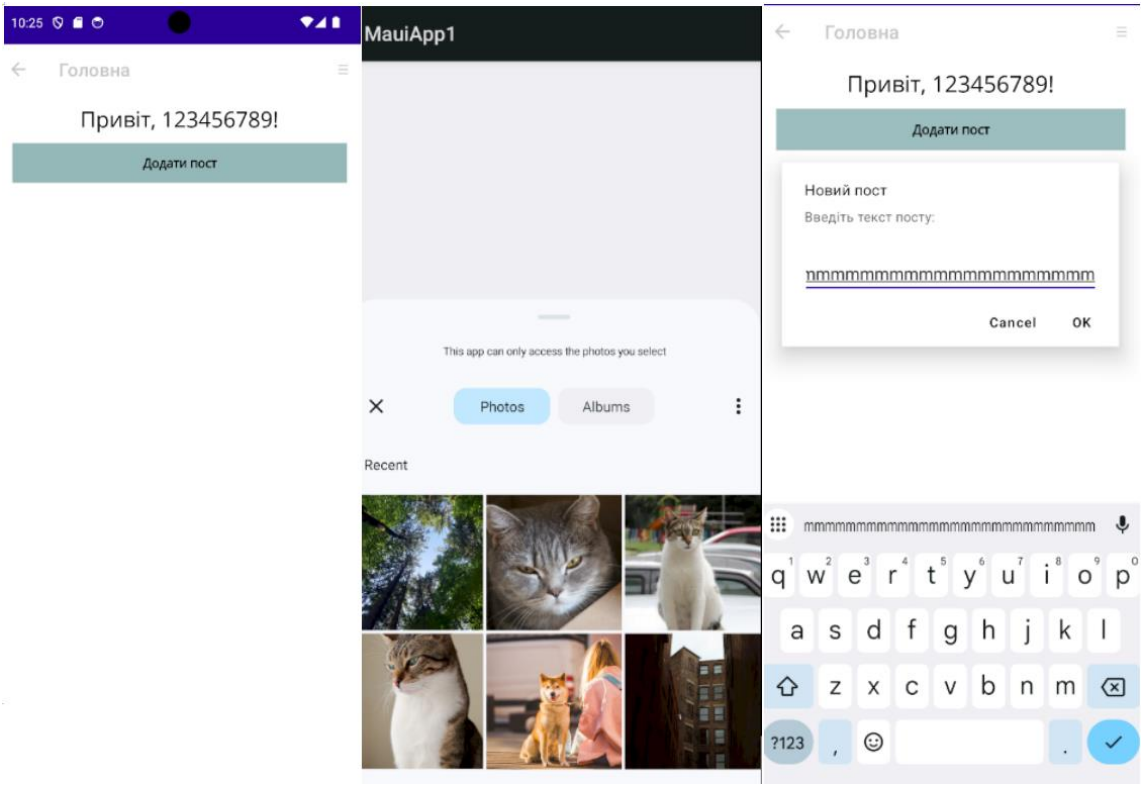

Рисунок 3.7 - Вибір фото та додавання опису до фото

При успішному додаванні посту, новостворений пост буде відображено в стрічці користувача з датою створення. Користувач зможе переглянути власні пости і при необхідності видалити небажані пости. Вигляд стрічки користувача (рисунок 3.8).

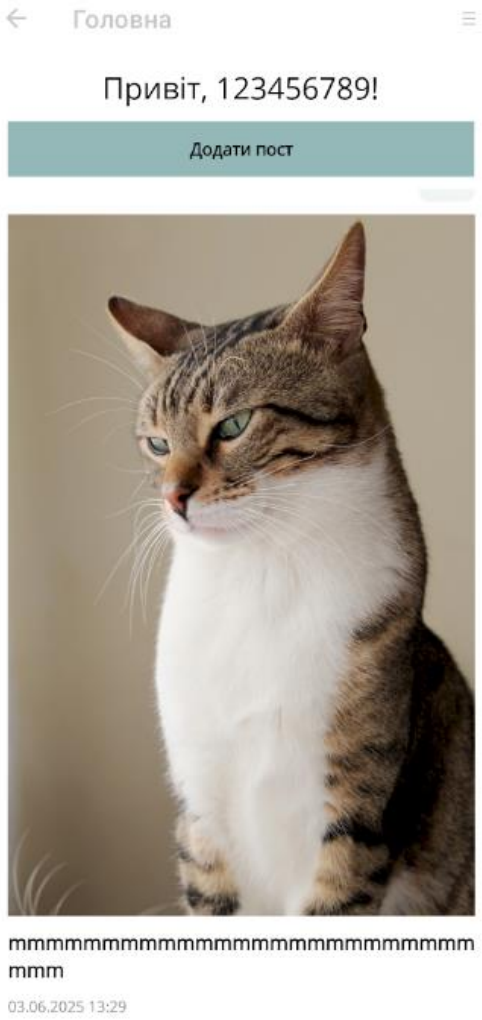

Рисунок 3.8 - Відображення посту у стрічці

Щоб викликати меню потрібно натиснути на кнопку у верхньому правому кутку, де буде відображатися меню. В залежності від того, на якій сторінці знаходиться користувач, будуть відображатися кнопки переходу на інші сторінки. Наприклад, меню на головній сторінці буде відображати такі кнопки як “Рекомендації”, “Сповіщення”, “Налаштування”(рисунок. 3.9).

Рисунок 3.9 - Меню

На сторінці рекомендації відображаються пости інших користувачів. Під кожним постом авторизований користувач має змогу проголосувати, чи подобається це зображення, чи ні. Один користувач під кожним постом може поставити тільки один відгук: або подобається або не подобається. Додатково при натисканні кнопки з “ім’ям користувача”, відобразиться профіль вибраного користувача, де є можливість також ставити відмітку стосовно вподабання будь-якого зображення (рисунок. 3.10).

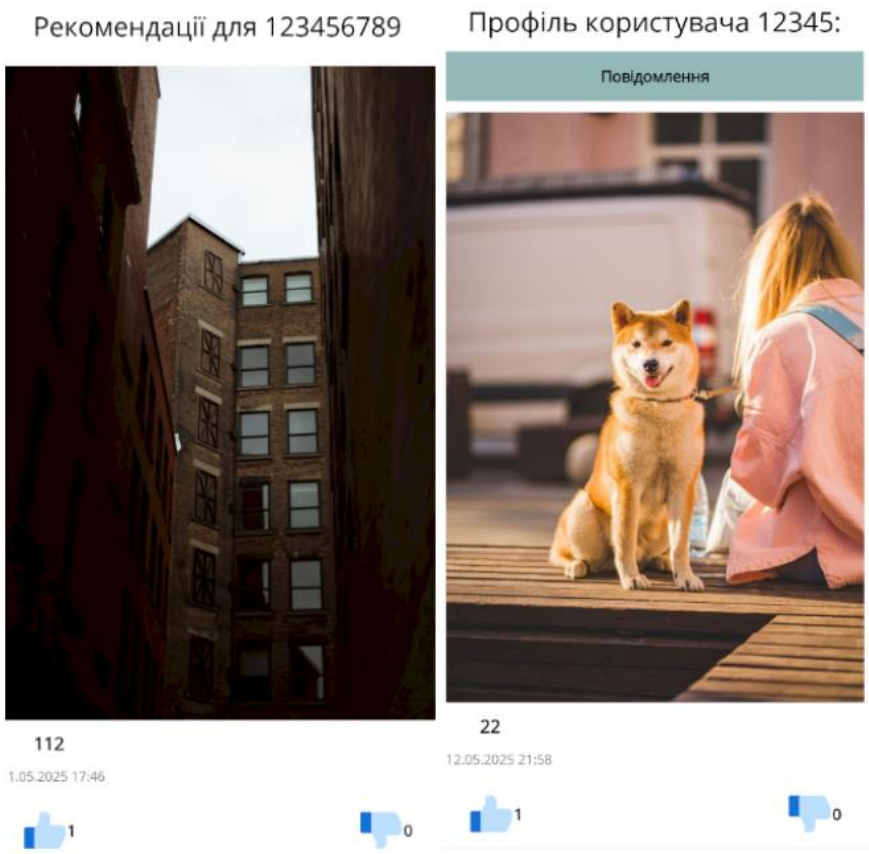

Рисунок 3.10 - Взаємодія з іншими користувачами

Сторінка сповіщення відображає список користувачів з якими спілкується авторизований користувач. При виборі користувача відкривається чат з цим користувачем (рисунок. 3.11).

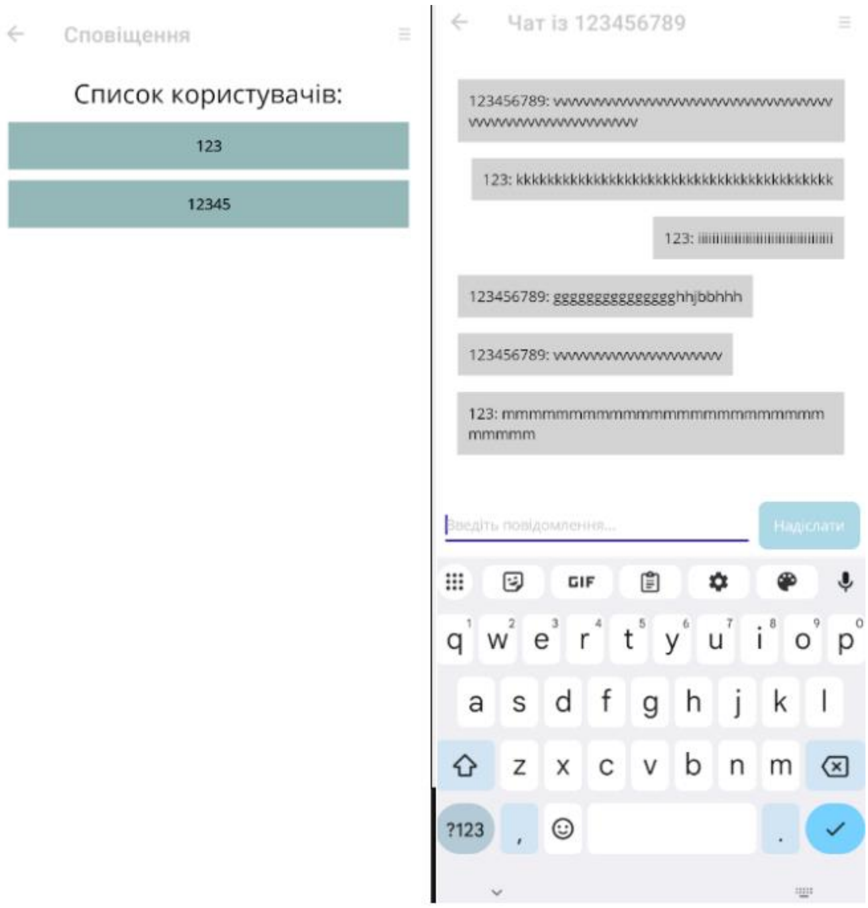

Рисунок 3.11 - Чат з іншими користувачами

Сторінка налаштування має кнопку “Вихід” яка завершує сесію користувача (рисунок 3.12).

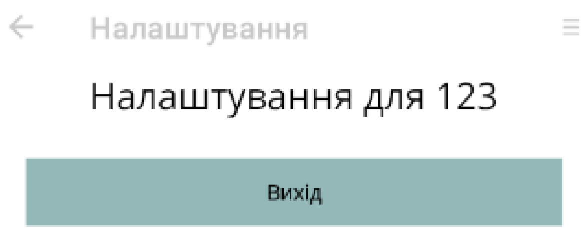

Рисунок 3.12 - Сторінка налаштування

3.3 Тестування функціональності та інтерфейсу застосунку

Основною метою тестування є виявлення критичних багів та завчасне виправлення, перед тим як опублікувати застосунок. Тестування буде проводитися до того моменту, коли виправлять критичні та явні баги. Якщо користувач виявить значну помилку в застосунку, тоді компанія може мати наслідки, тому тестування важливе для подальшого використання застосунку.

Щоб перевірити відповідність функціональних вимог до поставленого завдання та його фактичних характеристик, використовують тестування функціональності. За допомогою функціонального тестування розробник перевіряє щоб програмний продукт мав увесь необхідний функціонал.

Тестування включає такі етапи:

а) Реєстрація нового користувача. Перевірка відображення повідомлення, якщо логін співпадає з зареєстрованим користувачем. В результаті тестування було отримано повідомлення, що вже є користувач зареєстрований з таким логіном .

б) Вхід користувача з неправильним логіном та паролем. Було проведено тестування, де вводились не коректні дані. При введені помилкового паролю відображалось повідомленя, що неравильний логін або пароль. Тестування пройдене успішно.

в) Перевірка завантаження фото. Для перевірки коректності роботи натискаємо кнопку “Додати пост”. Було запропановано вибрати фото з галереї, після натиснення на фото пропонувалося додати допис до фото. Потім обновилася сторінка та відобразився пост в профілі користувача. Якщо не було вибране фото, відображалося сповіщеня, що зображення не було вибране. Або якщо не було додано допис, тоді буде повідомлення, що текст посту не може бути порожнім. Результатом тестування було коректне завантаженя посту. Також, якщо не було вибрано зображеня та допис до фото, відображалась помилка.

г) Відображення постів на сторінці користувача, рекомендації та профілі інших користувачів. Під час тестуваня було проведено перевірку на відображеня постів. На головній сторінці відображались тільки пости авторизованого користувача, сторінка рекомендацій відображала пости усіх користувачів, сторінка інших користувачів відображала пости іншого користувача. Результат тестування виконано успішно.

ґ) Перевірка відправки та отримання повідомлення. Результатом перевірки було коректна та швидка відправка повідомлення, і отримання цього повідомлення. Результат тестування виконано успішно.

д) Збільшення або зменшення кількості лайків. Задачею тестування було відслідкувати, щоб користувач під кожним постом міг вибрати або подобається або неподобається. Один користувач має тілки один голос для кожного посту. В результаті тестування було виконано успішно перевірку вибору вподобань користувачем.

Таким чином проведене функціональне тестування дає змогу відслідкувати коректність роботи застосунку.

Тестування інтерфейсу користувача дає змогу перевірити коректність та зручність його роботи з застосунком для користувача. Важливо щоб при тестуванні інтерфейсу всі його елементи правильно відповідали до технічного завдання та були зрозумілими, коректно відображалися на більшості мобільних пристроїв.

Для тестування інтерфейсу було використано Visual Studio Device Simulator. Перевіркою на кросплатформеність було використано такі мобільні пристрої Pixel 6 версія android 14. Тестування iOS недоступне, через відсутню можливість використовувати MacOS.

Логіка навігації працює коректно, усі кнопки ведуть до правильних сторінок, і виконують відповідні функції. Візуально та інтуїтивно зрозумілий інтерфейс застосунку.

3.4 Аналіз результатів

Результати тестуваня дають змогу детально зрозуміти де програма працює коректно, а де потрібно доопрацьовувати. Важливо, що використовуючи застосунок, користувач може швидко орієнтуватися в розділах. Продуктивність застосунку показала стабільну швидкодію при завантаженні фото, стрічки та під час спілкування між користувачами. Для захисту данних користувачів було застосовано функцію зберігання паролів хешовано з використанням солі. В майбутньому можлива масштабованість застосуноку, який побудований з урахуванням додавання нових функцій, таких як, завантаження відео, коментування або системи сповіщень.

ВИСНОВКИ

У ході виконання даної роботи розроблено мобільний застосунок з використанням .NET MAUI. Розроблено реєстрацію та авторизацію користувача, створення та видалення постів, можливість спілкування користувачів.

Застосунок надалі можливо доопрацювати, додавши швидке завантаження, завантаження відео, розширення системи повідомлень, впровадження гнучких налаштувань профілю користувача, додавання можливості коментування під публікаціями.

Було здійсненно порівнняння з іншими існуючими застосунками, також досліджено основні переваги та недоліки. Більшість застосункув з використанням .NET MAUI не має великої популярності, оскільки .NET MAUI є відносно новим фреймворком. Розроблений застосунок є перспективним як застосунок для спілкування між студентами в освітньому середовищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is .NET MAUI: A Comprehensive Guide URL: https://grialkit.com/blog/what-is-net-maui-a-comprehensive-guide - 14.06,2025.

2. .NET MAUI: The Future of Cross-Platform Development URL: https://medium.com/@BlueflameLabs/net-maui-the-future-of-cross-platform-development-0ca7a3461947 - 14.06,2025.

3. Alshamrani A., Bahattab A., Ayesh A. Taxonomy of Cross-Platform Mobile Applications Development Approaches / A. Alshamrani, A. Bahattab, A. Ayesh // Journal of Systems Architecture. - 2016. - Vol. 62, No. 6. - P. 83–105. - URL: https://www.sciencedirect.com/science/article/pii/S2090447915001276 - Заголовок з екрану - 14.06,2025.

4. Model-View-ViewModel (MVVM) URL: https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm - 14.06,2025.

5. MVVM Advantages URL: https://www.tutorialspoint.com/mvvm/mvvm_advantages.htm - 14.06,2025.

6. Трекер бюджету, облік витрат URL: https://play.google.com/store/apps/details?id=com.freeman.moneymanager - 14.06,2025.

7. .NET MAUI - Weather '21 URL: https://github.com/edinSahbaz/maui-weather/ - 14.06,2025.

8. Seeing AI URL: https://play.google.com/store/apps/details?id=com.microsoft.seeingai - 14.06,2025.

9. FinTrack URL: https://play.google.com/store/apps/details?id=com.sonu.fintrack&hl=uk - 14.06,2025.

10. Створення кросплатформних застосунків на .NET URL: https://abitap.com/1-1-stvorennya-krosplatformnyh-dodatkiv-na-net/ - 14.06,2025.

11. IDE: Visual Studio URL: https://visualstudio.microsoft.com/ -

14.06,2025.

12. Meijer E. Introduction to Visual Studio and C# / Erik Meijer. - [Місце видання не визначено] : [Видавництво не визначене], 2019. - 200 с..

13. .NET SDK URL: https://dotnet.microsoft.com/en-us/download/visual-studio-sdks - 14.06,2025.

14. Android SDK URL: https://developer.android.com/tools/releases/platform-tools - 14.06,2025.

15. Williams H., M.M. Seyed. Learning MySQL. - Beijing; Sebastopol : O’Reilly Media, 2007. - 826 p. - ISBN 978-0-596-00942-3.

16. Griffiths R. C# Programming Yellow Book / Rob Griffiths. - Hull : University of Hull, 2019. - 185 с.

17. XAML Syntax In Detail URL: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/advanced/xaml-syntax-in-detail - 14.06,2025.

18. What is .NET MAUI? URL: https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-9.0 - 14.06,2025.