МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ

«До захисту допущена»

Кафедра комп'ютерних наук та

прикладної математики

_____

« ___ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

Розробка мобільного додатку «Навігаційний асистент покупок»

Виконала: Щербатюк Ольга Василівна

студентка навчально-наукового інституту автоматики, кібернетики обчислювальної техніки

група КН-21інт

_____

Керівник: к.е.н., доцент Бачишина Л.Д.

_____

Рівне – 2023

ЗМІСТ

Національний університет водного господарства та природокористування

Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки
Кафедра **комп'ютерних наук та прикладної математики**
Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки та інформаційні технології»**

«ЗАТВЕРДЖУЮ»
Завідувач кафедри
_____
_____
"____" _____20__ року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТКИ

Щербатюк Ольги Василівни

1. Тема проєкту **Розробка мобільного додатку «Навігаційний асистент покупок»**
керівник проєкту *к.е.н., доцент Бачишина Лариса Дмитрівна*

затверджена наказом вищого навчального закладу від "19*" квітня 2023* року С-№449

2. Термін здачі студентом закінченої роботи *28.06.2023 р.*

3. Вихідні дані до проєкту: *інформація про існуючі додатки для навігації підчас шопінгу*

4. Зміст розрахунково-пояснювальної записки *у першому розділі здійснено аналіз технологій використання навігаційних карт, опис та порівняння. Другий розділ присвячений розробці дизайну додатку за допомогою Figma та реалізація інтерфейсу в Xamarin.Forms. Третій розділ містить опис функціональних особливостей додатку "навігаційний асистент покупок".*

_____

5. Перелік графічного матеріалу _____

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	*к. е. н, доцент, Бачишина Л. Д.*	*10.10.2022*	*10.10.2022*
2	*к. е. н, доцент, Бачишина Л. Д.*	*07.11.2022*	*07.11.2022*
3	*к. е. н, доцент, Бачишина Л. Д.*	*05.12.2022*	*05.12.2022*

7. Дата видачі завдання ____ _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	*Вивчення літератури за обраною тематикою*	*19.04.2023*	*виконала*
2	*Формулювання постановки задачі*	*20.04.2023*	*виконала*
3	*Зведення матеріалів до початку виконання роботи*	*21.04.2023*	*виконала*
4	*Проєктування структури додатку*	*24.04.2023*	*виконала*
5	*Створення інтерфейсу додатку*	*25.04.2023*	*виконала*
6	*Написання першого розділу*	*26.04.2023*	*виконала*
7	*Написання другого розділу*	*27.04.2023*	*виконала*
8	*Написання третього розділу*	*28.04.2023*	*виконала*
11	*Загальні висновки до роботи*	*04.05.2023*	*виконала*
12	*Підготовка звіту кваліфікаційної роботи*	*07.06.2023*	*виконала*
13	*Підготовка мультимедійної презентації*	*14.06.2023*	*виконала*
14	*Підготовка до виступу*	*15.06.2023*	*виконала*

Студентка _____ (_____)

Керівник кваліфікаційної роботи _____ (_____)

РЕФЕРАТ

Кваліфікаційна робота: 64 с., 8 малюнків, 12 додатків, 10 джерел.

Мета роботи:

- здійснити аналіз технологій використання навігаційних карт, опис та порівняння;

- провести розробку дизайну додатку за допомогою Figma та реалізація інтерфейсу в Xamarin.Forms;

- дослідити функціональні особливості додатку "Навігаційний асистент покупок";

- провести порівняльний аналіз СКБД, а в особливості SQLite та загальний огляд використаної бази даних;

- провести тестування та валідацію додатку.

Методи вивчення – вивчення теоретичних матеріалів по темі дослідження, створення моделі програмного додатка.

Створено програмну реалізацію мобільного додатку «Навігаційний асистент покупок». Організовано внутрішню базу даних та систему збереження проміжних та кінцевих результатів роботи програми. Налаштовано систему взаємодії користувача та сервісів навігації на основі Google Maps API.

Ключові слова: ***мобільний додаток, навігаційний асистент, навігаційні карти google maps, figma, Xamarin.Forms, sqlite***

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API — це абревіатура «Application Programming Interface «(інтерфейс програмування додатків, програмний інтерфейс програми)

SQL — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних.

СКБД (Система управління базами даних) — набір взаємопов'язаних даних і програм для доступу до цих даних.

CRUD — 4 основні функції управління даними «створення, читання, оновлення і вилучення».

ВСТУП

У сучасному світі мобільні додатки стали неодмінною частиною повсякденного життя, допомагаючи задовольняти потреби та вирішувати повсякденні завдання. А їх розробка є актуальною та широко поширеною галуззю програмування. Одним із ключових аспектів мобільних додатків є навігація та використання навігаційних карт. Вони відіграють важливу роль у забезпеченні користувачам зручного та ефективного способу орієнтації в просторі, знаходження необхідних місць та планування маршрутів. Однак, при виборі технології для використання навігаційних карт у мобільних додатках, дослідники та розробники стикаються з рядом технологічних викликів та варіантів вибору. Один з популярних сервісів для розробки мобільних додатків - Google Maps API.

Актуальність цього проєкту є високою, оскільки використання Google Maps API має великий потенціал у навігації, пошуку місць, геолокації та інших сферах. Розвиток цієї галузі та зростання популярності мобільних додатків роблять дослідження та розробку за допомогою Google Maps API значущими. У цій роботі будуть застосовані різні методи та підходи до вирішення поставлених завдань, такі як аналіз літературних джерел, порівняння технологій, проєктування інтерфейсу, розробка функціоналу, реалізація бази даних, тестування та валідація. Результати цієї роботи сприятимуть покращенню процесу навігації при здійсненні покупок та нададуть користувачам зручний та персоналізований досвід.

Завданням роботи є аналіз та порівняння популярних технологій використання навігаційних карт, таких як Google Maps, Mapbox та OpenStreetMap, з урахуванням їхньої функціональності, підтримки платформ та вартості, розробка дизайну додатку за допомогою інструменту Figma та реалізація інтерфейсу використовуючи Xamarin.Forms, розробка функціоналу та особливостей додатку "навігаційний асистент покупок", який включатиме можливість пошуку магазинів, навігацію до них, планування маршрутів та збереження улюблених місць, розробка бази даних та управління даними з

використанням СКБД SQLite для збереження інформації про магазини, користувачів та налаштування додатку, а також тестування та валідація додатку з метою перевірки його функціональності, стабільності та зручного користування.

Метою цієї роботи є розробка функціонального та ефективного мобільного додатку з використанням Google Maps API.

Ця робота присвячена розробці мобільного додатку з використанням Google Maps API і є актуальною у контексті зростаючої популярності мобільних додатків та потреби у функціональних та ефективних рішеннях для навігації та геолокації.

РОЗДІЛ 1. АНАЛІЗ ТЕХНОЛОГІЙ ВИКОРИСТАННЯ НАВІГАЦІЙНИХ КАРТ

Система глобального позиціювання все більше знаходить своє місце у повсякденному житті. Цифрові навігаційні карти та сервіси навігації є невіддільною частиною сучасного технологічного прогресу, які мають велике значення в сфері транспорту, туризму та навігації загалом. Вони забезпечують користувачів актуальною інформацією про географічні обставини, шляхи, об'єкти та інші показники, які необхідні для успішної навігації в просторі.

Цифрові навігаційні карти є можуть бути використані на різних електронних пристроях, таких як смартфони, планшети, навігаційні системи у транспортних засобах тощо. Цифрові карти зазвичай мають різні рівні деталізації, що дозволяє користувачам вибирати необхідний рівень деталей залежно від їхніх потреб.

Навігаційні карти надають користувачам інформацію про географічне розташування об'єктів, шляхів, рельєфу, природних об'єктів та інших деталей, що можуть бути важливими для їхньої орієнтації. Вони включають різноманітні елементи, такі як дороги, річки, озера, гори, будівлі, мости, аеропорти та багато іншого. Крім того, на навігаційних картах можуть бути позначені координати, шкали, символи та кольори, що полегшують розуміння та інтерпретацію інформації.

Навігаційні карти допомагають користувачам в орієнтації, навігації та знаходженні місць, надаючи їм засоби для планування маршрутів і визначення оптимальних шляхів до певних місць. Користувачі можуть скористатися навігаційними картами для виявлення свого поточного місцеперебування, визначення напрямку руху, вибору альтернативних шляхів або оцінки відстаней. Також навігаційні карти дозволяють знаходити цікаві об'єкти, такі як ресторани, готелі, визначні пам'ятки, бензоколонки тощо, що робить їх корисними для туристів та мандрівників.

1.1. Популярні технології навігаційних карт

1.1.1 Google Maps

Google Maps є однією з найпопулярніших платформ для навігаційних карт, що надає широкий спектр функціональних можливостей. Цей сервіс, розроблений компанією Google, став невіддільною складовою сучасного світу, де мобільні додатки з підтримкою навігації стали частиною нашого повсякденного життя.

Однією з ключових переваг Google Maps є його потужне API, яке надає розробникам доступ до різноманітних функцій та даних, що стосуються картографії, маршрутизації та локаційної інформації. API Google Maps дозволяє інтегрувати ці функції у мобільні додатки, розширюючи їхні можливості та забезпечуючи зручну навігацію для користувачів.

Однією з ключових функціональних можливостей Google Maps є можливість відображення картографічних даних у режимі реального часу. Користувачі можуть переглядати актуальну інформацію про дорожні умови, трафік та транспортні засоби на мапі. Це дозволяє планувати оптимальний маршрут з урахуванням поточної ситуації на дорозі.

Крім того, Google Maps надає можливість побудови оптимального маршруту від однієї локації до іншої. Вона враховує різні параметри, такі як довжина маршруту, тривалість подорожі та наявність пробок. Користувачі можуть отримувати детальні інструкції про кожний поворот та зручно переміщатись по маршруту за допомогою голосових підказок.

Однією з головних переваг Google Maps є легкість використання його API. Інтерфейс розробника має простий та зрозумілий набір для програмування, що дозволяє швидко інтегрувати картографічні дані у свої проєкти. Крім того, API має велику кількість документації та підтримку спільноти розробників, що дозволяє швидко розв’язувати будь-які проблеми, які можуть виникнути під час розробки.

Однак, використання Google Maps API пов'язане з деякими обмеженнями. Наприклад, використання API зазвичай пов'язане з певними обмеженнями на кількість запитів, які можуть бути зроблені до API протягом певного періоду часу. Крім того, Google може змінити умови використання API, що може вплинути на функціональність та доступність API для розробників.

Однією з основних функцій Google Maps API є маршрутизація, яка дозволяє розробникам створювати маршрути для користувачів між різними точками на карті. Ця функція дає змогу розробникам створювати спеціалізовані програми для планування маршрутів, такі як додатки для навігації або планувальники подорожей.

Крім того, API надає функцію знайомства з місцевістю та інформацією, пов'язаною з нею. Завдяки цій функції, користувачі можуть дізнатися про місцеві визначні місця, ресторани та інші місця, що можуть бути цікавими для відвідування.

API також має можливість додавання маркерів на карту, яка дозволяє розробникам позначати точки на карті, що мають певну важливість або цікавість для користувачів. Крім того, розробники можуть додавати зображення та відео до карти, що робить її більш інтерактивною та зручною для користувачів.

Застосування Google Maps API дозволяє створювати високопродуктивні та інтерактивні картографічні програми. Вона також дозволяє використовувати кращі практики управління даними, такі як індексування даних та зберігання місцеперебування у базі даних, що дозволяє забезпечити швидкий та надійний доступ до даних.

Узагалі, Google Maps API є потужним та ефективним інструментом для розробки програм та вебсайтів, пов'язаних з картографією та геопросторовими даними. Вона надає доступ до великої кількості геопросторових даних, таких як картографічні зображення, зображення з супутників, поверхні терену, інформації про транспортну доступність та іншого.

Одним з переваг Google Maps API є підтримка різних мов програмування, включаючи JavaScript, Python, Ruby та інших. Це дозволяє розробникам використовувати API на різних платформах та в різних середовищах.

Google Maps API також має можливість інтеграції з іншими сервісами Google, такими як Google Places та Google Street View. Це дозволяє розробникам додавати додаткові функції та можливості до своїх програм та веб-сайтів.

Однак, використання Google Maps API може бути обмежене залежно від тарифного плану, який вибере розробник. Безплатний план надає обмежені можливості, такі як обмежена кількість запитів та обмежена кількість картографічних зображень. Платні плани надають більші можливості, включаючи безліч запитів та більші обсяги даних.

Узагалі, Google Maps API є важливим інструментом для розробників, які працюють з геопросторовими даними та картографією. Вона дозволяє створювати високопродуктивні та інтерактивні програми, що надають користувачам широкі можливості взаємодії з геопросторовими даними.

1.1.2 Mapbox

Mapbox є однією з популярних платформ для цифрових навігаційних карт, яка надає розширені можливості та вражає своїми характеристиками. Ця платформа відрізняється декількома особливостями, зокрема зручністю роботи з API, можливістю створення власних стилів карт та використання власних географічних даних.

Перш за все, Mapbox пропонує потужне API, яке дозволяє розробникам легко інтегрувати картографічні функції у свої додатки. Це API забезпечує доступ до різних сервісів, таких як геокодування (перетворення адрес на географічні координати) та маршрутизація (розрахунок оптимального маршруту між точками). Розробникам надається можливість налаштування вигляду карт, додавання шарів з кастомними даними та використання різних векторних та растрових мап.

Другою важливою перевагою Mapbox є можливість створення власних стилів карт. Користувачі можуть налаштовувати кольори, шрифти, значки та інші елементи карти для досягнення потрібного візуального враження. Це дозволяє

створювати унікальний дизайн карт, який відповідає конкретним вимогам і бренду користувача.

Третя особливість Mapbox полягає в можливості використання власних географічних даних. Користувачі можуть завантажувати свої дані та використовувати їх на карті, додавати власні шари та об'єкти. Це дозволяє створювати спеціалізовані карти з урахуванням унікальних вимог та потреб користувача.

У порівнянні з Google Maps, Mapbox пропонує схожий функціонал, але зокрема відрізняється своїми можливостями налаштування стилів карт та використання власних географічних даних. Вартість використання обох платформ може варіюватися в залежності від масштабу проєкту та обсягу використовуваних ресурсів.

Що стосується інтеграції в додаток, обидві платформи надають зручні інструменти для вбудовування карт. Вибір між Mapbox та Google Maps залежить від конкретних вимог та потреб додатка. Для деяких проєктів може бути вигідніше використовувати Mapbox, зокрема якщо необхідні спеціалізовані стилі карт або власні географічні дані.

Узагалі, Mapbox є цікавою альтернативою до Google Maps, забезпечуючи широкі можливості для створення інтерактивних цифрових навігаційних карт зі зручним API, вибором стилів та використанням власних географічних даних.

1.1.3 OpenStreetMap

OpenStreetMap (OSM) є географічною базою даних, яка розвивається спільнотою відкритих джерел. Вона надає безплатний доступ до глобальних географічних даних та функцій навігації. OSM відрізняється від Google Maps та Mapbox у своїй природі, оскільки це колективний проєкт, що залучає внесок користувачів з усього світу для покращення географічної інформації.

Однією з головних переваг OpenStreetMap є його відкритий характер. Бувши колективним проєкт, OpenStreetMap надає можливість кожному користувачеві вносити зміни та виправлення до картографічних даних. Це

дозволяє актуалізувати та вдосконалювати інформацію про дороги, будівлі, місця інтересу та інші об'єкти на мапі.

Порівняно з Google Maps та Mapbox, OpenStreetMap має деякі особливості. По-перше, OpenStreetMap надає безплатний доступ до своїх картографічних даних та сервісів API. Це робить його привабливим варіантом для розробників, які шукають відкриті та вільні дані для використання у своїх проєктах.

По-друге, OpenStreetMap вирізняється своїм спільнотним підходом до розвитку та підтримки. Велика глобальна спільнота допомагає у вдосконаленні картографічних даних, а також надає підтримку та відповіді на питання користувачів. Цей підхід сприяє постійному вдосконаленню якості та актуальності даних OpenStreetMap.

Однак, порівнюючи OpenStreetMap з Google Maps та Mapbox, слід зазначити, що OSM може мати меншу кількість деталей та покриття у деяких регіонах, особливо в менш населених місцевостях. В той самий час, Google Maps та Mapbox володіють більшою кількістю ресурсів та можуть надавати більш детальні та широкі функціональні можливості.

Що стосується вартості, Google Maps та Mapbox надають платні плани для комерційного використання, в той час, як OpenStreetMap залишається безплатний для використання.

1.2. Особливості використання навігаційних карт

Одним з основних типів картографічних візуалізацій є растрові карти. Растрова карта являє собою зображення, побудоване на основі пікселів. Вона відображає географічну інформацію у вигляді растрового зображення, що дозволяє детально передати ландшафтні особливості та природні об'єкти. Однак, растрові карти мають обмежену масштабованість і не дозволяють виконувати динамічні зміни у відображенні даних.

Інший тип картографічної візуалізації – векторні карти, які базуються на математичних розрахунках і геометричних об'єктах. Вони зберігають географічну інформацію у вигляді векторних об'єктів, таких як точки, лінії та

полігони. Векторні карти мають перевагу у масштабованості, оскільки їх можна збільшувати або зменшувати без втрати чіткості. Крім того, вони дозволяють виконувати різні операції з геоданими, такі як пошук шляху, побудова маршруту тощо.

Для досягнення інтерактивності в мобільних додатках використовуються тривимірні моделі. Тривимірні моделі надають користувачеві можливість переглядати карту у тривимірному просторі, що дозволяє отримати більш реалістичне представлення географічних об'єктів та їх взаємного розташування.

У відображенні об'єктів на карті важливо враховувати їх семантику та інформаційну цінність. Позначки можуть використовуватись для виділення певних місць або об'єктів на карті. Шляхи, які з'єднують різні точки, можуть використовуватись для показу маршруту або найкоротшого шляху між двома місцями. Області можуть використовуватись для показу територій або географічних зон.

Особливості використання цих візуальних елементів для додатка "Навігаційний асистент покупок" полягають у можливості відображення магазинів, торгових центрів та інших місць, пов'язаних з покупками, на карті за допомогою позначок. Шляхи можуть використовуватись для побудови оптимального маршруту між різними магазинами обраними магазинами.

РОЗДІЛ 2: РОЗРОБКА ДИЗАЙНУ ДОДАТКА ЗА ДОПОМОГОЮ FIGMA ТА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ В XAMARIN.FORMS

2.1. Використання Figma для розробки дизайну додатка

2.1.1. Опис інструменту Figma

Figma – це векторний графічний редактор і інструмент для створення прототипів, призначений для використання в Інтернеті, з додатковою офлайн-функціональністю, доступною за допомогою настільних додатків для macOS і Windows. Додатки Figma Mirror для Android та iOS дозволяють переглядати прототипи Figma на мобільних пристроях. Набір функцій Figma зосереджений на використанні користувацького інтерфейсу та дизайні користувацького досвіду, з сильним акцентом на співпрацю в реальному часі.

Figma дозволяє швидко створювати та проєктувати інтерфейси. Платформа Figma пишається тим, що є інструментом для спільного проєктування, що дозволяє декільком користувачам працювати над проєктом одночасно. Це виявляється дуже ефективним, коли у формуванні результатів проєкту беруть участь багато зацікавлених сторін. Це ідеальний інструмент для живих проєктів, де розробники, копірайтери та дизайнери, наприклад, повинні працювати над чимось одночасно.

2.1.2. Процес розробки дизайну в Figma

Перед створенням мобільного додатка необхідно детально обдумати його концепцію, розібратися у його цілях та уявити, як він буде працювати. Цей процес покращується за допомогою розробки програмного прототипу. Візуалізація дозволяє зрозуміти, який результат має бути досягнутий.

Процес розробки прототипу мобільного додатку може бути спрощено до кількох етапів. По-перше, проводиться ряд консультацій та попередніх аналізів, щоб зібрати необхідну інформацію. Ці дані використовуються для визначення цінності та перспективи ідеї. Наступним кроком є створення першого ескізу та

розробка "скелету" додатку, який відображає візуальну частину та спосіб взаємодії з користувачем.

Далі ідея розширюється на основі зібраних даних. В результаті отримується два типи прототипів: інтерфейсний та функціональний. Інтерфейсний прототип показує, як користувач буде взаємодіяти з додатком та відображає властивості готового продукту. Функціональний прототип демонструє різні процеси, що задовольняють потреби клієнта, включаючи бізнес-процеси, результати запитів, логістику та інструменти комунікації.

За допомогою спеціального програмного забезпечення створюється робочий прототип. Його можна зробити статичним, презентуючи окремі сторінки програми з елементами дизайну, або інтерактивним, де користувач може взаємодіяти з клікабельними елементами. Варіант з інтерактивним прототипом найбільш дорогий, але надає можливість наочно оцінити взаємодію з додатком.

Цей процес прототипування мобільного додатку не закінчується тут. Прототип піддається тестуванню, щоб виявити можливі поліпшення або ж недоліки. На основі результатів тестування робляться редагування, і цей цикл повторюється до досягнення бажаного результату. Після цього можна розпочинати реалізацію та створювати функціональний додаток.

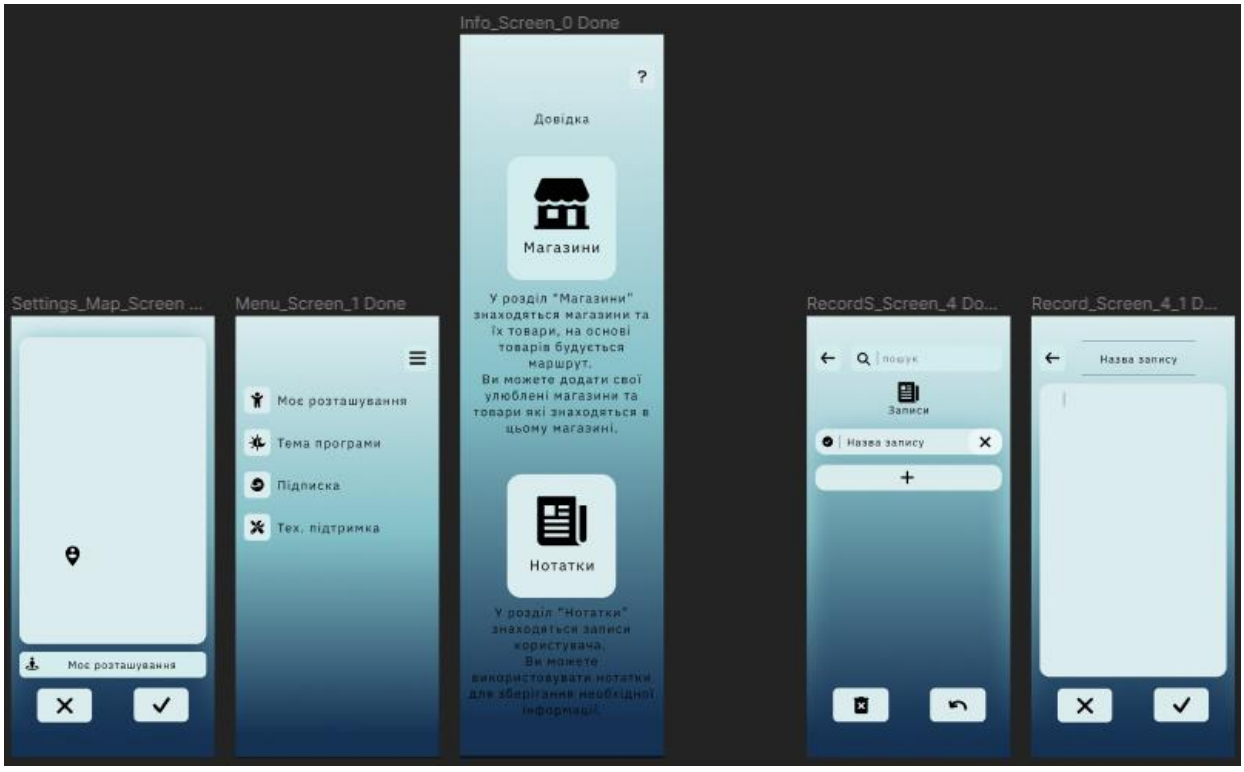

Рис. 2.1. Загальний вигляд створених фреймів

У проєкті в програмі Figma було створено такий перелік фреймів (загалом 23 фрейми):

а) Settings_Map Screen Done;

б) Info_Screen_0 Done;

в) Menu Screen 1 Done;

г) Main Screen Done;

д) LoyaltyCard_Screen_2 Done;

е) LoyaltyCard_Screen_2_1 Done;

ж)Shop_Screen_3 Done;

з) Shop_More_Menu Done;

и) Shop_Edit Screen Done;

к) Shop Edit Items_Screen Done;

л) Shop Add Screen Done;

м)Shop_Add_WatchHandAddItems_Screen_5_5 Done;

н) Shop_Add_Items_Screen Done;

о) RecordS_Screen_4 Done;

п) Record_Screen_4_1 Done;

р) Map SelectItem_Screen_5 Done;

с) Map SelectedItems_Screen_5_1 Done;

т) Map History_Screen_5_2 Done;

у) Map StartPoint_Screen_5_5 Done;

ф)Map Results_Screen_5_3 Done;

х) Map AskSave_Screen_5_4 Done;

ц) History_Screen_6 Done;

ч) History_Record_Screen_6 DONE.

Кожен з фреймів відповідає за один відображуваний екран мобільного додатку. Створення саме такої кількості фреймів зумовлено необхідністю виконання функціонала додатка відповідно до стандартів організації візуальних інтерфейсів у Android та iOS.

На рисунку 2.2 зображено структуру зв'язків між екранами створеного інтерактивного прототипу у Figma

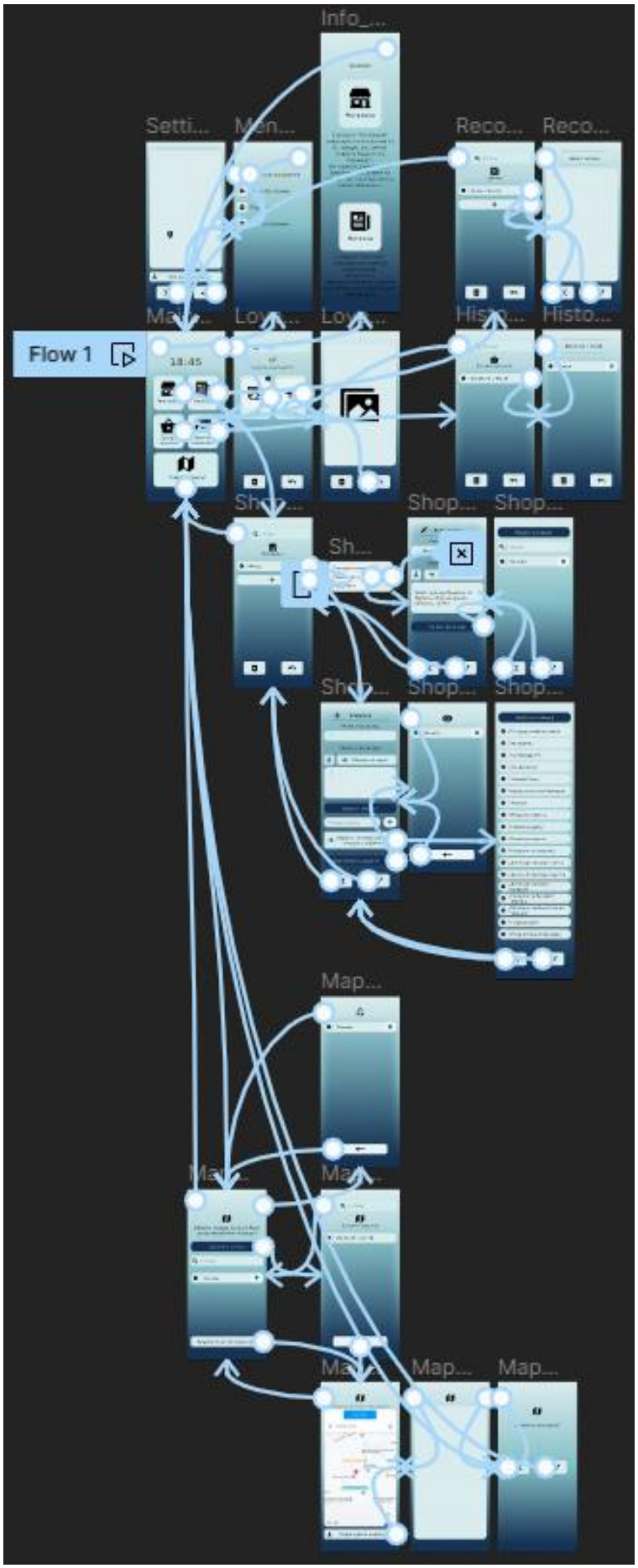

Рис. 2.2. Відображення зв'язків між екранами створеного інтерактивного прототипу у Figma

2.1.3. Експорт дизайну з Figma

Створення додатка за готовим дизайном є важкою та трудомісткою задачею, якщо використані ресурси та технології у дизайні не можуть співпрацювати з фреймворком розробки.

Для вирішення такого типу проблем зазвичай у розробці дизайну також беруть участь безпосередні програмісти додатка. Або ж, як було використано у цьому проєкті – було застосовано плагін-сервіс з експорту дизайну до формату фреймворку Xamarin. Нами було застосовано плагін Export Kit – Lightning Storm.

На рисунку 2.3 зображено розташування та запуск плагін конвертації інтерфейсу додатка. На рисунку 2.4 зображено інтерфейсу плагіну.

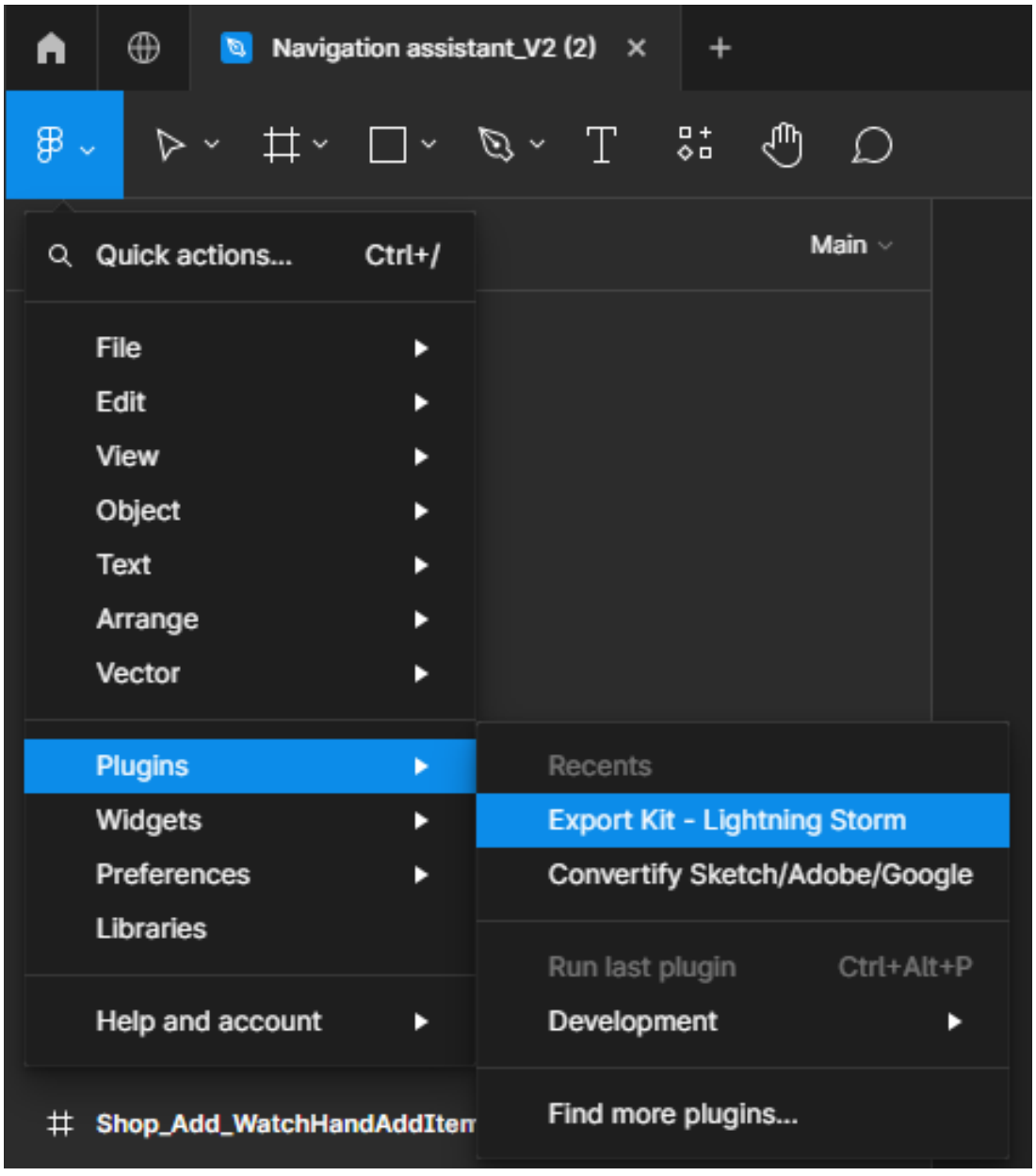

Рис. 2.3. Порядок доступу до плагіну конвертації дизайну

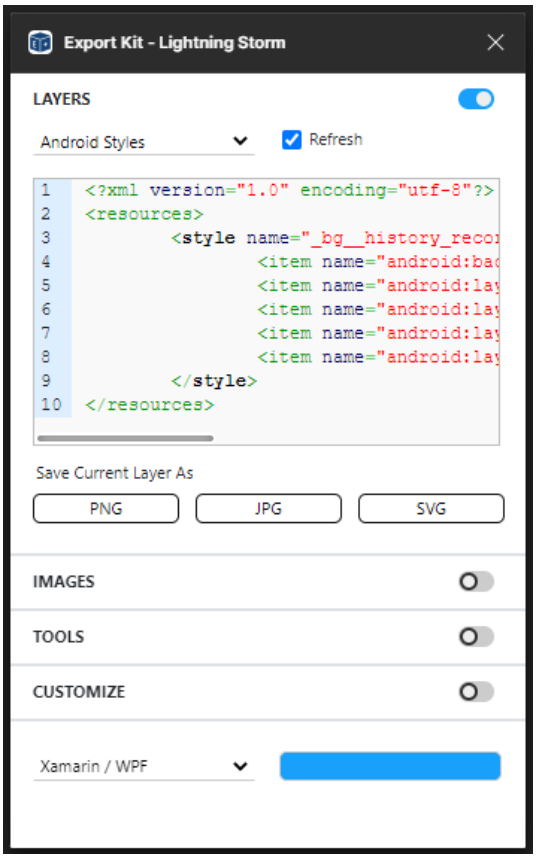

Рис. 2.4.Інтерфейс плагіну Export Kit – Lightning Storm

У результаті конвертації плагін пропонує завантажити та зберегти архів з набором файлів для розгортання дизайну у проєкті Visual Studio. Наповнення архіву зображено на рисунку 2.5.

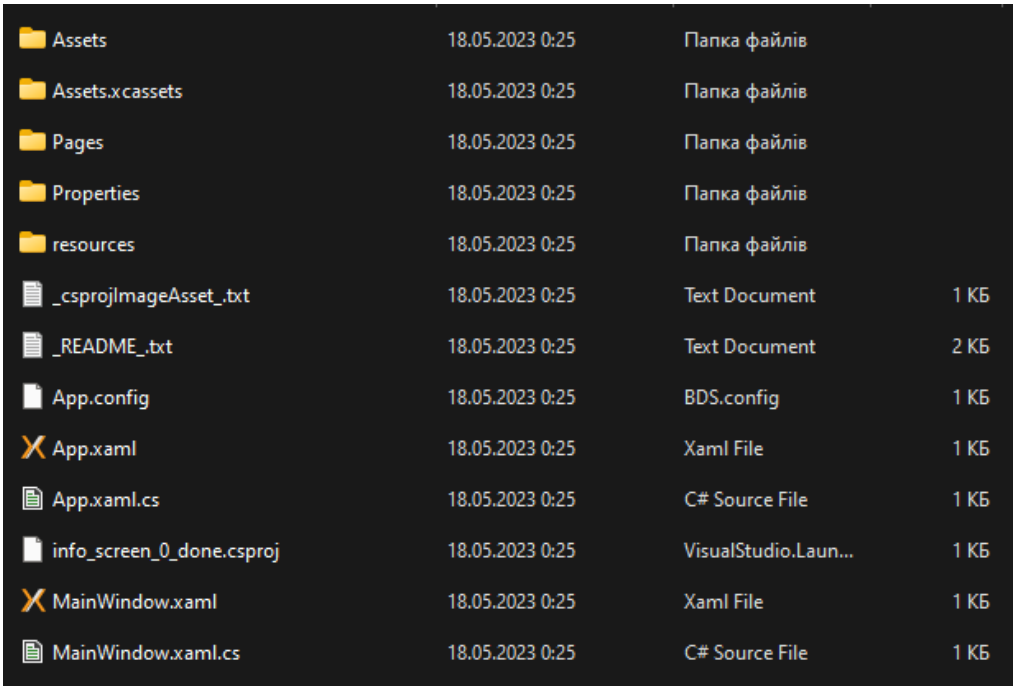

Рис. 2.5. Вміст архіву з експортованим дизайном

При цьому вид, налаштування та порядок відображення елементів перебувають у файлі формату XAML у теці Pages. Зображення та інші використані графічні елементи знаходяться у теці resources.

Реалізація програмної роботи графічних елементів інтерфейсу покладається на користувача, хоча файли для її організації збережено у файлі формату .cs.

2.2. Реалізація інтерфейсу в Xamarin.Forms

2.2.1. Огляд Xamarin.Forms

Xamarin.Forms - це багатоплатформовий інструментарій для користувача інтерфейсу для Android, iOS та Windows Presentation Foundation.

Xamarin.Forms включає повністю багатоплатформовий набір інструментів, який надає єдиний набір елементів керування для інтерфейсу користувача, макетів і сторінок, які належним чином зіставляються з відповідними рідними прив'язками для iOS, Android і Windows Phone. Оскільки Xamarin.Forms є новішою платформною бібліотекою, вона також має менше функціоналу. Кожен реліз наближає нас до повнофункціонального крос-платформного рішення, але іноді нам потрібно більше, ніж може запропонувати Xamarin.Forms "з коробки".

У таких випадках ми використовуємо платформозалежні бібліотеки для всієї сторінки або лише для її окремих частин, використовуючи PageRenderers – рендерери сторінок Xamarin.Forms. Підхід, орієнтований на конкретну платформу, є старішим, деталізованим і повнофункціональним. Він включає бібліотеки, які безпосередньо прив'язуються до специфічних переносних API: Xamarin.Android для Android і Xamarin.iOS для iOS. Для Windows Phone ми використовуємо Windows Phone SDK, рідне API, яке не вимагає прив'язки до Xamarin. Ці платформозалежні бібліотеки надають глибокий доступ до нативного інтерфейсу користувача, що створює візуально цікавий та легкодоступний користувацький досвід. Однак це має свою ціну: код, специфічний для платформи, потребує окремого проєкту інтерфейсу користувача

для кожної платформи з обмеженою кількістю коду, який можна повторно використовувати.

2.2.2. Розробка інтерфейсу в Xamarin.Forms

Розробка інтерфейсу є одним з ключових аспектів реалізації мобільних додатків у фреймворку Xamarin.Forms. Цей пункт розділу описує процес розробки інтерфейсу для нашого додатка "Навігаційний асистент покупок" з використанням Xamarin.Forms, який забезпечує написання платформо незалежного коду для різних мобільних платформ, таких як Android та iOS.

Основою розробки інтерфейсу у Xamarin.Forms є використання декларативної мови розмітки XAML (eXtensible Application Markup Language), що дозволяє описати інтерфейс додатка у структурованому форматі. За допомогою XAML можна визначити розміщення елементів інтерфейсу, їх зовнішній вигляд, обробку подій та зв'язування даних, що власне і було зроблено під час роботи з проєктом. Зв'язування даних дозволяє пов'язувати властивості елементів інтерфейсу з даними з програмної логіки додатку. Це забезпечує автоматичне оновлення інтерфейсу при зміні даних і спрощує роботу з передачею даних між компонентами додатка.

Загальна структура інтерфейсу додатка у вигляді коду XAML:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.MainPage">

    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="ms_wall_2"
                        Source="ms_wall_2.png"

RelativeLayout.WidthConstraint="{ConstraintExpression     Type=RelativeToParent,
Property=Width}"

RelativeLayout.HeightConstraint="{ConstraintExpression    Type=RelativeToParent,
Property=Height}"
                            Aspect="AspectFill"/>
                </RelativeLayout>
                <Label x:Name="ms_time"
                    FontFamily="IBM Plex Sans"
                    HorizontalTextAlignment="Center"
                    Opacity="0.9"
                    FontSize="50"
```

```
                        TextColor="#000000"
                        AbsoluteLayout.LayoutBounds="0.5,0.15,0.5,0.275"
                        AbsoluteLayout.LayoutFlags="All"
                        Text="{Binding CurrentTime}" />
                    <!--ІНФО-->
                    <ImageButton x:Name="ms_info_button"
                            Source="ms_info_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.95,0.01,0.15,0.09"
                            AbsoluteLayout.LayoutFlags="All"
Clicked="ms_info_button_Clicked"/>
                    <!--МЕНЮ-->
                    <ImageButton x:Name="ms_menu_button"
                            Source="ms_menu_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.05,0.01,0.15,0.09"
                            AbsoluteLayout.LayoutFlags="All"
Clicked="ms_menu_button_Clicked"/>
                    <!--МАГАЗИНИ-->
                    <ImageButton x:Name="ms_shop_button"
                            Source="ms_shop_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.15,0.3,0.39,0.23"
                            AbsoluteLayout.LayoutFlags="All"
Clicked="ms_shop_button_Clicked"/>
                    <!--НОТАТКИ-->
                    <ImageButton x:Name="ms_note_icon_button"
                            Source="ms_note_icon_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.85,0.3,0.39,0.23"

                            AbsoluteLayout.LayoutFlags="All"
                            Clicked="ms_note_icon_button_Clicked"/>
                    <!--ІСТОРІЯ ПОКУПОК-->
                    <ImageButton x:Name="ms_history_button"
                            Source="ms_history_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.15,0.65,0.39,0.23"
                            AbsoluteLayout.LayoutFlags="All" />
                    <!--КАРТА ЛОЯЛЬНОСТІ-->
                    <ImageButton x:Name="ms_loyaltycard_button"
                            Source="ms_loyaltycard_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.85,0.65,0.39,0.23"
                            AbsoluteLayout.LayoutFlags="All"
                            Clicked="ms_loyaltycard_button_Clicked"/>
                    <!--ЗНАЙТИ МАРШРУТ-->
                    <ImageButton x:Name="ms_map_button"
                            Source="ms_map_button.png"
                            BackgroundColor="Transparent"
                            AbsoluteLayout.LayoutBounds="0.5,1,0.834,0.23"
                            AbsoluteLayout.LayoutFlags="All"
                                 Clicked="ms_map_button_Clicked"/>
                </AbsoluteLayout>
            </ScrollView>
        </ContentPage.Content>
    </ContentPage>
```

2.2.3. Адаптація дизайну до різних пристроїв та платформ

Однією з великих труднощів під час створення мобільного додатка є розробка гнучкого дизайну. Це пов'язано з тим, що сучасні моделі смартфонів

мають різні розширення та співвідношення сторін екранів. Зазвичай розробники мають створювати окремі версії додатку, які відповідають певним параметрам, таким як розмір, операційна система та орієнтація. Проте Xamarin.Forms пропонує засоби адаптивного дизайну «з коробки». Наприклад, розміщення елементів на екрані пристрою динамічно змінюватиметься, якщо вони будуть описані за допомогою контейнера компонування <RelativeLayout>.

Усі контейнери компонування Xamarin є панелями, які успадковані від абстрактного класу Layout. Клас Layout пропонує численні панелі, кожна з яких по-своєму обслуговує внутрішні елементи. У випадку <RelativeLayout> – основною особливістю цього контейнера є можливість динамічного вирівнювання елементів Xamarin відносно будь-якої ширини та довжини екрана.

2.2.4. Інтеграція дизайну з функціональністю додатка

Під час інтеграції дизайну з функціональністю, виконуються наступні кроки:

а) розміщення елементів інтерфейсу: Згідно зі спроєктованим дизайном, елементи інтерфейсу, такі як кнопки, тексти, зображення тощо, розміщуються на відповідних сторінках додатку. Використовуються властивості розміщення та вирівнювання, щоб забезпечити правильну компоновку елементів на екрані.

б) встановлення стилів та ресурсів: Застосування стилів та ресурсів дозволяє надати єдність в зовнішньому вигляді додатку. Відповідно до розробленого дизайну, встановлюються відповідні стилі для елементів інтерфейсу, що забезпечують спільні властивості оформлення, такі як кольори, шрифти, розміри тощо.

в) обробка подій та зв'язування даних: Виконується програмна реалізація функціональних можливостей додатка, пов'язаних з елементами інтерфейсу. Це включає обробку подій, таких як натискання кнопок, введення даних у текстові поля тощо. Також встановлюються зв'язки між елементами інтерфейсу та даними з програмної логіки, щоб забезпечити коректну взаємодію між ними.

2.2.5. Тестування інтерфейсу

Завдяки вбудованим функціям відлагодження та тестування у Visual Studio Community, таким як – XAML Live Preview (зображено на рисунку 2.6) та XAML Hot Reload було проведено ручне тестування графічних та функціональних особливостей додатку.

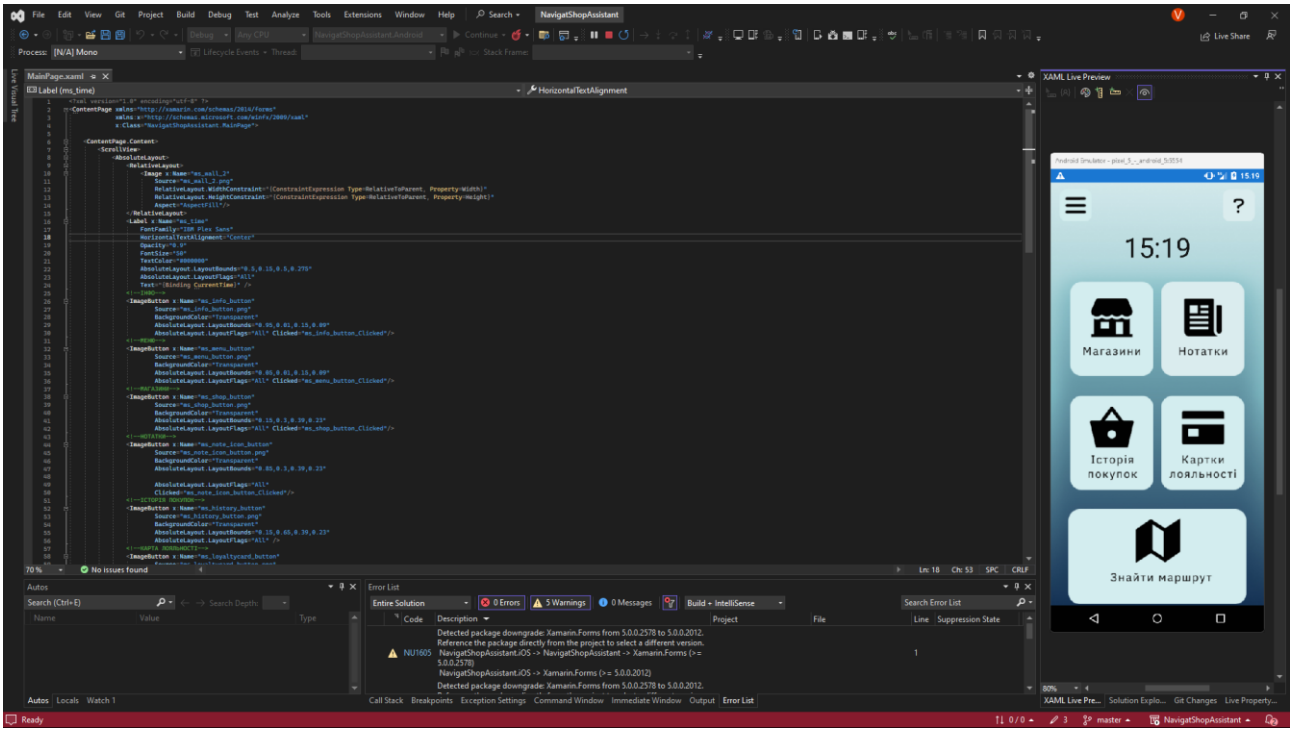

Рис. 2.6. Інтерфейс тестування XAML Live Preview

РОЗДІЛ 3. ФУНКЦІОНАЛ ТА ОСОБЛИВОСТІ ДОДАТКУ

3.1. Опис функціоналу додатку

Вимоги до мобільних додатків зростають разом зі збільшенням кількості користувачів смартфонів та планшетів. Кожен додаток повинен відповідати вимогам індустрії, щоб забезпечити високу якість користувачам та конкурентоспроможність на ринку. Розробка мобільного додатку з використанням Google Maps API не є винятком.

Перш за все, мобільний додаток повинен відповідати функціональним вимогам користувачів. Основна функція мобільного додатку з використанням Google Maps API - це надання користувачам інформації про локації та маршрути. Тому першою вимогою є забезпечення точної та швидкої роботи додатка. Крім того, мобільний додаток повинен мати інтуїтивний та зручний інтерфейс користувача, щоб користувачі могли з легкістю знайти необхідну інформацію.

Другою важливою вимогою є забезпечення безпеки та конфіденційності. Для цього мобільний додаток повинен відповідати вимогам безпеки в Інтернеті та мати захист від шкідливих програм та зловмисних атак. Крім того, мобільний додаток повинен забезпечувати конфіденційність даних користувачів та не збирати та не зберігати непотрібну інформацію.

Третьою вимогою є масштабованість та доступність мобільного додатку. Для цього мобільний додаток повинен бути оптимізований для роботи на різних пристроях та різних операційних системах. Крім того, мобільний додаток повинен бути доступний в різних регіонах та мовах, щоб забезпечити широке охоплення користувачів.

Четвертою вимогою є забезпечення високої продуктивності та швидкості роботи додатка. Для цього мобільний додаток повинен бути оптимізований для роботи на різних пристроях з різними характеристиками та на різних мережах з різною швидкістю підключення. Крім того, мобільний додаток повинен мати ефективне управління пам'яттю та енергоспоживанням, щоб забезпечити оптимальну роботу додатка.

П'ятою вимогою є забезпечення підтримки та оновлення мобільного додатку. Для цього мобільний додаток повинен мати можливість оновлення, які забезпечують підтримку нових функцій та виправлення помилок. Крім того, мобільний додаток повинен мати технічну підтримку та взаємодію з проблемами користувачів.

Шостою вимогою є забезпечення сумісності мобільного додатку з іншими додатками та сервісами. Для цього мобільний додаток повинен підтримувати інтеграцію з іншими додатками та сервісами, що використовуються користувачами, такими як соціальні мережі, електронні поштові сервіси, фото та відео хостинги та інші.

Узагальнюючи, розробка мобільного додатку з використанням Google Maps API потребує відповідності певним вимогам щодо функціональності, безпеки, масштабованості, продуктивності, підтримки та інтеграції. Враховуючи ці вимоги, розробники можуть забезпечити ефективну та надійну роботу додатка, який буде корисним для користувачів.

3.2. Особливості інтерфейсу та взаємодії з користувачем

У процесі розробки мобільного додатку важливо взаємодіяти з користувачами та забезпечити їхніми потребами, щоб забезпечити високу якість та задоволеність від користування додатком. Розробники повинні також забезпечувати безпеку даних, захист від зловмисних атак та забезпечувати високу швидкість та продуктивність додатка.

Крім того, важливо забезпечити сумісність мобільного додатку з різними платформами та пристроями, що використовуються користувачами. Для цього можуть використовуватися різні фреймворки та інструменти, що забезпечують оптимальну сумісність та роботу додатка на різних пристроях.

3.3. Інтеграція зовнішніх сервісів та API

Розглянемо покроково процес сервісу Google Maps API.

Крок 1: Реєстрація додатка в Google Cloud Console

Першим кроком до створення мобільного додатку з Google Maps API є реєстрація додатка в Google Cloud Console. Для цього потрібно мати обліковий запис Google та перейти на сторінку Google Cloud Console (https://console.cloud.google.com/). Після цього слід створити новий проєкт, який буде використовуватися для розробки додатка з Google Maps API. Після створення проєкту потрібно включити API для Google Maps та отримати ключ API.

Крок 2: Додавання Google Maps в мобільний додаток

Другим кроком є додавання Google Maps в мобільний додаток. Для цього можна використовувати Google Maps SDK для Android або Google Maps SDK для iOS, в залежності від того, на яку платформу розроблюється додаток. Наступним кроком є додавання ключа API до проєкту, щоб забезпечити авторизацію запитів до Google Maps.

Крок 3: Використання Google Maps API в мобільному додатку

Після додавання Google Maps до мобільного додатку можна почати використовувати Google Maps API для реалізації різних функцій. Наприклад, можна використовувати Google Maps API для відображення мапи, знаходження місцеперебування користувача, пошуку місць за адресою або назвою.

Крок 4: Відображення мапи

Для відображення мапи в мобільному додатку з використанням Google Maps API потрібно додати об’єкт MapFragment або SupportMapFragment до макета (layout) додатку. Після цього слід отримати об’єкт GoogleMap, який дозволяє керувати мапою.

Наприклад, у фрагменті можна додати код для отримання об'єкта GoogleMap наступним чином:

```
public class MapsFragment extends Fragment implements OnMapReadyCallback
{
private GoogleMap mMap;
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState)
{
```

```
View = inflater.inflate(R.layout.fragment_maps, container, false);
SupportMapFragment mapFragment = (SupportMapFragment)
getChildFragmentManager().findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
return view;
}

@Override
public void onMapReady(GoogleMap googleMap)
{
mMap = googleMap;

LatLng sydney = new LatLng(-34, 151);
mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in
Kiyv"));
mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
}
```

Крок 5: Знаходження місце перебування користувача

Для знаходження місце перебування користувача у мобільному додатку можна використовувати Google Location Services API. Цей API дозволяє отримати геолокацію користувача в режимі реального часу.

Для використання Google Location Services API спочатку потрібно додати залежність у файл build.gradle (Module: app):

```
dependencies
{
implementation 'com.google.android.gms:play-services-location :18.0.0'
}
```

Наступним кроком є отримання дозволу на доступ до місцеперебування користувача. Для цього можна використовувати клас ActivityCompat із пакета androidx.core.app.

```
if (ContextCompat.checkSelfPermission(getContext(),
Manifest.permission.ACCESS_FINE_LOCATION)
!= PackageManager.PERMISSION_GRANTED)
{
```

```
ActivityCompat.requestPermissions(getActivity(),
new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
MY_PERMISSIONS_REQUEST_LOCATION);
}
```

Якщо дозвіл на доступ до місцеперебування користувача було отримано, можна використовувати клас FusedLocationProviderClient для отримання місцеперебування користувача.

```
FusedLocationProviderClient fusedLocation
Client fusedLocationClient = LocationServices.getFusedLocationProviderClient(getActivity());
fusedLocationClient.getLastLocation().addOnSuccessListener(getActivity(), new
OnSuccessListener<Location>()
{
@Override
public void onSuccess(Location location) {
if (location != null) {
LatLng currentLocation = new LatLng(location.getLatitude(),
 location.getLongitude());
mMap.addMarker(new MarkerOptions().position(currentLocation).title("Marker in current location"));
mMap.moveCamera(CameraUpdateFactory.newLatLng(currentLocation));
}}});
```

Крок 6: Додавання маркерів на мапу

Для додавання маркерів на мапу можна використовувати клас MarkerOptions, який дозволяє налаштувати маркер, такі як позицію, заголовок, іконку та інше.

```
LatLng kyiv = new LatLng(-34, 151);
mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in
Sydney"));
```

Крок 7: Додавання маршруту на мапу

Для додавання маршруту на мапу потрібно використовувати клас PolylineOptions. Цей клас дозволяє задати точки маршруту та параметри лінії, такі як колір та ширина.

```
PolylineOptions polylineOptions = new PolylineOptions()
.add(new LatLng(37.7749, -122.4194))
.add(new LatLng(37.7749, -122.4304))
.add(new LatLng(37.7849, -122.4304))
.add(new LatLng(37.7849, -122.4194))
.add(new LatLng(37.7749, -122.4194))
.width(5)
.color(Color.RED);
mMap.addPolyline(polylineOptions);
```

Крок 8: Відображення інформації про місця на мапі

Для відображення інформації про місця на мапі можна використовувати клас InfoWindowAdapter. Цей клас дозволяє налаштувати вигляд вікна з інформацією про місце.

```
mMap.setInfoWindowAdapter(new GoogleMap.InfoWindowAdapter() {
@Override
public View getInfoWindow(Marker marker)
{
return null;
}

@Override
public View getInfoContents(Marker marker)
{
View view =
getLayoutInflater().inflate(R.layout.custom_info_window, null);
TextView tvTitle = view.findViewById(R.id.tv_title);
TextView tvSnippet = view.findViewById(R.id.tv_snippet);

tvTitle.setText(marker.getTitle());
tvSnippet.setText(marker.getSnippet());
return view;
}});
```

Крок 9: Обробка подій на мапі

Для обробки подій на мапі можна використовувати інтерфейс OnMapClickListener. Цей інтерфейс містить метод onMapClick, який викликається при натисканні на мапу.

```
mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener()
{
    @Override
    public void onMapClick(LatLng latLng)
{
      // Тут описується логіка мапи
}});
```

Крок 10: Отримання інформації про місця з Google Places API

Для отримання інформації про місця можна використовувати Google Places API. Цей API дозволяє отримувати інформацію про місця за їх назвою, адресою або координатами.

```
String placeId = "ChIJrTLr-GyuEmsRBfy61i59si0";

List<Place.Field> placeFields = Arrays.asList(Place.Field.NAME, Place.Field.LAT_LNG, Place.Field.ADDRESS);

FetchPlaceRequest request = FetchPlaceRequest.newInstance(placeId, placeFields);

placesClient.fetchPlace(request).addOnSuccessListener((response) -> {
    Place place = response.getPlace();
    Log.i(TAG, "Place found: " + place.getName());
}).addOnFailureListener((exception) -> {
    Log.e(TAG, "Place not found: " + exception.getMessage());
});
```

Слід зазначити, що для використання Google Places API необхідно налаштувати додаток у консолі розробника Google, де ви зможете отримати ключ API.

Крок 11: Отримання маршруту між двома точками з Google Directions API

Для отримання маршруту між двома точками можна використовувати Google Directions API. Цей API дозволяє отримати маршрут за координатами початкової та кінцевої точок, а також вказати параметри маршруту, такі як тип транспорту, режим руху та інше.

```
String origin = "37.785834,-122.406417";
String destination = "37.7749,-122.4194";
DirectionsApiRequest directions = DirectionsApi.newRequest(context)
      .mode(TravelMode.DRIVING)
      .origin(origin)
      .destination(destination);
directions.await();
DirectionsResult result = directions.await();
DirectionsRoute route = result.routes[0];
Log.i(TAG, "Route distance: " + route.legs[0].distance);
```

Крок 12: Розробка користувацького інтерфейсу додатка

Останнім кроком є розробка користувацького інтерфейсу додатка, який буде взаємодіяти з Google Maps API.

Для відображення карти використовується об'єкт MapView. Цей об'єкт можна додати в макет додатка, а потім ініціалізувати в коді активності.

```
<com.google.android.gms.maps.MapView
    android:id="@+id/map_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
MapView mapView = findViewById(R.id.map_view);
mapView.getMapAsync(new OnMapReadyCallback()
{
      @Override
      public void onMapReady(GoogleMap googleMap)
{
      // Logic to handle map ready event}});
```

3.4 Проєктування структури бази даних

Підчас проєктування структури необхідної бази даних для реалізації додатку "Навігаційний асистент покупок" було вирішено виділити такі сутності: товари (таблиця "Item"), зв'язки між магазинами та товарами (таблиця "Connections"), магазини (таблиця "Shop"), поточні кошики покупок (таблиця "CurrentBasket") та історичні записи кошиків покупок (таблиця "History").

Таблиця "Item" містить інформацію про окремі товари, зокрема їхні ідентифікатори, назви та описи. Зв'язки між магазинами та товарами зберігаються в таблиці "Connections", де для кожного зв'язку зазначаються його унікальний ідентифікатор, ідентифікатор магазину та ідентифікатор товару. Таблиця "Shop" містить дані про магазини, такі як ідентифікатор, місцезнаходження, координати та назву.

Для відстеження поточних кошиків покупок використовується таблиця "CurrentBasket", де зберігаються дані про кожен кошик, зокрема його унікальний ідентифікатор та дата створення. Поле "BasketConnections" в цій таблиці є списковим типом даних і містить зв'язані з кошиком зв'язки з таблиці "Connections".

Окрім поточних кошиків, база даних також зберігає історичні записи кошиків покупок у таблиці "History". Кожен історичний запис має свій унікальний ідентифікатор та містить зв'язки зі списками з таблиці "CurrentBasket".

Ця структура бази даних дозволяє ефективно зберігати та отримувати інформацію про товари, магазини, зв'язки між ними, поточні кошики покупок та їх історію. Вона створена з урахуванням потреб додатку "Навігаційний асистент покупок", щоб забезпечити зручну навігацію та збереження списків покупок для користувачів.

Загальний вигляд структура бази даних зображено на рисунку 3.1.

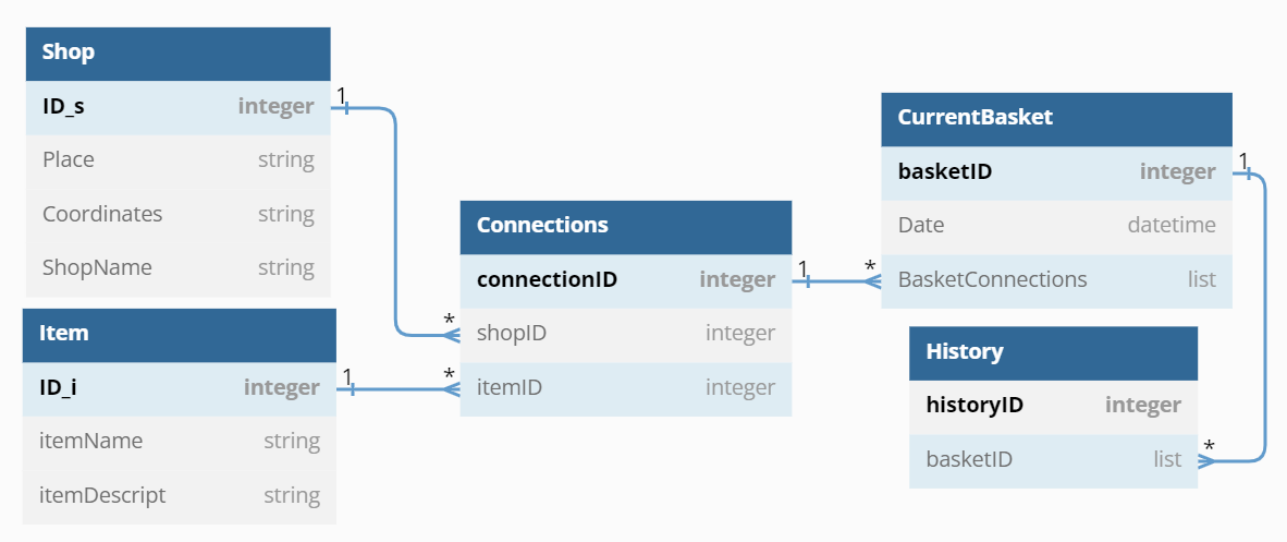

Рис. 3.1. Структура бази даних

Так як, додаток працює на мобільному пристрої, то і СКБД буде використана мобільна та кросплатформова.

SQLite – це полегшена реляційна система керування базами даних. SQLite є бібліотекою, яка реалізує багато функцій стандарту SQL-92. Вихідний код SQLite доступний як вільне програмне забезпечення, що означає його безкоштовне використання без обмежень. Розробники SQLite отримують фінансову підтримку від спеціального консорціуму, в який входять такі компанії, як Adobe, Oracle, Mozilla, Nokia, Bentley і Bloomberg.

Процес збереження даних на пристрої відбувається за допомогою автоматизованого створення файлу, що міститиме екземпляри даних у таблиці. Файл із розширенням .sqlite являє собою полегшений файл бази даних SQL. Це база даних у самому файлі, що реалізує автономний, повнофункціональний, високонадійний механізм бази даних SQL.

Шлях до файлів, у яких зберігаються дані для бази даних є статичним у рамках проєкту та їх створення автоматизоване при першому запуску програми. Видалення програми призведе до очищення бази даних.

Процес зі створення та організації таблиць на пристрої відбувається за допомогою створення конкретних публічних класів, що міститимуть у собі дані про наповнення таблиці. Опис допоміжних класів створених у рамках проєкту можна побачити на рисунку 3.2.

3.5 Взаємодія з базою даних

Для взаємодії з базою даних використовується клас DatabaseHelper, який містить необхідні методи для створення, отримання, оновлення та видалення даних з таблиць бази даних.

Клас DatabaseHelper включає наступні основні методи:

а) Методи отримання списків з таблиць:

1) GetItems(): Повертає список всіх продуктів з таблиці "Item";
2) GetNotes(): Повертає список всіх нотаток з таблиці "Record";
3) GetShops(): Повертає список всіх магазинів з таблиці "Shop";
4) GetConnections(): Повертає список всіх зв'язків з таблиці "Connections";
5) GetNote(int record): Повертає конкретну нотатку за заданим ідентифікатором.

а) Методи вставки нових записів в таблиці:

1) SaveItem(Item item): Додає новий запис про продукт до таблиці "Item";
2) SaveShop(Shop shop): Додає новий запис про магазин до таблиці "Shop";
3) SaveNote(Record note): Додає новий запис про нотатку до таблиці "Record";
4) SaveConnection(Connections connections): Додає новий запис про зв'язок до таблиці "Connections".

б) Методи видалення даних:

1) DeleteAllItems(): Видаляє всі записи з таблиці "Item";
2) DeleteAllShop(): Видаляє всі записи з таблиці "Shop";
3) DeleteAllNotes(): Видаляє всі записи з таблиці "Record";
4) DeleteShop(Shop shop): Видаляє конкретний запис про магазин з таблиці "Shop";
5) DeleteItem(Item item): Видаляє конкретний запис про продукт з таблиці "Item".

Клас DatabaseHelper також містить тимчасові змінні LastAddress та LastCoordinates, які зберігають останню адресу та координати для використання в додатку.

Всі операції з базою даних виконуються за допомогою класу SQLiteConnection, який є вбудованим файловим рішенням для роботи з базою даних SQLite. У конструкторі DatabaseHelper відбувається створення та налаштування з'єднання з базою даних, а також створення таблиць, необхідних для функціонування додатку.

У разі виникнення помилок під час виконання операцій з базою даних, метод PushMesssage(string msg) відображає повідомлення про помилку.

Вищевказаний код класу DatabaseHelper виконує функції, необхідні для реалізації операцій створення, отримання, оновлення та видалення даних (CRUD) у базі даних, яка використовується в додатку "Навігаційний асистент покупок".

Діаграма класу DatabaseHelper зображена на рисунку 3.2.

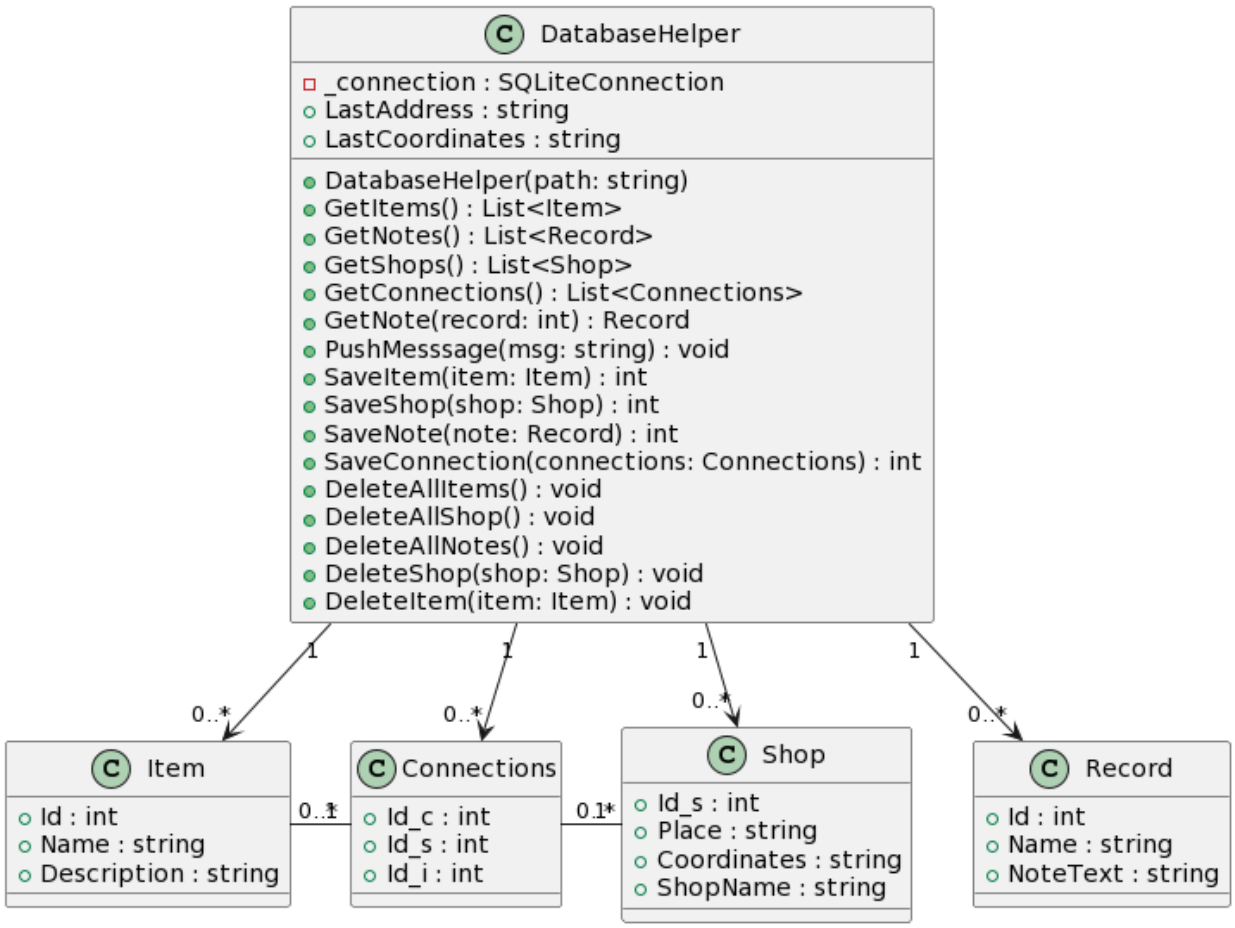

Рис. 3.2. Зведена діаграма класів проєкту

3.6 Тестування та валідація додатка

3.5.1. План тестування

План тестування розроблено з метою забезпечення якості та надійності мобільного додатку "навігаційний асистент покупок". У цьому розділі ми наведемо загальну стратегію тестування, а також детальніші плани для кожного виду тесту.

Використані стратегії тестування:

а) використання ручного тестування для забезпечення широкого покриття функціональності та недопущення помилок;

б) виконання тестування на різних пристроях, емуляторах та версіях операційної системи Android для забезпечення сумісності та стабільності додатка;

в) встановлення метрик якості, таких як час відгуку, завантаження системи та реагування на навантаження, для оцінки продуктивності додатка.

План тестування включає наступні етапи:

а) функціональне тестування:

1) перевірка основних функцій додатка, включаючи створення списків покупок, пошук товарів та маршрутизацію;

2) виконання тестових сценаріїв для перевірки правильності реалізації функціональності;

3) виявлення та документування помилок та неполадок;

б) інтеграційне тестування:

1) перевірка взаємодії додатка з зовнішніми сервісами, такими як геолокація та база даних продуктів;

2) виконання тестових сценаріїв для перевірки правильності передачі даних та взаємодії з іншими компонентами системи;

3) виявлення та документування проблем з інтеграцією;

в) тести на продуктивність та навантаження:

1) виконання тестів для вимірювання часу відгуку додатка та швидкості обробки запитів при різних навантаженнях;

2) визначення максимальних меж продуктивності та перевірка, чи відповідає додаток цим вимогам;

г) тести на безпеку:

1) виконання тестів на перехоплення даних, SQL-ін'єкції та інші вразливості, щоб переконатися в стійкості додатка до потенційних атак;

2) виявлення та документування потенційних проблем безпеки та розробка заходів для їх виправлення;

д) валідація результатів:

1) перевірка виправлених помилок після тестування та переконання в їх коректності;

2) повторне виконання тестів, щоб перевірити, чи відповідає додаток встановленим вимогам після внесених змін.

Документування результатів тестування та валідації для подальшого аналізу та представлення у документації дипломної роботи.

3.5.2. Функціональне тестування

Функціональне тестування є важливою складовою частиною процесу розробки мобільного додатку "навігаційний асистент покупок". Цей вид тестування спрямований на перевірку коректності та відповідності функціональних вимог додатка.

У розділі "Функціональне тестування" ми успішно провели перевірку коректності та відповідності функціональних вимог мобільного додатку "навігаційний асистент покупок".

Під час тестування ми виконали наступні завдання:

Перевірили основні функції додатка, включаючи створення списків покупок, додавання елементів до них, пошук товарів та фільтрацію результатів,

відображення маршрутів до магазинів та навігацію, а також збереження покупок та історії списків. Усі функції працюють коректно та відповідають вимогам.

Виконали тестові сценарії, щоб перевірити правильність функціонування додатка. Задані сценарії взаємодії з додатком успішно виконані, дані обробляються правильно, і очікувані результати збігаються з фактичними.

Перевірили взаємодію з іншими компонентами системи, зокрема з геолокаційним сервісом для точного визначення місцезнаходження користувача та побудови маршрутів, інтеграцію з базою даних продуктів для отримання актуальної інформації, а також правильне відображення та взаємодію з іншими модулями додатка. Усі взаємодії працюють належним чином.

Перевірили коректну обробку помилок, включаючи тестування обробки некоректних або недійсних даних. Додаток належним чином реагує на помилки та надає інформативні повідомлення.

Задокументували результати тестування, включаючи виявлені проблеми та рекомендації щодо виправлення. Описали проходження тестових сценаріїв, результати перевірок та спостереження.

На підставі функціонального тестування ми можемо зробити висновок, що функціональність додатка "навігаційний асистент покупок" відповідає вимогам та працює стабільно та надійно. Ми готові переходити до наступного етапу розробки та тестування, зосереджуючись на інтеграційному тестуванні.

3.5.3. Інтеграційне тестування

Інтеграційне тестування стосується взаємодії додатка та підсистем середовища на якому його запущено. Загалом цей тип тестування застосовується для забезпечення сумісності та правильності обміну даними між компонентами системи загалом.

Наприклад, ми перевірили інтеграцію з базою даних та забезпечили коректне збереження та вилучення даних зі СКБД SQLite. Також, переконалися, що дані відображаються правильно та зберігаються без втрати.

Зокрема було перевірено взаємодію з зовнішніми сервісами, зокрема з геолокаційним сервісом для отримання актуального місцезнаходження користувача та побудови маршрутів. Переконалися, що дані передаються та обробляються правильно.

3.5.4. Тести на продуктивність та навантаження

Для забезпечення оптимальної продуктивності та надійності додатку "Навігаційний асистент покупок", були проведені тести на продуктивність та навантаження. Ці тести допомогли оцінити, як додаток впорядковується з великим обсягом даних та різними навантаженнями.

У процесі тестування на продуктивність були виконані наступні кроки, наприклад тестування завантаження додатку (проведене тестування швидкості завантаження додатку на різних пристроях та під різними умовами мережі). Також в рамках цього виду тестування було проведено вимірювання швидкості виконання операцій, таких наприклад, як побудова маршруту та оновлення даних, завантаження БД.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи були проведені дослідження та розроблено мобільний додаток «Навігаційний асистент покупок». Даний додаток має на меті надати користувачам зручну навігаційну систему для покупок, використовуючи можливості навігаційних карт та інтеграцію зовнішніх сервісів.

У розділі "Аналіз технологій використання навігаційних карт та їх порівняння" було проведено аналіз різних технологій навігаційних карт, зокрема Google Maps, Mapbox та OpenStreetMap. Було виявлено, що для реалізації задач додатка найбільш відповідною технологією є Google Maps, завдяки своїй широкій функціональності та зручному інтерфейсу.

У розділі "Розробка дизайну додатка за допомогою Figma та реалізація інтерфейсу в Xamarin.Forms" було детально описано процес розробки дизайну додатка за допомогою інструменту Figma та реалізація інтерфейсу в середовищі Xamarin.Forms. Були враховані особливості адаптації дизайну до різних пристроїв та платформ, а також інтеграція дизайну з функціональністю додатка.

У розділі "Розробка бази даних та управління даними з використанням СКБД SQLite" було проведено проєктування структури бази даних та реалізовано CRUD-операції для збереження та управління даними в додатку.

Для забезпечення якості додатка був розроблений план тестування, який включав функціональне тестування, інтеграційне тестування та тести на продуктивність та навантаження.

Загальним результатом кваліфікаційної роботи є розроблений мобільний додаток «Навігаційний асистент покупок», який забезпечує зручну навігаційну систему для покупок з використанням навігаційних карт та інтеграції зовнішніх сервісів. Даний додаток має наукову, практичну та економічну цінність, сприяє поліпшенню процесу покупок та забезпечує зручну взаємодію з користувачем. Результати роботи можуть бути використані для подальшого розвитку та удосконалення подібних мобільних додатків у майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Charles Petzold. Creating Mobile Apps with Xamarin.Forms First Edition / Petzold C. — Redmond: Microsoft Press, 2016. — 1187 c.
2. Dharani, R. Web API Design: Crafting Interfaces that Developers Love – Independently published, December, 2017. - 37 p.
3. Jonathan Peppers. Xamarin Cross-platform Application Development Second Edition / Peppers J. — Birmingham: Packt, 2015. — 459 c.
4. Nilanchala Panigrahy. Xamarin Mobile Application Development for Android Second Edition / Panigrahy N. — Birmingham: Packt, 2015. — 296 c.
5. Distance Matrix API: developer's guide. - [Електронний ресурс]. Режим доступу: https://developers.google.com/maps/documentation/distance-matrix/start?hl=ua (дата звернення 23.06.2023). – Назва з екрана.
6. Maps SDK for Android. - [Електронний ресурс]. Режим доступу: https://developers.google.com/maps/documentation/android-sdk/mapwith-marker?hl=ua (дата звернення: березень-квітень 2023).
7. Офіційна документація Figma [Електронний ресурс] – Режим доступу: https://www.figma.com/developers/ (дата звернення 23.06.2023). – Назва з екрана.
8. Офіційна документація Google Maps API [Електронний ресурс] – Режим доступу: https://developers.google.com/maps/documentation/ (дата звернення 23.06.2023). – Назва з екрана.
9. Офіційна документація SQLite [Електронний ресурс] – Режим доступу: https://www.sqlite.org/docs.html (дата звернення 23.06.2023). – Назва з екрана.
10. Офіційна документація Xamarin.Forms [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-us/xamarin/xamarin-forms/ (дата звернення 23.06.2023). – Назва з екрана.

Додаток А

Текст файлу Info_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Page1">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="info_wall_1"
                        Source="info_wall_1.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                <Label x:Name="info_logo_text"
                            FontFamily="IBM Plex Sans"
                            HorizontalTextAlignment="Center"
                            Opacity="0.9"
                            FontSize="24"
                            TextColor="#000000"
                            AbsoluteLayout.LayoutBounds="12,135,375,68"
                            Text=""                          />
                <ImageButton x:Name="ms_info_button"
                        Source="ms_info_button.png"
                        AbsoluteLayout.LayoutBounds="0.95,0.01,0.15,0.1"
                        AbsoluteLayout.LayoutFlags="All"/>
            </AbsoluteLayout>
        </ScrollView>
    </ContentPage.Content>
    <!--<ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <AbsoluteLayout x:Name="page_info_screen_0_done_ek1"
                         AbsoluteLayout.LayoutBounds="0,0,390,844"
                  <BoxView x:Name="_bg__info_screen_0_done_ek2"
                            Color="#"
                            AbsoluteLayout.LayoutBounds="0,0,390,844" >    <!--
                   </BoxView>
                   <Image x:Name="info_wall_1"
                            Source="info_wall_1.png"
                            AbsoluteLayout.LayoutBounds="0,0,390,844" />
                   <Label x:Name="info_logo_text"
                            FontFamily="IBM Plex Sans"
                            HorizontalTextAlignment="Center"
                            Opacity="0.9"
                            FontSize="24"
                            TextColor="#000000"
                            AbsoluteLayout.LayoutBounds="12,135,375,68"
                            Text=""                          />
                   <Image x:Name="info_button"
                            Source="info_button.png"
                            AbsoluteLayout.LayoutBounds="324,55,50,50" />
                </AbsoluteLayout>
            </AbsoluteLayout>
        </ScrollView>
    </ContentPage.Content>-->
</ContentPage>
```

Додаток Б

Текст файлу LoyaltyCard_Screen_2.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.LoyaltyCard_Screen_2">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="lc_wall_1"
                        Source="lc_wall_1.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                <Image x:Name="lc_del_button"
                             Source="lc_del_button.png"
                             AbsoluteLayout.LayoutBounds="50,711,107,64.85" />
                <Image x:Name="lc_cancel_button"
                             Source="lc_cancel_button.png"
                             AbsoluteLayout.LayoutBounds="236,711,107,64.85" />
                <BoxView x:Name="lc_plate_2_container_p2"
                             CornerRadius="20"
                             Color="#DAECEE"
                             AbsoluteLayout.LayoutBounds="205,220,150,214" />
                <Label x:Name="lc_plate_2_text"
                             FontFamily="IBM Plex Sans"
                             HorizontalTextAlignment="Center"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="210,381,149,60"
                             Text="Test2"                          />
                <ImageButton x:Name="lc_photo_photo_add"
                             Source="lc_photo_photo_add.png"
                            BackgroundColor="Transparent"
                            Clicked="lc_photo_photo_add_Clicked"
                             AbsoluteLayout.LayoutBounds="215,235,130,146" />
                <BoxView x:Name="lc_plate_1_container_p2"
                             CornerRadius="20"
                             Color="#DAECEE"
                             AbsoluteLayout.LayoutBounds="35,220,150,214" />
                <Label x:Name="lc_plate_1_text_example"
                             FontFamily="IBM Plex Sans"
                             HorizontalTextAlignment="Center"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="40,381,149,60"
                             Text="Test1"/>
                <Image x:Name="lc_photo_photo_example"
                             Source="lc_photo_photo_example.png"
                             AbsoluteLayout.LayoutBounds="45,235,130,146" />
                <Image x:Name="lc_selected_plate_icon"
                             Source="lc_selected_plate_icon.png"
                             AbsoluteLayout.LayoutBounds="157.81,222.81,25,25" >
                       <!--ЛОГОТИП МАШИНИ-->
                <Image x:Name="lc_logo_icon"
                             Source="lc_logo_icon.png"
                             AbsoluteLayout.LayoutBounds="90,90,205,64" />
                <Image x:Name="lc_search_icon_container"
```

```
                    Source="lc_search_icon_container.png"
                    AbsoluteLayout.LayoutBounds="85,20,289,50" />
            <Label x:Name="lc_search_text"
                    FontFamily="IBM Plex Sans"
                    FontSize="20"
                    TextColor="#000000"
                    AbsoluteLayout.LayoutBounds="145,30,243,40"
                    Text="Пошук"/>
            <ImageButton x:Name="lc_back_button"
                    BackgroundColor="Transparent"
                    Source="lc_back_button.png"
                    Clicked="lc_back_button_Clicked"
                    AbsoluteLayout.LayoutBounds="16,20,50,50" />
        </AbsoluteLayout>
    </ScrollView>
  </ContentPage.Content>
</ContentPage>
```

Додаток В

Текст файлу LoyaltyCard_Screen_2_1.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.LoyaltyCard_Screen_2_1">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <AbsoluteLayout x:Name="page_loyaltycard_screen_2_1_done_ek1"
                        AbsoluteLayout.LayoutBounds="0,0,390,844"      >
                    <BoxView x:Name="_bg__loyaltycard_screen_2_1_done_ek2"
                             Color="#"
                             AbsoluteLayout.LayoutBounds="0,0,390,844" >
                        <!-- Effect support coming soon for Xamarin.Forms -->
                    </BoxView>
                    <Image x:Name="lc2_1_wall_2"
                             Source="lc2_1_wall_2.png"
                             AbsoluteLayout.LayoutBounds="0,0,390,846" />
                    <Image x:Name="lc2_1_del_button"
                             Source="lc2_1_del_button.png"
                             AbsoluteLayout.LayoutBounds="50,711,107,64.85" />
                    <Image x:Name="lc2_1_cancel_button"
                             Source="lc2_1_cancel_button.png"
                             AbsoluteLayout.LayoutBounds="236,711,107,64.85" />
                    <BoxView x:Name="lc2_1_photo_1_container"
                             CornerRadius="20"
                             Color="#DAECEE"
                             AbsoluteLayout.LayoutBounds="16,62,354.67,603" />
                    <Image x:Name="lc2_1_photo_1_icon_example"
                             Source="lc2_1_photo_1_icon_example.png"
                             AbsoluteLayout.LayoutBounds="99,282,188,165" />
                </AbsoluteLayout>
            </AbsoluteLayout>
        </ScrollView>
    </ContentPage.Content>
</ContentPage>
```

Додаток Г

Текст файлу MainPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.MainPage">

    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="ms_wall_2"
                        Source="ms_wall_2.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                <Label x:Name="ms_time"
                    FontFamily="IBM Plex Sans"
                    HorizontalTextAlignment="Center"
                    Opacity="0.9"
                    FontSize="50"
                    TextColor="#000000"
                    AbsoluteLayout.LayoutBounds="0.5,0.15,0.5,0.275"
                    AbsoluteLayout.LayoutFlags="All"
                    Text="{Binding CurrentTime}" />
                <!--ІНФО-->
                <ImageButton x:Name="ms_info_button"
                        Source="ms_info_button.png"
                        BackgroundColor="Transparent"
                        AbsoluteLayout.LayoutBounds="0.95,0.01,0.15,0.09"
                        AbsoluteLayout.LayoutFlags="All"
Clicked="ms_info_button_Clicked"/>
                <!--МЕНЮ-->
                <ImageButton x:Name="ms_menu_button"
                        Source="ms_menu_button.png"
                        BackgroundColor="Transparent"
                        AbsoluteLayout.LayoutBounds="0.05,0.01,0.15,0.09"
                        AbsoluteLayout.LayoutFlags="All"
Clicked="ms_menu_button_Clicked"/>
                <!--МАГАЗИНИ-->
                <ImageButton x:Name="ms_shop_button"
                        Source="ms_shop_button.png"
                        BackgroundColor="Transparent"
                        AbsoluteLayout.LayoutBounds="0.15,0.3,0.39,0.23"
                        AbsoluteLayout.LayoutFlags="All"
Clicked="ms_shop_button_Clicked"/>
                <!--НОТАТКИ-->
                <ImageButton x:Name="ms_note_icon_button"
                        Source="ms_note_icon_button.png"
                        BackgroundColor="Transparent"
                        AbsoluteLayout.LayoutBounds="0.85,0.3,0.39,0.23"

                        AbsoluteLayout.LayoutFlags="All"
                        Clicked="ms_note_icon_button_Clicked"/>
                <!--ІСТОРІЯ ПОКУПОК-->
                <ImageButton x:Name="ms_history_button"
                        Source="ms_history_button.png"
                        BackgroundColor="Transparent"
                        AbsoluteLayout.LayoutBounds="0.15,0.65,0.39,0.23"
```

```
                    AbsoluteLayout.LayoutFlags="All" />
                <!--КАРТА ЛОЯЛЬНОСТІ-->
                <ImageButton x:Name="ms_loyaltycard_button"
                    Source="ms_loyaltycard_button.png"
                    BackgroundColor="Transparent"
                    AbsoluteLayout.LayoutBounds="0.85,0.65,0.39,0.23"
                    AbsoluteLayout.LayoutFlags="All"
                    Clicked="ms_loyaltycard_button_Clicked"/>
                <!--ЗНАЙТИ МАРШРУТ-->
                <ImageButton x:Name="ms_map_button"
                    Source="ms_map_button.png"
                    BackgroundColor="Transparent"
                    AbsoluteLayout.LayoutBounds="0.5,1,0.834,0.23"
                    AbsoluteLayout.LayoutFlags="All"
                         Clicked="ms_map_button_Clicked"/>
            </AbsoluteLayout>
        </ScrollView>
    </ContentPage.Content>

</ContentPage>
```

Додаток Д

Текст файлу Map_Pick_Point_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:maps="clr-
namespace:Xamarin.Forms.Maps;assembly=Xamarin.Forms.Maps"
             x:Class="NavigatShopAssistant.Map_Pick_Point_Screen">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>


                <AbsoluteLayout x:Name="page_map_startpoint_screen_5_5_done_ek1"
                        AbsoluteLayout.LayoutBounds="0,0,390,844"
                        >

                    <BoxView x:Name="_bg__map_startpoint_screen_5_5_done_ek2"
                             Color="#"
                             AbsoluteLayout.LayoutBounds="0,0,390,844" >
                        <!-- Effect support coming soon for Xamarin.Forms -->
                    </BoxView>
                    <Image x:Name="map_startpoint_wall_1"
                             Source="map_startpoint_wall_1.png"
                             AbsoluteLayout.LayoutBounds="0,0,390,844" />
                    <BoxView x:Name="map_startpoint_map_container"
                             CornerRadius="20"
                             Color="#DAECEE"
                             AbsoluteLayout.LayoutBounds="16,136,358,579" />
                    <Image x:Name="map_startpoint_logo_icon"
                             Source="map_startpoint_logo_icon.png"
                             AbsoluteLayout.LayoutBounds="175,60,40,40" />
                    <Image x:Name="map_startpoint_back_button"
                             Source="map_startpoint_back_button.png"
                             AbsoluteLayout.LayoutBounds="16,55,50,50" />
                    <ImageButton x:Name="map_startpoint_mathpath_button"
                             Source="map_startpoint_mathpath_button.png"
                           BackgroundColor="Transparent"

Clicked="map_startpoint_mathpath_button_Clicked"
                             AbsoluteLayout.LayoutBounds="16,727,358,49.8" />
                    <Label x:Name="map_startpoint_logo_text"
                             FontFamily="IBM Plex Sans"
                             HorizontalTextAlignment="Center"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="13,105,373,45"
                             Text="  "
                         />
                      <maps:Map x:Name="map"
                    IsShowingUser="True"
                    MapType="Street"
                               MapClicked="map_MapClicked"
                               AbsoluteLayout.LayoutBounds="16,136,358,575">

                      </maps:Map>
                </AbsoluteLayout>
            </AbsoluteLayout>

        </ScrollView>
    </ContentPage.Content>
</ContentPage>
```

Додаток Е

Текст файлу Map_Selectitem_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Map_Selectitem_Screen">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="map_si_wall_1"
                        Source="shop_wall_2.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                <!--РОЗРАХУНОК МАРШРУТУ-->
                <ImageButton x:Name="map_si_next_button"
                             Source="map_si_next_button.png"
                       Clicked="map_si_next_button_Clicked"
                             AbsoluteLayout.LayoutBounds="0.5,0.9,0.92,0.058"
                          AbsoluteLayout.LayoutFlags="All" />
                <CollectionView x:Name="itemsCollection"
                    AbsoluteLayout.LayoutBounds="15,320,2,1"
                                SelectionMode="Multiple"

SelectionChanged="itemsCollection_SelectionChanged"
                        AbsoluteLayout.LayoutFlags="SizeProportional" >
                    <CollectionView.ItemTemplate>
                        <DataTemplate>
                            <AbsoluteLayout>

                                <!--ФОН ТОВАРУ-->
                                <Image x:Name="map_si_selected_plate_button"

Source="map_si_selected_plate_button.png"
                                                    AbsoluteLayout.LayoutBounds="1,5"
/>
                                <!--ПЛЮСИК ДЛЯ ТОВАРУ-->
                                <Image x:Name="map_si_addplate_button"

Source="map_si_addplate_button.png"

AbsoluteLayout.LayoutBounds="300,5" />
                                <!--НАЗВА ТОВАРУ-->
                                <Label x:Name="map_si_plate_1_text_example"
                                            FontFamily="IBM Plex Sans"
                                            FontSize="20"
                                            TextColor="#000000"

AbsoluteLayout.LayoutBounds="60,15"
                                            Text="{Binding Name}"
                                        />
                            </AbsoluteLayout>
                        </DataTemplate>
                    </CollectionView.ItemTemplate>
                </CollectionView>
                <!--ОКРЕМА ГАЛОЧКА ДЛЯ ОБРАНОГО ТОВАРУ-->
                <!--<Image x:Name="map_si_selected_plate_icon"
                             Source="map_si_selected_plate_icon.png"
```

```
                    AbsoluteLayout.LayoutBounds="30,428,20,20" />-->
            <!--КОНТЕЙНЕР ДЛЯ ПОШУКУ-->
            <Image x:Name="map_si_search_container"
                    Source="map_si_search_container.png"
                    AbsoluteLayout.LayoutBounds="16,258,358,49.54" />
            <!--ТЕКСТ ПОШУКУ-->
            <Label x:Name="map_si_search_text"
                    FontFamily="IBM Plex Sans"
                    FontSize="20"
                    TextColor="#000000"
                    AbsoluteLayout.LayoutBounds="76,270,243,40"
                    Text="Текст пошуку"/>
            <!--ПОШУК З ІСТОРІЇ ПОКУПОК-->
            <ImageButton x:Name="map_si_fromhistory_button"
                    Source="map_si_fromhistory_button.png"
                   Clicked="map_si_fromhistory_button_Clicked"
                    AbsoluteLayout.LayoutBounds="16,188,358,49.8" />
            <!--ЛОГОТИП-->
            <Image x:Name="map_si_logo_icon"
                    Source="map_si_logo_icon.png"
                    AbsoluteLayout.LayoutBounds="32,55,324,104.4" />
            <!--НОТАТКА?-->
            <Image x:Name="map_si_whatadd_button"
                    Source="map_si_whatadd_button.png"
                    AbsoluteLayout.LayoutBounds="324,10,50,50" />
            <!--КНОПКА ПОВЕРНЕННЯ-->
            <ImageButton x:Name="map_si_back_button"
                    Source="map_si_back_button.png"
                    BackgroundColor="Transparent"
                    Clicked="map_si_back_button_Clicked"
                    AbsoluteLayout.LayoutBounds="16,10,50,50" />
            <!--РОЗРАХУНОК МАРШРУТУ-->
            <ImageButton x:Name="map_si_next_button_2"
                   Clicked="map_si_next_button_Clicked"
                         BackgroundColor="Transparent"
                          AbsoluteLayout.LayoutBounds="0.5,0.9,0.92,0.058"
                       AbsoluteLayout.LayoutFlags="All" />
        </AbsoluteLayout>
    </ScrollView>
  </ContentPage.Content>
</ContentPage>
```

Додаток Ж

Текст файлу Menu_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Menu_Screen">
    <ContentPage.Content>
        <ScrollView>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="*" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                </Grid.ColumnDefinitions>
                <AbsoluteLayout>
                    <Image x:Name="menu_s_wall_1"
                       Source="menu_s_wall_1.png"
                       AbsoluteLayout.LayoutBounds="0,0,390,844" />
                    <Label x:Name="menu_s_support_text_button"
                       FontFamily="IBM Plex Sans"
                       Opacity="0.9" FontSize="24"
                       TextColor="#000000"
                       AbsoluteLayout.LayoutBounds="80,378,328,68"
                       Text="Тех. підтримка" />
                    <Image x:Name="menu_s_support_icon_button"
                       Source="menu_s_support_icon_button.png"
                       AbsoluteLayout.LayoutBounds="16,378,51,51" />
                    <Label x:Name="menu_s_buy_text_button"
                       FontFamily="IBM Plex Sans"
                       Opacity="0.9" FontSize="24"
                       TextColor="#000000"
                       AbsoluteLayout.LayoutBounds="80,297,328,68"
                       Text="Підписка" />
                    <Image x:Name="menu_s_buy_icon_button"
                       Source="menu_s_buy_icon_button.png"
                       AbsoluteLayout.LayoutBounds="16,297,51,51" />
                    <Label x:Name="menu_s_theme_text_button"
                       FontFamily="IBM Plex Sans"
                       Opacity="0.9" FontSize="24" TextColor="#000000"
                       AbsoluteLayout.LayoutBounds="80,216,323,68"
                       Text="Тема програми" />
                    <Image x:Name="menu_s_theme_icon_button"
                       Source="menu_s_theme_icon_button.png"
                       AbsoluteLayout.LayoutBounds="16,216,51,51" />
                    <Label x:Name="menu_s_place_text_button"
                       FontFamily="IBM Plex Sans"
                       Opacity="0.9" FontSize="24"   TextColor="#000000"
                       AbsoluteLayout.LayoutBounds="80,135,289,68"
                       Text="Моє розташування" />
                    <Image x:Name="menu_s_place_icon_button"
                       Source="menu_s_place_icon_button.png"
                       AbsoluteLayout.LayoutBounds="16,135,51,51" />
                    <Image x:Name="menu_s_button"
                       Source="menu_s_button.png"
                       AbsoluteLayout.LayoutBounds="324,55,50,50" />
                </AbsoluteLayout>
            </Grid>
        </ScrollView>
    </ContentPage.Content>
</ContentPage>
```

Додаток И

Текст файлу Notes_Record_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Notes_Record_Screen">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="records_wall_2"
                        Source="records_wall_2.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                <!--<AbsoluteLayout x:Name="page_records_screen_4_done_ek1"
                         AbsoluteLayout.LayoutBounds="0,0,390,844"
                         >
                    <Image x:Name="records_wall_2"
                             Source="records_wall_2.png"
                             AbsoluteLayout.LayoutBounds="0,0,390,846" />-->

                <!--ВИДАЛИТИ ВСІ-->
                    <Image x:Name="records_del_button"
                             Source="records_del_button.png"
                             AbsoluteLayout.LayoutBounds="50,711,107,64.85" />
                <!--ВІДМІНА-->
                    <Image x:Name="records_cancel_button"
                             Source="records_cancel_button.png"
                             AbsoluteLayout.LayoutBounds="236,711,107,64.85" />

                <CollectionView x:Name="itemsCollection"
                    AbsoluteLayout.LayoutBounds="0,0,1,1"

                        AbsoluteLayout.LayoutFlags="SizeProportional" >
                    <CollectionView.ItemTemplate>
                        <DataTemplate>
                            <AbsoluteLayout>
                                <!--ФОН ЗАПИСУ-->
                                <ImageButton x:Name="records_plate_1_button"
                                         Source="records_plate_1_button.png"
                                            BackgroundColor="Transparent"

Clicked="records_plate_1_button_Clicked"
                                            BindingContext="{Binding Id}"

AbsoluteLayout.LayoutBounds="16,215,358,50.16" />


                                <!--ВИБІР ЗАПИСУ-->
                                <!--<Image
x:Name="record_noselected_plate_button"

Source="record_noselected_plate_button.png"

AbsoluteLayout.LayoutBounds="16,215,46,50.35" />-->
                            <!--ВИДАЛИТИ КОНКРЕТНИЙ ЗАПИС-->
                                <Image x:Name="records_plate_del_button"
                                         Source="records_plate_del_button.png"
```

```
AbsoluteLayout.LayoutBounds="310,215,64,50.26" />
                            <!--НАЗВА ЗАПИСУ-->
                                <Label x:Name="records_plate_1_text"
                                         FontFamily="IBM Plex Sans"
                                         FontSize="20"
                                         TextColor="#000000"

AbsoluteLayout.LayoutBounds="77,225,276,62"
                                         Text="{Binding Name}"
                                      />
                            <!--ІКОНКА ОБРАНОГО ЗАПИСУ-->
                                        <Image
x:Name="records_pusher_plate_icon"

Source="records_selected_plate_icon.png"

AbsoluteLayout.LayoutBounds="30,230,20,20" />
                                    <ImageButton
x:Name="records_pusher_plate_1_button"
                                                BackgroundColor="Transparent"

Clicked="records_plate_1_button_Clicked"
                                                BindingContext="{Binding Id}"

AbsoluteLayout.LayoutBounds="16,215,358,50.16" />
                                </AbsoluteLayout>
                        </DataTemplate>
                    </CollectionView.ItemTemplate>
                    <CollectionView.Footer>
                        <!--ДОДАТИ-->
                        <ImageButton x:Name="records_add_plate_button"
                                     Source="records_add_plate_button.png"
                                     BackgroundColor="Transparent"
                                     Clicked="records_add_plate_button_Clicked"
                                     />
                    </CollectionView.Footer>
                </CollectionView>
                <!--ГОЛОВНА ІКОНКА МЕНЮ-->
                    <Image x:Name="records_logo_icon"
                             Source="records_logo_icon.png"
                             AbsoluteLayout.LayoutBounds="156,126,78,62" />
                <!--ПОШУК ПО НОТАТКАМ-->
                    <Image x:Name="records_search_container"
                             Source="records_search_container.png"
                             AbsoluteLayout.LayoutBounds="85,55,289,50" />
                <!--ЗАГЛУШКА ТЕКСТ ПОШУКУ-->
                    <Label x:Name="records_search_text"
                             FontFamily="IBM Plex Sans"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="145,67,243,40"
                             Text=""                          />
                <!--КНОПКА НАЗАД-->
                    <ImageButton x:Name="records_back_button"
                             Source="records_back_button.png"
                                 BackgroundColor="Transparent"
                             AbsoluteLayout.LayoutBounds="16,55,50,50"
                           Clicked="records_back_button_Clicked"/>
                </AbsoluteLayout>
            <!--</AbsoluteLayout>-->

        </ScrollView>

    </ContentPage.Content>
</ContentPage>
```

Додаток І

Текст файлу Record_Screen_4.1.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Record_Screen_4">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="record_wall_2"
                        Source="record_wall_2.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                    <!--<Image x:Name="record_wall_2"
                             Source="record_wall_2.png"
                             AbsoluteLayout.LayoutBounds="0,0,390,846" />-->
                    <ImageButton x:Name="record_next_button"
                             Source="record_next_button.png"
                                BackgroundColor="Transparent"
                                Clicked="record_next_button_Clicked"
                             AbsoluteLayout.LayoutBounds="235,711,105,64.95" />
                    <ImageButton x:Name="record_cancel_button"
                             Source="record_cancel_button.png"
                                BackgroundColor="Transparent"
                             AbsoluteLayout.LayoutBounds="50,712,105,64.95"
                           Clicked="record_cancel_button_Clicked"/>
                    <BoxView x:Name="record_onplate_container"
                             CornerRadius="20"
                             Color="#DAECEE"
                             AbsoluteLayout.LayoutBounds="16,129,358,561" />
                    <Editor x:Name="record_text" Placeholder="Введіть текст"
                    AbsoluteLayout.LayoutBounds="25,135,340,561"/>
                    <!--<BoxView x:Name="record_onplate_line"
                             Opacity="0.8"
                             Color="#153254"
                             AbsoluteLayout.LayoutBounds="49,159,27,1" />-->
                    <Image x:Name="record_name_icon"
                             Source="record_name_icon.png"
                             AbsoluteLayout.LayoutBounds="89,34,235,90.97" />
                    <Entry x:Name="record_logo_text"
                             FontFamily="IBM Plex Sans"
                             HorizontalTextAlignment="Center"
                             FontSize="20"
                             TextColor="#000000"

                             AbsoluteLayout.LayoutBounds="95,68,220,40"
                             Placeholder="Введіть назву сюди"
                         />
                    <ImageButton x:Name="record_back_button"
                             Source="record_back_button.png"
                                BackgroundColor="Transparent"
                                Clicked="record_back_button_Clicked"
                             AbsoluteLayout.LayoutBounds="16,55,50,50" />
                </AbsoluteLayout>
        </ScrollView>
    </ContentPage.Content>
</ContentPage>
```

Додаток К

Текст файлу Shop_Add_Items_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Shop_Add_Items_Screen">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <AbsoluteLayout x:Name="page_shop_add_items_screen_done_ek1"
                        AbsoluteLayout.LayoutBounds="0,0,390,1368"
                        >
                    <Image x:Name="sai_wall_1_long"
                             Source="sai_wall_1_long.png"
                             AbsoluteLayout.LayoutBounds="0,0,390,1368" />
                    <Image x:Name="sai_next_button"
                             Source="sai_next_button.png"
                             AbsoluteLayout.LayoutBounds="235,1234,105,64.95" />
                    <Image x:Name="sai_cancel_button"
                             Source="sai_cancel_button.png"
                             AbsoluteLayout.LayoutBounds="50,1235,105,64.95" />


                    <Image x:Name="sai_selected_hypermarket_plate_button"
                             Source="sai_selected_hypermarket_plate_button.png"
                             AbsoluteLayout.LayoutBounds="16,1003,358,50.16" />
                    <Label x:Name="sai_plate_hypermarket_text"
                             FontFamily="IBM Plex Sans"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="81,1003,304,62"
                             Text=""
                        />
                    <Image
x:Name="sai_selected_householdgoodsstore_plate_button"

      Source="sai_selected_householdgoodsstore_plate_button.png"
                             AbsoluteLayout.LayoutBounds="16,944,358,50.16" />
                    <Label x:Name="sai_plate_householdgoodsstore_text"
                             FontFamily="IBM Plex Sans"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="77,945,304,62"
                             Text="  "
                        />
                    <Image x:Name="sai_selected_childrenstoystore_plate_button"

      Source="sai_selected_childrenstoystore_plate_button.png"
                             AbsoluteLayout.LayoutBounds="16,883,358,50.16" />
                    <Label x:Name="sai_plate_childrenstoystore_text"
                             FontFamily="IBM Plex Sans"
                             FontSize="20"
                             TextColor="#000000"
                             AbsoluteLayout.LayoutBounds="77,884,304,62"
                             Text="  "
                        />
                    <Image x:Name="sai_selected_childrensshoestore_plate_button"

      Source="sai_selected_childrensshoestore_plate_button.png"
                             AbsoluteLayout.LayoutBounds="16,824,358,50.16" />
                    <Label x:Name="sai_plate_childrensshoestore_text"
                             FontFamily="IBM Plex Sans"
```

```
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,825,304,62"
                                Text="  "
                            />
                        <Image
x:Name="sai_selected_childrensclothingstore_plate_button"

      Source="sai_selected_childrensclothingstore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,765,358,50.16" />
                        <Label x:Name="sai_plate_childrensclothingstore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,766,304,62"
                                Text="  "
                            />
                        <Image x:Name="sai_selected_cosmeticsstore_plate_button"

      Source="sai_selected_cosmeticsstore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,706,358,50.16" />
                        <Label x:Name="sai_plate_cosmeticsstore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,707,304,62"
                                Text=" "
                            />
                        <Image x:Name="sai_selected_shoestore_plate_button"
                                Source="sai_selected_shoestore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,647,358,50.16" />
                        <Label x:Name="sai_plate_shoestore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,648,304,62"
                                Text=" "
                            />
                        <Image x:Name="sai_selected_clothingstore_plate_button"

      Source="sai_selected_clothingstore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,588,358,50.16" />
                        <Label x:Name="sai_plate_clothingstore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,589,304,62"
                                Text=" "
                            />
                        <Image x:Name="sai_selected_furniturestore_plate_button"

      Source="sai_selected_furniturestore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,529,358,50.16" />
                        <Label x:Name="sai_plate_furniturestore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,530,304,62"
                                Text=" "
                            />
                        <Image x:Name="sai_selected_pharmacy_plate_button"
                                Source="sai_selected_pharmacy_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,470,358,50.16" />
                        <Label x:Name="sai_plate_pharmacy_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
```

```
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,471,304,62"
                                Text=""                          />
                        <Image x:Name="sai_selected_stationerystore_plate_button"
                            Source="sai_selected_stationerystore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,411,358,50.16" />
                        <Label x:Name="sai_plate_stationerystore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,412,304,62"
                                Text=" "                         />
                        <Image x:Name="sai_selected_telemarketer_plate_button"

      Source="sai_selected_telemarketer_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,352,358,50.16" />
                        <Label x:Name="sai_plate_telemarketer_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,353,304,62"
                                Text=""                          />
                        <Image x:Name="sai_selected_petstore_plate_button"
                                Source="sai_selected_petstore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,293,358,50.16" />
                        <Label x:Name="sai_plate_petstore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,294,304,62"
                                Text=""                          />
                        <Image x:Name="sai_selected_supermarket_plate_button"
                                Source="sai_selected_supermarket_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,235,358,50.16" />
                        <Label x:Name="sai_plate_supermarket_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,235,304,62"
                                Text=""                          />
                        <Image x:Name="sai_selected_departmentstore_plate_button"
                            Source="sai_selected_departmentstore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,175,358,50.16" />
                        <Label x:Name="sai_plate_departmentstore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,176,304,62"
                                Text=""                          />
                        <Image x:Name="sai_selected_grocerystore_plate_button"
                            Source="sai_selected_grocerystore_plate_button.png"
                                AbsoluteLayout.LayoutBounds="16,117,358,50.16" />
                        <Label x:Name="sai_plate_grocerystore_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="77,117,304,62"
                                Text=" "                         />
                        <Image x:Name="sai_item_logo_icon"
                                Source="sai_item_logo_icon.png"
                                AbsoluteLayout.LayoutBounds="16,49,358,49.18" />
                    </AbsoluteLayout>
                </AbsoluteLayout>
            </ScrollView>
        </ContentPage.Content>
    </ContentPage>
```

Додаток Л

Текст файлу Shop_Add_Screen.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:control="clr-
namespace:dotMorten.Xamarin.Forms;assembly=dotMorten.Xamarin.Forms.AutoSuggestBo
x"                    x:Class="NavigatShopAssistant.Shop_Add_Screen">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <RelativeLayout>
                    <Image x:Name="shop_add_wall_3"
                        Source="shop_add_wall_3.png"
                        RelativeLayout.WidthConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Width}"
                        RelativeLayout.HeightConstraint="{ConstraintExpression
Type=RelativeToParent, Property=Height}"
                        Aspect="AspectFill"/>
                </RelativeLayout>
                <!--ЛОГОТИП-->
                <Image x:Name="shop_add_logo_icon"
                            Source="shop_add_logo_icon.png"
                            AbsoluteLayout.LayoutBounds="0.5,0.001,0.75,0.12"
                    AbsoluteLayout.LayoutFlags="All" />
                <!--ТЕКСТ НАЗВА МАГАЗИНУ-->
                <Label x:Name="shop_add_nameshop_text"
                            FontFamily="IBM Plex Sans"
                            HorizontalTextAlignment="Center"
                            FontSize="20"
                            TextColor="#000000"
                            AbsoluteLayout.LayoutBounds="0.5,0.11,0.75,0.12"
                    AbsoluteLayout.LayoutFlags="All"
                            Text="*Назва магазину:"/>
                <!--ПОЛЕ ВВОДУ НАЗВИ+ФОН-->
                <BoxView x:Name="shop_add_nameshop_container_bg"
                            CornerRadius="20"
                            Color="#DAECEE"
                            AbsoluteLayout.LayoutBounds="0.5,0.151,0.91,0.05"
                    AbsoluteLayout.LayoutFlags="All"  />
                <Entry x:Name="shop_add_nameshop_container"
                            Background="shop_add_nameshop_container.png"
                       Placeholder="Введіть назву магазину"
                            AbsoluteLayout.LayoutBounds="0.5,0.155,0.89,0.05"
                    AbsoluteLayout.LayoutFlags="All" />
                <!--ТЕКСТ АДРЕСА МАГАЗИНУ-->
                <Label x:Name="shop_add_addressshop_text"
                            FontFamily="IBM Plex Sans"
                            HorizontalTextAlignment="Center"
                            FontSize="20"
                            TextColor="#000000"
                            AbsoluteLayout.LayoutBounds="0.5,0.215,0.9,0.05"
                    AbsoluteLayout.LayoutFlags="All"
                            Text="*Адреса магазину:"/>
                <!--КНОПКА ОБРАТИ НА КАРТІ-->
                <ImageButton x:Name="shop_add_address_button"
                            Source="shop_add_address_button.png"
                    BackgroundColor="Transparent"
                    Clicked="shop_add_address_button_Clicked"
                            AbsoluteLayout.LayoutBounds="0.5,0.255,0.9,0.07"
                    AbsoluteLayout.LayoutFlags="All" />
                <BoxView x:Name="shop_add_addressstreat_container"
```

```
                                CornerRadius="20"
                                Color="#DAECEE"
                                AbsoluteLayout.LayoutBounds="0.5,0.38,0.91,0.16"
                            AbsoluteLayout.LayoutFlags="All" />
                        <Editor x:Name="shop_add_adress" Placeholder="Введіть адресу"
                            AbsoluteLayout.LayoutBounds="0.5,0.38,0.90,0.17"
                            AbsoluteLayout.LayoutFlags="All"/>
                        <ImageButton x:Name="shop_add_logoadditem_button"
                                Source="shop_add_logoadditem_button.png"
                            BackgroundColor="Transparent"

                                AbsoluteLayout.LayoutBounds="0.5,0.54,0.90,0.17"
                            AbsoluteLayout.LayoutFlags="All" />
                        <BoxView x:Name="shop_add_nameitem_container"
                                CornerRadius="20"
                                Color="#DAECEE"
                                AbsoluteLayout.LayoutBounds="0.16,0.63,0.71,0.065"
                            AbsoluteLayout.LayoutFlags="All" />
                        <!--<Entry x:Name="shop_add_nameitem_text"
                                FontFamily="IBM Plex Sans"
                                FontSize="20"
                                TextColor="#000000"
                                AbsoluteLayout.LayoutBounds="0.2,0.63,0.67,0.065"
                            AbsoluteLayout.LayoutFlags="All"
                                Placeholder="*Назва товару:"/>-->
                        <control:AutoSuggestBox x:Name="shop_add_nameitem_text"
                                TextColor="#000000"
TextChanged="shop_add_nameitem_text_TextChanged"
QuerySubmitted="shop_add_nameitem_text_QuerySubmitted"
                                AbsoluteLayout.LayoutBounds="0.2,0.63,0.67,0.065"
                            AbsoluteLayout.LayoutFlags="All"
                                PlaceholderText="*Назва товару:"/>
                        <ImageButton x:Name="shop_add_plate_button"
                                Source="shop_add_plate_button_2.png"
                            BackgroundColor="Transparent"
                            Clicked="shop_add_logoadditem_button_Clicked"
                                AbsoluteLayout.LayoutBounds="0.94,0.635,0.16,0.1"
                            AbsoluteLayout.LayoutFlags="All" />
                        <ImageButton x:Name="shop_add_addtemplateitems_button"
                            BackgroundColor="Transparent"
                                Source="shop_add_addtemplateitems_button.png"
                                AbsoluteLayout.LayoutBounds="0.5,0.78,0.90,0.17"
                            AbsoluteLayout.LayoutFlags="All" />
                        <ImageButton x:Name="shop_add_lookadditems_button"
                                Source="shop_add_lookadditems_button.png"
                            BackgroundColor="Transparent"
                            Clicked="shop_add_lookadditems_button_Clicked"
                                AbsoluteLayout.LayoutBounds="0.5,0.91,0.90,0.17"
                            AbsoluteLayout.LayoutFlags="All" />
                        <ImageButton x:Name="shop_add_next_button"
                                Source="shop_add_next_button.png"
                            BackgroundColor="Transparent"
                                AbsoluteLayout.LayoutBounds="0.82,1.03,0.27,0.17"
                            AbsoluteLayout.LayoutFlags="All"
                            Clicked="shop_add_next_button_Clicked"/>
                        <!--ВІДМІНИТИ ДОДАВАННЯ-->
                        <ImageButton x:Name="shop_add_cancel_button"
                                Source="shop_add_cancel_button.png"
                            BackgroundColor="Transparent"
                                AbsoluteLayout.LayoutBounds="0.2,1.03,0.27,0.17"
                            AbsoluteLayout.LayoutFlags="All"
                            Clicked="shop_add_cancel_button_Clicked"/>
                    </AbsoluteLayout>
                </ScrollView>
            </ContentPage.Content>
</ContentPage>
```

Додаток М

Текст файлу Shop_Add_Watchhandadditems.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="NavigatShopAssistant.Shop_Add_Watchhandadditems">
    <ContentPage.Content>
        <ScrollView>
            <AbsoluteLayout>
                <AbsoluteLayout
x:Name="page_shop_add_watchhandadditems_screen_5_5_done_ek1"
                         AbsoluteLayout.LayoutBounds="0,0,390,844"
                         >
                    <!--ФОН-->
                             <Image x:Name="shop_add_whai_wall_4"
                             Source="shop_add_whai_wall_4.png"
                             AbsoluteLayout.LayoutBounds="0,0,390,845" />

                    <Image x:Name="shop_add_whai_logo_icon"
                             Source="shop_add_whai_logo_icon.png"
                             AbsoluteLayout.LayoutBounds="170,61,50,34.09" />
                    <Image x:Name="shop_add_whai_back_button"
                             Source="shop_add_whai_back_button.png"
                             AbsoluteLayout.LayoutBounds="16,55,50,50" />
                    <CollectionView x:Name="itemsCollection" >
                        <CollectionView.ItemTemplate>
                            <DataTemplate>
                                <AbsoluteLayout>
                                    <Image
x:Name="shop_add_whai_selected_plate_button"

      Source="shop_add_whai_selected_plate_button.png"

      AbsoluteLayout.LayoutBounds="16,124,358,50.16" />
                                                <Image
x:Name="shop_add_whai_plate_del_button"

      Source="shop_add_whai_plate_del_button.png"

      AbsoluteLayout.LayoutBounds="310,124,64,50.26" />
                                                <Label
x:Name="shop_add_whai_plate_1_text"
                                                       FontFamily="IBM Plex Sans"
                                                       FontSize="20"
                                                       TextColor="#000000"

      AbsoluteLayout.LayoutBounds="77,125,276,62"
                                                       Text="{Binding Name}"/>
                                    </AbsoluteLayout>
                            </DataTemplate>
                        </CollectionView.ItemTemplate>
                    </CollectionView>
                    <Image x:Name="shop_add_whai_next_icon"
                             Source="shop_add_whai_next_icon.png"
                             AbsoluteLayout.LayoutBounds="62,727,266,49.74" />
                </AbsoluteLayout>
            </AbsoluteLayout>
        </ScrollView>
    </ContentPage.Content>
</ContentPage>
```