КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота
на здобуття освітнього рівня бакалавра

за спеціальністю 121 Інженерія програмного забезпечення

на тему:

РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ ДЛЯ СМАРТФОНІВ. ІНФОРМАЦІЙНИЙ ДОДАТОК ДЛЯ ПОДОРОЖЕЙ УКРАЇНОЮ – UKRAINE GUIDE.

Виконав студент 4-го курсу
Станіслав ДЗУНДЗА

_____
(підпис)

Науковий керівник:
асистент, кандидат фіз.-мат. наук
Костянтин ЖЕРЕБ

_____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем
«___» _____2021р.,
протокол № ____
Завідувач кафедри
Олександр ПРОВОТАР _____
(підпис)

Київ – 2021

РЕФЕРАТ

Обсяг роботи 65 сторінок, 33 ілюстрації, 1 таблиця, 47 джерел посилань.

ANDROID, API, APP STORE, C++, IOS, JAVA, OBJECTIVE-C, PLAY MARKET, QML, QT, SWIFT, UML, МОБІЛЬНИЙ ДОДАТОК, МАГАЗИН МОБІЛЬНИХ ДОДАТКІВ.

Об’єктом дослідження є процес розробки мобільних кросплатформених додатків. Предмет дослідження є мобільний кросплатформений додаток для організації туристичних подорожей Україною.

Метою роботи є розробка мобільного додатку для спрощення організації власних туристичних подорожей Україною.

Для розробки даної програми використовувалась мова програмування C++, набір бібліотек Qt та декларативна мова QML. Перед безпосереднім створенням мобільного додатку був розроблений приблизний дизайн його основних вікон. Також була створена його архітектура, використовуючи UML діаграми.

Результат роботи: проаналізовано вплив мобільних технологій на туристичну галузь. Сформовані вимоги до мобільного додатку організації подорожей по Україні. На основі визначених вимог було проведене об’єктно-орієнтоване проектування мобільного додатку, створена його архітектура та спроектований графічний інтерфейс. Проведений аналіз підходів до розробки мобільних додатків та обрані мова C++ та Qt як інструментальні засоби реалізації кросплатформеного додатка. Розроблений кросплатформений мобільний додаток «Ukraine Guide» для спрощення організації подорожей по Україні та проведене його ручне тестування.

Розроблений програмний продукт може використовуватись будь-ким, хто володіє пристроєм під управлінням операційної системи Android чи iOS.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ 4
ВСТУП..... 5
РОЗДІЛ 1 РОЛЬ МОБІЛЬНИХ ДОДАТКІВ У СУЧАСНОМУ СУСПІЛЬСТВІ7
1.1. Сучасні операційні системи для смартфонів та їх особливості 7
1.2. Мобільні додатки в інформаційному суспільстві..... 10
1.3. Використання мобільних додатків у туристичній галузі..... 13
РОЗДІЛ 2 ОСОБЛИВОСТІ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ СМАРТФОНІВ 16
2.1. Мобільні додатки для OS Android. Особливості, методи та засоби для їхньої розробки 16
2.2. Мобільні додатки для iOS. Особливості, методи та засоби для їхньої розробки 24
2.3. Кросплатформені мобільні додатки..... 24
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ “UKRAINE GUIDE”..... 33
3.1. Формування специфікації для додатку..... 33
3.2. Розробка користувацького інтерфейсу 43
3.3. Вибір технологій для розроблюваного додатку, їхня характеристика..... 48
3.4. Наповнення інформацією та тестування готового додатку..... 50
3.5. Порівняння розробленого додатку з існуючими аналогами..... 50
ВИСНОВКИ..... 60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ..... 62

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

AVD – Android Virtual Device

API – Application Programming Interface, прикладний програмний інтерфейс;

IDE – Integrated Development Environment, інтегроване середовище розробки;

HTML – HyperText Markup Language, мова розмітки гіпертексту;

JSON – JavaScript Object Notation;

NDK – Native Development Kit;

OEM – Original Equipment Manufacturer;

QML – Qt Makup Language;

UML – Unified Modeling Language, уніфікована мова моделювання;

ОС – Операційна система;

НП – Населений пункт.

ВСТУП

Оцінка сучасного стану об’єкта розробки. Останнім часом популярність мобільних технологій та додатків постійно зростає і, безсумнівно, вносить значний внесок у повсякденне життя користувачів. Нова ера інформаційно-комунікаційних технологій відкрила безліч нових інструментів для туристичної галузі.

В інноваційному туризмі широко використовуються інформаційні технології (наприклад, Інтернет, мобільний зв’язок та доповнена реальність) для збору величезних обсягів даних та надання підтримки в режимі реального часу всім зацікавленим особам.

Сучасний інноваційний туризм передбачає надання туристичних послуг через веб-додатки з можливостями геолокації; зміна ролі туристів як співвиробників туристичного контенту; підвищення власного туристичного досвіду за допомогою нових технологій (доповнена реальність); підключення та взаємодія з місцевими громадами та іншими туристами у місці призначення.

У інноваційному туризмі технологія – це інфраструктура, яка об’єднує апаратні, програмні та мережеві технології для надання послуг у режимі реального часу, що дозволяє більш розумно приймати рішення в певних ситуаціях. Мобільні технології, особливо використання смартфонів та їх застосунків, мають значний вплив на розвиток інноваційного туризму.

Актуальність роботи та підстави для її виконання. Все більше споживачів покладаються на свої мобільні пристрої в туристичних цілях. Якщо раніше люди постійно звертались з даним запитом до туристичних агентств, то зараз вони хочуть самостійно планувати свої маршрути, оцінювати їх привабливість. І це вони можуть зробити без проблем, оскільки мають доступ до величезного об’єму інформації, на відміну від давніших часів. Тому додатки для мобільних пристроїв, що дають змогу отримати структуровану та корисну інформацію про місця майбутніх подорожей будуть користуватися великим попитом. Поки що в Україні таких не дуже багато, крім того більшість з них працюють під управлінням лише однієї операційної системи (ОС). Отже,

створення кросплатформеного додатку є задачею актуальною та практично необхідною.

Мета й завдання роботи. Метою роботи є розробка мобільного додатку для організації власних туристичних подорожей Україною.

Для досягнення мети потрібно вирішити такі задачі:

- проаналізувати вплив мобільних технологій на туристичну галузь;
- проаналізувати аналоги, що представлені на ринку програмних продуктів;
- проаналізувати принципи розробки мобільних додатків для різних ОС;
- провести об’єктно-орієнтоване проектування мобільного додатку для організації туристичних подорожей;
- розробити мобільний додаток для організації туристичних подорожей;
- провести мануальне тестування готового додатку.

Об'єкт, методи й засоби розроблення. Об’єктом дослідження є процес розробки мобільних кросплатформених додатків. Предмет дослідження є мобільний кросплатформений додаток для організації туристичних подорожей Україною. Методи дослідження: структурний аналіз та проектування інформаційних систем, моделювання з використанням мови UML, методи об’єктно-орієнтованого аналізу, проектування і програмування.

Можливі сфери застосування. Практична значущість роботи полягає в розробці зручного мобільного додатку, що може використовуватись споживачами в інформаційних цілях з метою організації подорожей по населених пунктах України.

РОЗДІЛ 1 РОЛЬ МОБІЛЬНИХ ДОДАТКІВ У СУЧАСНОМУ СУСПІЛЬСТВІ

1.1. Сучасні операційні системи для смартфонів та їх особливості

Одними з найбільш широко використовуваних сьогодні технологій є саме мобільні технології. Зростання пристроїв, які їх використовують, відбувається експоненційно, що суттєво впливає на життя людей [1, 2].

Смартфон перестає відігравати роль пристрою, що використовувався для дзвінків та передачі голосових повідомлень, а переростає у все більш досконалі комп'ютери загального призначення, і це призвело до розвитку різноманітних платформ та операційних систем у мобільному просторі.

Сьогодні операційна система (ОС), на основі якої працює мобільний пристрій, є настільки ж важливою, як і виробник пристрою (OEM). Різні функції, графічний інтерфейс, швидкість обробки та багато інших речей, залежатимуть від базової ОС. Мобільна ОС управляє апаратними та програмними ресурсами мобільного пристрою, подібно до комп'ютерної ОС [3, 4].

Враховуючи, що на сучасних телефонах буде працювати все більше програмного забезпечення, такого як інтернет-браузери, навігаційні пакети, ігри та музичні / відеоплеєри, прикладні платформи повинні забезпечити основу для них. Прикладні платформи побудовані на операційних системах ядра нижчого рівня, таких як Rex, Linux. Програмне забезпечення, що працює на пристрої, отримує доступ до ресурсів платформи через набір інтерфейсів прикладного програмування або API. Щоб максимізувати доступний ринок, розробник мобільних пристроїв, як правило, розробляє додатки для декількох платформ операційної системи, але це може бути складним і трудомістким завданням, оскільки різні платформи використовують різні виклики API для загальних операцій ОС та доступу до ресурсів ОС, таких як пам'ять та файли.

Зараз існує значна кількість мобільних ОС. За популярністю до 90% всього ринку займають дві основних ОС: Android та iOS (рис. 1.1).

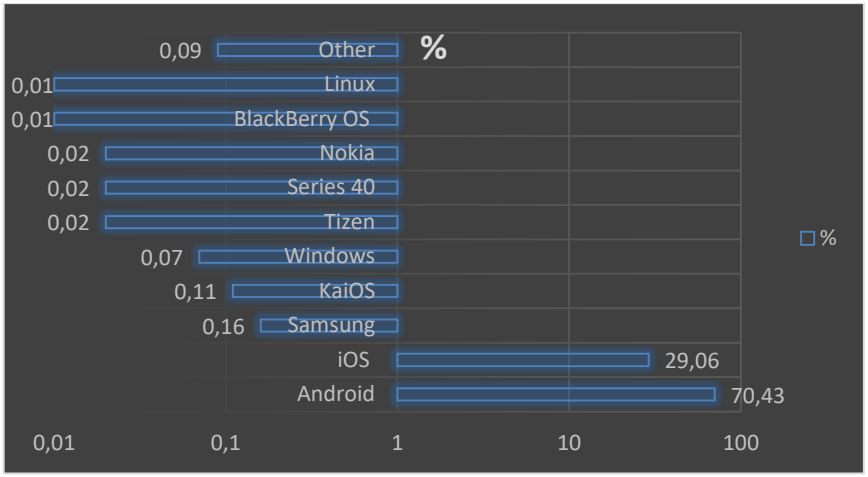

Рисунок 1.1 – Поширення мобільних операційних систем у світі (2020 р.) [6]

Операційні системи для мобільних пристроїв мають свої особливості:

– врахування більш жорстких обмежень по пам'яті та більш низької швидкості процесора. Незважаючи на те, що мобільні пристрої знаходяться на етапі бурхливого розвитку, за своїми параметрами вони все поки поступаються комп'ютерам. Тому додатки для них, що вимагають великого обсягу пам'яті, відтворюються з неповними можливостями. Такі інструменти як, наприклад, Java, поширюються для мобільних пристроїв в спеціальних версіях, які були розроблені з врахуванням певних обмежень;

– врахування особливостей екранів і екранних навігаторів певних гаджетів. Багато типів смартфонів різних виробників мають екрани, що істотно відрізняються, та різноманітні види екранних навігаторів (трекбол та ін.). Ці відмінності також доводиться враховувати, що ускладнює розробку програмного забезпечення;

– підтримка комунікаційних і мережевих технологій: Wi-Fi / WiMAX, Bluetooth, GPRS, EVDO, GSM, CDMA. Мобільний пристрій і його ОС повинні надавати користувачу доступ до сучасних видів комунікації в бездротових мережах [4].

Зупинимось на особливостях основних мобільних операційних систем, а саме Android та iOS.

Android – це операційна система з відкритим кодом для мобільних пристроїв, розроблені Google та Open Handset Alliance [7].

Системна архітектура складається з [8]:

– модифіковане ядро Linux, на якому базуються такі системні сервіси, як безпека, управління пам’яттю, мережевий стек і модель драйверів;

– бібліотеки з відкритим кодом на мовах C і C++, які використовуються різними компонентами системи Android;

– Android Runtime, яка управляє ключовими бібліотеками, які представляють більшу частину функціональності;

– фреймворк додатків, що управляє сервісами та бібліотеками;

– програми, які працюють в цій ОС.

У середовищі виконання код програми виконується з повними дозволами та має доступ до важливих системних ресурсів. З іншого боку, код програми виконується всередині обмеженого простору, що називається пісочницею.

Перевагами даної ОС є багатозадачність, простота доступу до великої кількості додатків. Недоліком, відповідно, є наявність значної кількості неперевірених додатків, які можуть мати шкідливий вміст.

Мобільна операційна система iOS використовується лише для пристроїв Apple, і є подальшим розвитком операційної системи Mac OS X. Архітектура системи ідентична архітектурі Mac OS X і складається з наступних компонентів [9, 10]:

- Основна ОС: ядро операційної системи;
- Ключові сервіси, які підрозділяються на різні фреймворки, що базуються на C та Objective C. Наприклад, фреймворк CFNetwork , що забезпечує роботу з відомими мережевими протоколами;
- Медіа: містить фреймворки високого рівня, які відповідають за використання графічних, аудіо- та відео- технологій;
- Cocoa Touch: включає UIKIT, який є фреймворком розробленим на Objective C, що забезпечує ряд можливостей, необхідних для розробки iOS додатків, таких як управління інтерфейсом користувача.

У мобільних пристроях на ОС iOS апаратне забезпечення відноситься до фізичних мікросхем, припаяних до схеми гаджета. Прошивка стосується специфічного коду мікросхеми, який або міститься в пам’яті в периферії, або всередині накопичувача для зазначеної периферії. Процесор посилається на набір інструкцій ARM та таблицю дескрипторів переривань, що налаштовуються під час процесу завантаження ОС. iOS являє собою ядро, драйвери та сервіси, що знаходяться між користувальницьким простором та обладнанням.

Середовище виконання складається з бібліотек динамічного зв’язку, а також базових бібліотек C. Фреймворки мають виклики API, які розповсюджуються за допомогою IPhone SDK. Програму, що зберігається в iPhone, слід придбати через App Store. Ця програма попередньо компілюється. Додаток працює повністю в середовищі користувача.

Перевагами даної ОС є відсутність шкідливих програм, забезпечення інформаційної безпеки завдяки використанню власних сертифікатів. Недоліками є залежність від апаратного забезпечення Apple (процес схвалення додатків є в основному чорним ящиком для розробників).

1.2. Мобільні додатки в інформаційному суспільстві

Головними рисами мобільних технологій є можливість зв’язатися з користувачами незалежно від того, де вони знаходяться та доступність до різноманітних послуг в будь-який час з будь-якої точки земної кулі. Тому основними перевагами мобільних послуг зазвичай називають такі чотири характеристики: мобільність, досяжність, локалізація і ідентифікація.

Мобільність – одна з основних особливостей сучасних технологій. Бездротові пристрої дозволяють користувачам отримати доступ до необхідної інформації практично з будь-якого місця.

Завдяки інфраструктурі локалізації місцезнаходження будь-якої людини, що має при собі мобільний пристрій, може бути легко встановлено. Така здатність визначення місця розташування користувачів є дуже важливою,

наприклад для служб, що надають інформацію про транспортні пробки, найближчі місця паркування, готелі або ресторани, історичні та туристичні пам’ятки.

У 2017 році у світі було 8,4 млрд. мобільних абонентів (що майже сягає кількості населення світу), 3,9 млрд. користувачів Інтернету (рис. 1.2).

Рисунок 1.2 – Зростання глобальної Інтернет-аудиторії за результатами дослідження компанії Kleiner Perkins Caufield & Byers [11]

Широку популярність серед користувачів мобільних пристроїв отримують мобільні додатки. Інтернет-ресурс App Annie повідомляє: у 2020 році кількість завантажень додатків збільшилася на 9% (на 18 млрд.), а загальна кількість завантажень за 2019 рік склала більше 200 мільярдів [12].

Як стверджувалося раніше, збільшення числа розроблених мобільних додатків безпосередньо пов’язано зі зростанням швидкості мобільного інтернету. А даний факт, у свою чергу, посприяв збільшенню часу проводження за смартфоном. Портал Hootsuite стверджує, що середньостатистичний користувач проводить в інтернеті з мобільних пристроїв в середньому 3 години 14 хвилин в день [13].

Мобільний додаток – це програмне забезпечення, розроблене спеціально для мобільних телефонів, смартфонів, планшетів, смарт-годинників чи інших гаджетів [14, 15]. Мобільні додатки дозволяють вирішити прикладні завдання для користувачів: отримання геоданих, перевірка електронної пошти, спілкування тощо. Крім виконання прикладних завдань, вони ще й виконують розважальну функцію. На ринку мобільних додатків величезний вибір ігор

різного роду. Саме мобільні додатки надають можливості зручно користуватися тими чи іншими послугами. Вони поділяються на декілька категорій, виходячи з того, для якої цільової аудиторії розробляються, які цілі переслідують, як будуть реалізовані.

Наведемо основні категорії за напрямком використання:

а) Розваги:
   1) ігри (стратегії, головоломки тощо);
   2) замовлення квитків в кіно, театр, на виставку тощо;
   3) програми для дітей. Все, що може зацікавити дитину: ігри, книги, мультфільми, музика, завдання та головоломки та інші розваги;

б) Подорожі:
   1) бронювання номеру в готелі, купівля квитків на транспорт;

в) Бізнес:
   1) програми для фінансових організацій і банків;
   2) онлайн-продажі представлені у вигляді аукціонів, розпродажів;
   3) програми для міста допомагають зорієнтуватися в мегаполісі, знайти потрібний об’єкт, прокласти маршрут, паркуватися і багато іншого;

г) Соціальні програми:
   1) соціальні мережі, які зручні для швидкого спілкування й обміну інформацією, перегляду новин і повідомлень;

д) Харчування:
   1) додатки, що забезпечують замовлення і доставку їжі з можливістю виставлення оцінок та залишення відгуків;

е) Освіта:
   1) інтерактивні навчальні курси;
   2) газети, журнали та інші ЗМІ.

Продажі додатків для iOS і Android у 2020 році зросли на 25% і досягли 112 млрд. дол. Про це свідчать дані платформи App Annie [12]. Згідно з

оприлюдненою статистикою, 81 млрд. дол. в 2020 році було витрачено по всьому світу на ігри і 31 млрд. дол. на інші програми. На покупки App Store прийшлося 65% виручки, а витрати в Google Play виросли майже на 30% [12, 13]. Зазвичай подібне зростання є характерним для ринків, що розвиваються, але в 2020 році пандемія коронавірусу COVID-19 стимулювала попит на мобільні рішення для роботи, навчання і розваг [12]. Найбільше часу користувачі в усьому світі витратили в категорії «Соціальні мережі та комунікації» – 1,5 трлн годин, на «Розваги і відео» прийшлося 879 млрд. годин, на «Ігри» – 284 млрд., а на «Шопінг» – 80 млрд. годин. Статистика зростання кількості мобільних додатків в період з 2018 по 2020 роки зображена на рис. 1.3.

Рисунок 1.3 – Зростання кількості мобільних додатків за результатами дослідження компанії Sensor tower [16]

Основними перевагами мобільних додатків, що сприяли бурхливому зростанню їх поширення, є інтерактивність, багатофункціональність, персоналізація.

1.3. Використання мобільних додатків у туристичній галузі

В туристичній галузі ще на початку 2000-х рр. з'явилися перші цифрові сервіси для туристів, ними стали системи бронювання житла та купівлі авіаквитків квитків, такі як Booking та Aviasales.

Сучасний туризм досить важко уявити без використання інформаційних технологій. Цифровий туризм здійснює свій розвиток в різноманітних формах, однією з яких є онлайн-покупка спроектованих турів. У переліку інтернет-

ресурсів, що пропонують подібні послуги, можна назвати пошукову систему туристичної агенції «Поїхали з нами», «ТурПошук», «Tonline» та ін. Пошукові системи позиціонують себе як ресурси, що дозволяють реалізувати пошук відповідних турів за пропозицією туроператорів на підставі заданих критеріїв, враховуючи особисті переваги і побажання клієнтів.

Однією з сучасних форм цифровізації туризму є розробка мобільних додатків, корисних для планування та здійснення туристичної подорожі. Так наприклад, послуги карпулінга потенційні споживачі шукали зі смартфонів в 2018 році частіше, ніж в 2017 році на 62%, запити на пошук залізничних квитків з мобільних пристроїв збільшилися в 2018 р порівнянні з 2017 р на 50%. Зрозуміло, що в 2020 відбувся спад таких запитів в зв'язку з пандемією. Але, враховуючи тенденції попередніх років та зростання популярності мобільних додатків, можна з впевненістю стверджувати про зростання таких запитів в майбутньому [17].

Варто відзначити, що придбання турів за допомогою мобільних пристроїв є не настільки популярною послугою, що цілком зрозуміло, і покупка туристичних продуктів є більше популярною з використанням персонального комп'ютера. Однак мобільні додатки є незамінними помічниками в отриманні довідкової інформації про місцезнаходження фізичної особи, про об'єкти, що його оточують в місці тимчасового перебування. Йдеться, наприклад, про Google Maps, сервіс «Місцеві експерти», що дозволяє користувачам додавати фотографії і опис місць, залишати відгуки, тим самим спрощуючи знайомство як місцевого населення, так і приїжджих з об'єктами певної території [18]. Також Google Maps надають користувачам можливість побудувати маршрут з однієї точки в іншу з використанням особистого автомобіля, громадського транспорту або пішки, що дуже зручно і корисно в разі подорожі по малознайомим місцевостям [2,17].

Використання мобільних додатків для організації самодіяльних турів також отримує розповсюдження. Це пов'язано з поширенням концепції Web 2.0. Сучасні користувачі Інтернету стають співучасниками формування

цифрового контенту, що дає змогу споживачам туристичних послуг самостійно здійснювати планування своїх туристських поїздок, не вдаючись до допомоги турагенств. Крім того, передумовою цього виду діяльності є зростання кількості власників автотранспорту.

У зв’язку з цим на ринку додатків для мандрівників почали з’являтися гіди – зручний варіант для самостійного знайомства з місцевими пам’ятками. У порівнянні зі своїми традиційними конкурентами такі гіди мають ряд незаперечних переваг. На відміну від паперових путівників в такому додатку представлена відео та аудіо інформація, що дає змогу користувачу отримувати відомості щодо тієї чи іншої пам’ятки. Досить часто в таких додатках містить інформація, яка сформована та оброблена професійними істориками та гідами. Інтерактивні путівники надають можливості геолокації і побудови маршрутів, інтеграції з соцмережами, віртуальними галереями, використання програми без постійного підключення до мережі, інтерактивне меню, включаючи відео, анімовану графіку, 3D, аудіокоментарі, кругові панорами. Можна впроваджувати в додатки маркетингові інструменти – акції, опитування, спеціальні лічильники, що дозволять проводити детальний аналіз взаємодії користувача з контентом. Можна також ввести інформацію про інфраструктурі всіх типів для людей з обмеженими фізичними можливостями.

Враховуючи сучасну ситуацію з пандемією у світі, особливу популярність починають отримувати віртуальні подорожі, що організовуються за допомогою мобільних сервісів. Користувачам пропонуються тривимірні віртуальні подорожі на природі, містом тощо. Завдяки додаткам віртуальної реальності можна познайомитися з реальними історіями мандрівників, подивитися, що вони дізналися і що дала їм ця подорож [1].

Отже, мобільні технології суттєво вплинули на розвиток туристичної галузі, змінивши підходи до організації бізнес-процесів та взаємовідношень між постачальниками туристичних послуг та їх споживачами.

РОЗДІЛ 2 ОСОБЛИВОСТІ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ СМАРТФОНІВ

2.1. Мобільні додатки для OS Android. Особливості, методи та засоби для їхньої розробки

Реалізацію мобільного додатка можна здійснити в одному з трьох варіантів: нативні, Web або кросплатформені.

Під нативним додатками розуміють мобільний додаток, який створюється для певної платформи і безпосередньо встановлюється на пристрій користувача (займаючи певний обсяг пам’яті). Такі програми користувач завантажує через магазин додатків тієї або іншої платформи, такої як Play Store для Android і Apple App Store для iOS [19].

Деякі важливі переваги нативних додатків:

– Дані дій (чи бездіяльності) користувача можуть бути легко зібрані та проаналізовані, таким чином полегшуючи оцінку ефективності всієї програми або її окремих функцій.

– Нативні додатки, як правило, працюють і «відчуваються» краще. Web-додатки іноді створюються для імітації нативних, але вони обмежуються швидкістю інтернету і можливостями дизайну.

Але, крім переваг, у таких додатків є такі недоліки [20]:

– Висока вартість і витрати часу на розробку. Без сумніву, створення окремих додатків відразу під кожну з декількох операційних систем може значно продовжити процес розробки. Один і той же програмний код не може бути розгорнутий на різних платформах, і програмістам буде потрібно більше часу для переписування коду, що збільшує витрати і час розробки. Якщо компанія хоче для кожної з платформ створювати окремі додатки, вона, можливо, повинна здійснити набір нових програмістів, що спеціалізуються в даній галузі. Наприклад, один розробник буде зосереджений на розробці додатків для iOS, а інший – на розробці додатків для Android.

– Несумісність з іншою операційною системою. Розроблюваний додаток буде несумісним з іншими ОС. Коли розробляється додаток під конкретну ОС, його розробники використовують мову, специфічну тільки для цієї операційної системи: наприклад, Objective-C або Swift – для iOS, для різних мобільних пристроїв на базі Android – Kotlin і Java. У цьому контексті нативний додаток, який спочатку написано для iOS, не буде зможе бути запущений на пристрої з ОС Android.
– Втрачені можливості. Розробка додатків, орієнтованих тільки на одну платформу, може привести до втрачених можливостей, особливо, якщо інші платформи заздалегідь не беруться до уваги. Свідоме скорочення цільового ринку може привести до втрати доходу.

Перейдемо до Web-додатків.

Web-додатки працюють через Web-браузер на пристрої користувача. Вони, по суті, є індивідуалізованими Web-сайтами, які зроблені таким чином, щоб виглядати і використовуватися як нативні додатки, але насправді вони не знаходяться на пристрої користувача. При якісній розробці Web-додатки часто працюють подібно до нативних [19,20].

Переваги Web-додатків:

– Легше підтримуються і можуть функціонувати на пристрої з будь-якою ОС;

– Розробники можуть пропонувати рішення без необхідності їх затвердження будь-якими магазинами додатків;

– Швидкість розробки.

Недоліками даного виду додатків є:

– Відсутність доступу до пристрою користувача. Це обмежує багато функцій, які використовуються для більш персоналізованого застосування;
– Необхідність постійного підключення до мережі інтернет;
– Пошук програми може бути складним, тому що не існує магазину веб-

додатків з каталогом і функцією пошуку в ньому.

Насамкінець, розглянемо кросплатформені додатки.

Кросплатформені додатки можуть працювати в різних операційних системах. Після написання коду програми її можна розгорнути на різних пристроях і платформах, не турбуючись про проблеми несумісності. Це універсальний підхід, який широко використовується для економії часу і грошей на розробку [19,20].

Переваги кросплатформених додатків:

- Один код доступний для повторного використання на інших платформах. Основною перевагою кросплатформеної розробки мобільних додатків є той факт, що один і той же код може використовуватися на різних платформах. Повторне використання коду дозволяє легко розгортати додаток на іншій платформі, оскільки можливості програми, реалізовані на одній платформі, будуть працювати і на інших;
- Розробка кросплатформених додатків економічно ефективна. Одна команда може реалізувати потрібну ідею відразу на всіх платформах, використовуючи єдиний стек технологій. Це призводить до менших витрат ресурсів;
- Просте і швидше розгортання. Розробникам кросплатформених додатків не потрібно вивчати кілька технологій різних платформ перед створенням своїх додатків. Оскільки немає необхідності створювати різні кодові бази, початкове розгортання на цільових платформах відбувається набагато швидше. Також майбутні зміни в коді програми можуть виконуватися одночасно, без змін на кожній платформі;
- Кросплатформені додатки покривають більш широку аудиторію Кросплатформені додатки пропонують розробникам більше можливостей для охоплення ширшої аудиторії, оскільки такі програми досягають користувачів всіх типів мобільних пристроїв, незалежно від

їх ОС. Це значно рентабельніше для бізнесу, ніж присутність додатку тільки на одній платформі;

– Кросплатформені додатки допускають однаковий інтерфейс і UX. Тоді як продуктивність важлива для будь-якого мобільного додатка, його зовнішній вигляд (UI) і досвід користування (UX) так само важливі. Використання єдиної спільної команди розробників і єдиного коду дозволяє компаніям використовувати однаковий зовнішній вигляд програми на всіх платформах [21; 22; 24].

Недоліки кросплатформеної розробки додатків [21; 22]:

– Кросплатформені додатки не є такими гнучкими, як нативні додатки. Хоча функції програми будуть реалізовуватися на всіх платформах, швидше за все готовий додаток не можливо буде адаптувати для використання максимальних можливостей кожної з платформ. Робота з уніфікованим стеком технологій не забезпечить такої ж гнучкості налаштування і оптимізації, як застосування стека технологій, індивідуального для кожної ОС;
– Кросплатформені програми не працюють так само добре, як нативні додатки. Внаслідок використання одного універсального стека технологій зменшується гнучкість додатка. Однак втрата гнучкості в розробці означатиме втрату можливості покращити продуктивність;
– Відправка кросплатформених додатків до відповідних магазинів може ускладнитися. Механізм додавання додатка, що є кросплатформеним, в App Store і в Google Play Store буде відрізнятися. Вимоги цих магазинів до програм у них різні. Проходження всіх перевірок і виконання всіх правил для відповідності обом магазинам будуть викликати певні складності.

Перейдемо до розгляду принципів розробки різних мобільних додатків. При розробці додатку під Android необхідно враховувати деякі особливості.

Перш за все, це велика різноманітність пристроїв з різними характеристиками (наприклад: розмір та роздільна здатність екрана, наявність та розташування фізичних кнопок тощо).

По-друге, наявність великої кількості різних версій операційної системи (рис. 2.1). Так кількість гаджетів з застарілими версіями Android займає значну частку ринку [23].

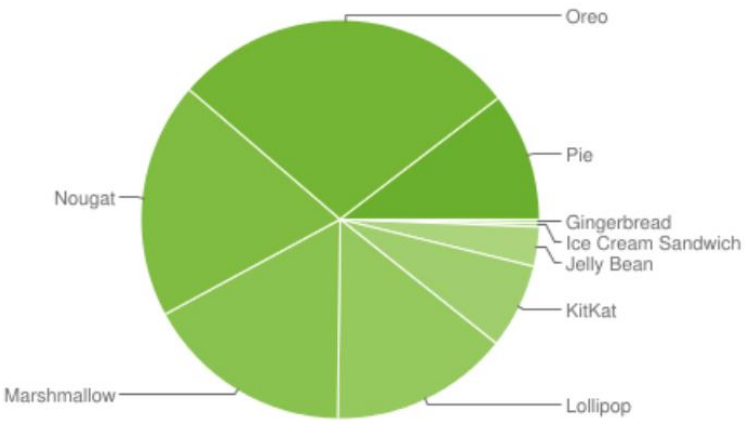

Рисунок 2.1 – Розподіл версій Android станом на 2020 рік за даними Google [26]

Крім того, необхідно враховувати архітектуру системи. В Android додаток збирається з логічно самостійних і відокремлених частин – Activity та фрагментів, які є будівельними блоками програми [21]. Activity (або діяльність) є найважливішим компонентом програми для Android. На відміну від парадигм програмування, в яких програми запускаються методом main(), система Android ініціює код у екземплярі Activity, викликаючи конкретні методи зворотного виклику, що відповідають певним етапам його життєвого циклу [24].

Досвід роботи з мобільним додатком відрізняється від настільного комп'ютера тим, що взаємодія користувача з додатком не завжди починається в одному місці. Натомість подорож користувача часто починається недетерміновано. Наприклад, якщо ви відкриваєте програму електронної пошти з головного екрану, то можете побачити список електронних листів. На відміну від цього, якщо ви використовуєте програму соціальних мереж, яка потім запускає вашу програму електронної пошти, ви можете перейти безпосередньо

на екран програми електронної пошти для складання листа [25].

Клас Activity забезпечує дотримання цієї парадигми. Коли одна програма викликає іншу, перша програма викликає діяльність в другій програмі. Таким чином, діяльність служить точкою входу для взаємодії програми з користувачем. Activity надає вікно, в якому програма відображає свій інтерфейс. Це вікно зазвичай заповнює екран, але може бути меншим за нього і знаходитись поверх інших вікон. Як правило, одна Activity реалізує один екран у програмі. Також одна з них вказується як основна, тобто є першим екраном, що з'являється, коли користувач запускає програму. Потім кожна Activity може розпочати іншу Activity, щоб виконати різні дії. Незважаючи на те, що всі вони працюють разом, щоб забезпечити роботу користувачів у програмі, кожна Activity лише слабо пов'язана з іншими.

Фрагмент являє собою багаторазову частину інтерфейсу додатка. Фрагмент визначає і управляє власним макетом, має власний життєвий цикл і може обробляти власні події введення. Фрагменти не можуть жити самі по собі – їх повинні розміщувати або на Activity, або на іншому фрагменті [23, 24]. Фрагменти вносять в інтерфейс Activity модульність та можливість повторного використання.

Життєвий цикл додатка в системі Android зображено на рис. 2.2. При запуску компонента, система запускає процес додатка, якому належить компонент, якщо той ще не працює, і створює екземпляри класів, необхідних компоненту. Оскільки відбувається запуск кожної програми в окремому процесі і існують обмеження на доступ до файлів, додаток не може напряму активувати компонент іншої програми. Таким чином для його активації необхідно надіслати системі повідомлення про намір запустити певний компонент [25]. Такий підхід був розроблений для того, щоб забезпечити роботу додатків на абсолютно будь-яких мобільних пристроях, в тому числі з дуже малим об'ємом оперативної пам'яті і дуже слабкими процесорами. Якщо частини програми незалежні, будь-яку з них можна в потрібний момент викинути з пам'яті і не витрачати на підтримку її життєвого циклу дорогоцінні

ресурси.

Наприклад, ви бачите список ресторанів, потім натискаєте на якийсь елемент і провалюєтесь в нього. Другий екран, картка ресторану, нічого не повинна знати про попередній список, тому що в будь-який момент часу, в тому числі відразу після переходу в картку, він може бути вивантажений з оперативної пам’яті і знищений. Це станеться, наприклад, якщо в фоновому режимі запущено багато процесів [24].

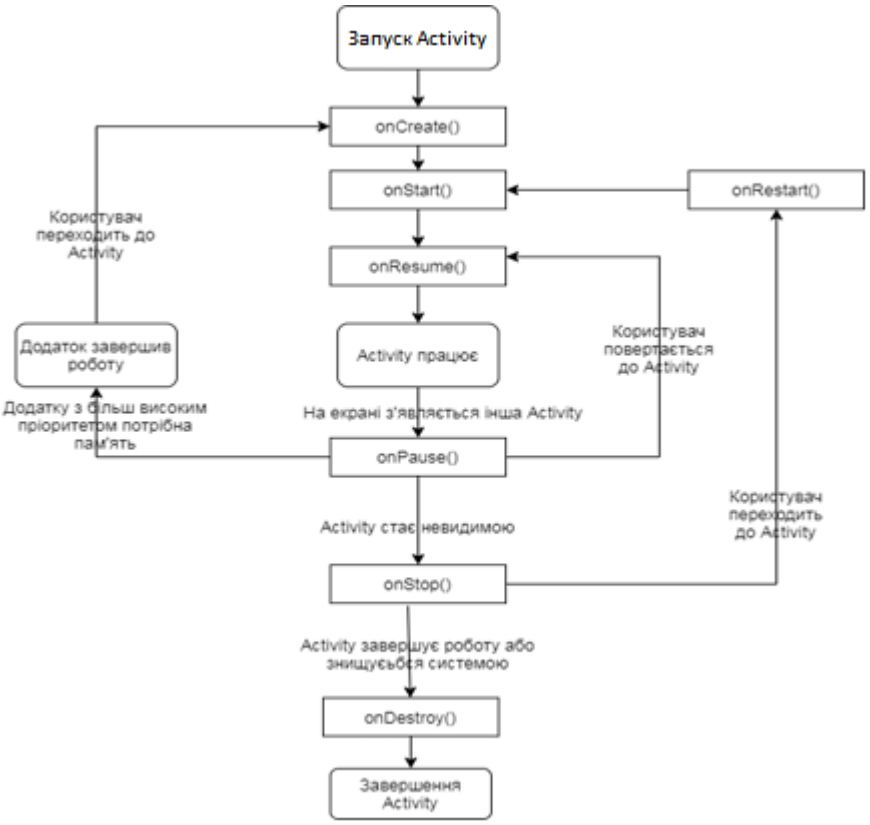

Рисунок 2.2 – Життєвий цикл додатка на Android [23]

Основними мовами, що використовуються при розробці Android-додатків є Java, Kotlin, C++.

Java є офіційною мовою програмування, що підтримується середовищем розробки Android Studio [27, 28]. На Java посилається більшість офіційної документації Google, також є велика кількість платних та безкоштовних бібліотек. При розробці на Java використовуються не тільки Java код, але також і файли маніфесту на мові XML, що надають системі основну інформацію про програму.

Мова Kotlin була офіційно представлена Android-спільноті в травні 2017

року на Google I/O. Вже в 2019 році Google визнала його кращою мовою програмування для Android-додатків, відсунувши Java на другий план. Це в тому числі означає, що створення нових стандартних інструментів (бібліотек, функцій Android Studio) буде в першу чергу націлено на Kotlin [27]. Kotlin включив в себе найкраще з сучасних мов програмування, таких як Java, Scala, TypeScript. Серед основних переваг – автоматичне виявлення типів даних, підтримка функціональної парадигми, усунення ряду найпоширеніших помилок, що виникають при компіляції програм на Java.

C++ дозволяє створювати нативні додатки, що може стати в нагоді для створення ігор або інших ресурсоємних програм. Застосування Android NDK (Native Development Kit) означає, що програма запускається не через Java Virtual Machine, а безпосередньо через пристрій, що надає розробнику більше контролю над такими елементами системи, як пам’ять, сенсори, жести тощо [28].

Зараз існує безліч інструментів для розробки Android-додатків.

Android Studio є офіційним IDE (інтегрованим середовищем розробки) для Android, створене компанією Google [27]. Він включає в себе компілятор, який може створювати APK файли, і організовує файлову систему для проекту. Крім цього він включає в себе редактор XML і розширений редактор макетів. Android Studio пропонує цілий набір додаткових інструментів. Головною перевагою даної IDE є зручний UI-редактор і автодоповнення коду. Однак даний він є досить ресурсоємний.

Інструмент AVD Manager йде в комплекті з Android Studio. Абревіатура AVD розшифровується як «Android Virtual Device», тому, по суті, це емулятор для запуску додатків Android на комп’ютері. Важливо, що AVD Manager дозволяє створювати безліч емуляторів з різними розмірами екрану, специфікаціями і версіями Android [27,29].

Крім Android Studio, існує ще багато варіантів, наприклад, Cordova або IntelliJ IDEA для створення простих додатків, або Andyroid і Blustacks, якщо потрібна емуляція Android без установки Android Studio і AVD Manager.

2.2. Мобільні додатки для iOS. Особливості, методи та засоби для їхньої розробки

На відміну від додатків для Android, додатки для операційної системи iOS розраховані на менш різноманітні варіанти як технічного оснащення гаджетів, так і версій операційної системи.

Основними мовами програмування для розробки програм на iOS є Objective-C та Swift. Обидві мови відносяться до мов об'єктно-орієнтованого програмування (ООП) [19].

Objective-C, створений на початку 1980-х років, спочатку сприймався, як проста надбудова над мовою C, що модифікує її деякі синтаксичні конструкції. Це основна мова, що використовується компанією Apple, знання якої дозволяє писати програми під будь-які платформи Apple.

Перевагами даної мови є [19]:

– велика кількість документації, технічної літератури та величезне співтовариство. Apple надає і регулярно оновлює офіційні книги і ресурси;

– швидкий перехід з однієї з мов сімейства C. Objective-C – це розширення мови C. Це означає, що будь-який код на C є також коректним кодом і для Objective-C, потрібно тільки звикнути до синтаксису;

– сумісність Objective-C всередині проектів, написаних на Swift, дозволить застосовувати дві мови одночасно.

Недоліками є [19]:

– низька читаність коду: на початку вивчення синтаксис здається складним;

– динамічна система типів даних, передбачає можливість появи помилок під час виконання програми;

– низька в порівнянні з мовою Swift продуктивність;

– взаємодія з файлами Swift відбувається за допомогою «моста» (умовний адаптер, який переводить код на Swift в формат Objective-C), що сильно гальмує процес розробки.

Мова Swift є молодою, потужною і відкритою мовою програмування загального призначення. Вона поєднує в собі все краще від C і Objective-C, але позбавлена обмежень останнього, що накладаються на догоду сумісності з C. У Swift використовується сувора типізація об'єктів. Також додані сучасні функції, такі як дженерики, замикання, множинні повернені значення і багато іншого, що перетворюють створення додатка в більш гнучкий процес. [19].

Перевагами є:

– швидкість. Зараз мова майже на одному рівні з C++, і Apple обіцяють, що це не межа;

– легка читаність коду, оскільки дана мова не побудована на C. Swift нагадує звичайний англійську мову і має спрощений синтаксис;

– лаконічність. Кількість коду зі Swift стає набагато менше.

– більші можливості в порівнянні з Objective-C.

– підвищена безпека. Swift, на відміну від Objective-C, строго типізований, тобто при оголошенні іменованих змінних потрібно явно вказувати тип даних. Це значно полегшує процес усунення помилок.

Недоліки даної мови:

– взаємодія з файлами Objective-C відбувається за допомогою «моста», який сильно гальмує процес компіляції.

Написання програм для ОС iOS здійснюється на комп'ютерах Mac компанії Apple в робочому середовищі X-Code. Його можна завантажити безкоштовно з AppStore для Mac. Xcode тісно інтегрований з framework Cocoa, який складається з бібліотек, API, і середовищ, які формують шар розробки для всіх MacOS X. Даний набір інструментів включає в себе [14]:

– Xcode IDE (середовище для написання коду, створення і налагодження додатків);

– IB (Interface Builder). IB – це додаток з набором інструментів для розробки графічних інтерфейсів, інтегрований в Xcode. Процес роботи нагадує створення дизайну "на полотні". Всю верстку можна зробити в IB, а потім зв’язати візуальні елементи з файлом реалізації, в якому описана вся логіка взаємодії з ними [19];

– Емулятор. Надає повний набір iOS-пристроїв, на яких можна запустити і протестувати свій додаток;

– Інструменти для аналізу поведінки і продуктивності;

– Десятки додаткових інструментів.

Програма в операційні системі iOS архітектурно представляє собою єдине ціле. Для реалізації мобільних додатків компанія Apple пропонує використовувати архітектуру проектування Cocoa MVC (рис. 2.3) [15].

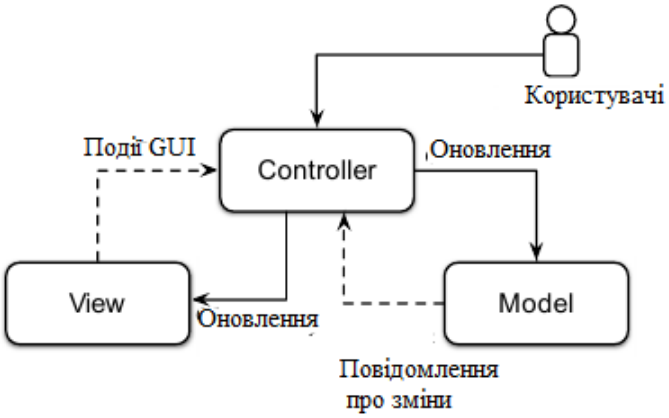

Рисунок 2.3 – Cocoa MVC модель [15]

В даному випадку контролер (controller) є посередником між видом і моделлю, так що вони не взаємодіють один з одним.

Особливістю даного шаблону проектування MVC є те, що контролер (controller) настільки залучений в життєвий цикл виду (View) за допомогою підкласу UIViewController, тому важко сказати, що вони повністю відокремлені.

Основні особливості даної архітектури:

– розподіл – вид (view) і модель (model) в даному випадку повністю розділені, але вигляд (view) і контролер (controller) тісно зв’язані між собою;

– тестованість – через такий розподіл можна тестувати тільки модель (model) окремо від виду (view);

– простота використання – кількість коду значно зменшується.

Cocoa MVC є найкращим шаблоном проектування з точки зору простоти і швидкої реалізації.

Таким чином, розробка мобільних додатків під iOS має як свої переваги, так і недоліки.

2.3. Кросплатформені мобільні додатки

Пристрої під управлінням ОС Android більш поширені на території України, але ринок iOS також постійно нарощує свою потужність. Тому для розробників найкращим вибором є розробка кросплатформених додатків.

Підтримка різних платформ досягається шляхом крос-компіляторів і уніфікації інтерфейсів. Крос-компілятори здатні виробляти на одній і тій же платформі код, який буде легко переноситися на інші платформи. Попередня збірка такого крос-компілятора вимагає певних витрат часу і ресурсів, а робота з ним не гарантує абсолютно успішного результату, так як існують різні специфікації цільових платформ.

На сьогоднішній день існує безліч засобів розробки програмного забезпечення, завдяки яким можна створювати програмний продукт, що переноситься з платформи на платформу без будь-яких серйозних зусиль і без додаткової зміни вихідного коду [21; 22].

Розглянемо фреймворки для розробки кросплатформених додатків.

Одним з перших кросплатформених фреймворків став Cordova. Його запуск відбувся у 2009 році. За офіційною архітектурною схемою основою рендеринга виступає WebView – тобто звичайний браузер. Не дивлячись на те, що це фреймворк, який дуже довго існує, він надає лише незначний набір з плагінів і зв’язки API для забезпечення їх взаємодії (рис. 2.4). Якогось специфічного набору інструментів у нього немає. Наявний скромний перелік

сервісів і бібліотек. Середовище розробки відсутнє. Відлагодження відбувається в консолі браузера. Перевагою Cordova є зручний механізм шаблонів, на основі яких можна генерувати проект [30].

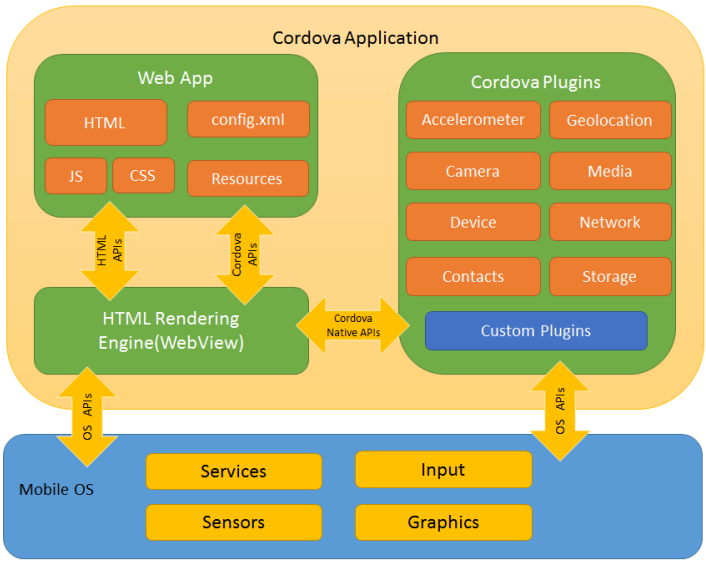

Рисунок 2.4 – Архітектура кросплатформеного додатка на основі фреймворка Cordova [30]

Фреймворк Xamarin у Visual Studio. Visual Studio – це IDE від Microsoft, що підтримує ряд мов, включаючи C #, VB.net, JavaScript і багато інших. За допомогою фреймворка Xamarin, можна створювати кросплатформені додатки на мові C#, а потім тестувати їх на кількох пристроях, підключених до хмари.

В порівнянні з JS-фреймворками код на C# компілюється, що дозволяє йому виконуватись швидше в порівнянні з додатками на WebView. Для нього наявні два варіанти архітектури (рис. 2.5). UI повинен розроблятися окремо для обох платформ, а бізнес-логіка загальна. Цей варіант буде цікавий, якщо виникає бажання зробити максимально нативний додаток з точки зору UI, використовуючи одну і ту ж бізнес-логіку. Виходить, що Xamarin вимагає наявності в команді спеціалістів, здатних створити UI для обох платформ окремо. Це хороший і безкоштовний вибір для додатків і для Android, і для iOS одночасно. Також він є відмінним вибором для тих, хто вже знайомий з C# і / або Visual Studio.

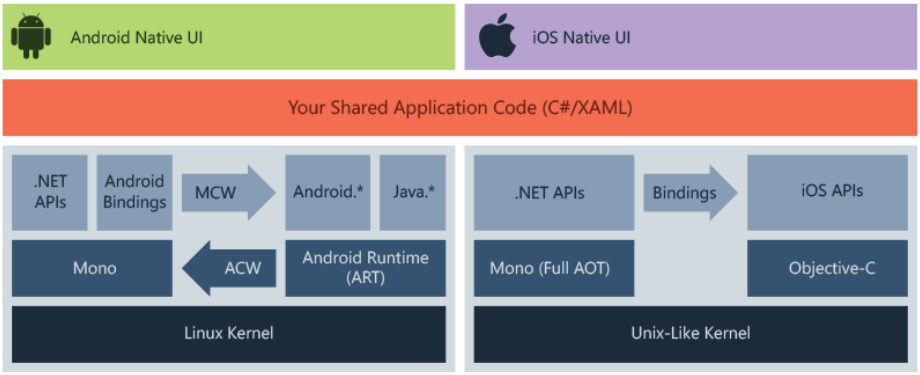

Рисунок 2.5 – Архітектура додатка на основі фреймворка Xamarin [30]

Фреймворк NativeScript випущений в 2014 році. За допомогою нього можна створювати додатки на Angular або Vue, які являються фреймворками для мови JavaScript. За довгі роки розвитку NativeScript накопичив чималий набір інструментів розробника: готовий арсенал UI-компонентів, власну пісочницю, магазин компонентів тощо. Для відлагоджування використовується або консоль браузера, або плагін для VS Code, що надається розробниками. Даний фреймворк має цікаву архітектуру – API платформи через рефлексію перекидають на сторону JavaScript (рис.2.6).

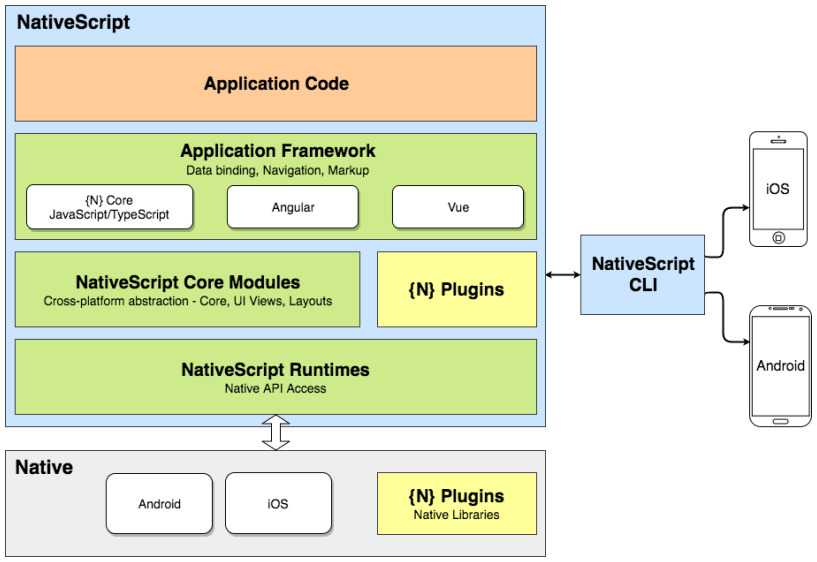

Рисунок 2.6 – Архітектура фреймворка NativeScript [30]

У повсякденній роботі використовується набір вбудованих layout-компонентів і CSS-подібний синтаксис.

Тепер розглянемо React Native – JS фреймворк. Перший його реліз відбувся в 2015 році [29]. Загальна архітектура дуже схожа на NativeScript

(рис.2.7). У проміжку між JS і платформою знаходиться Yoga – багатоплатформовий рушій, який реалізує технології розташування блоків на HTML сторінці (flex-box layout).

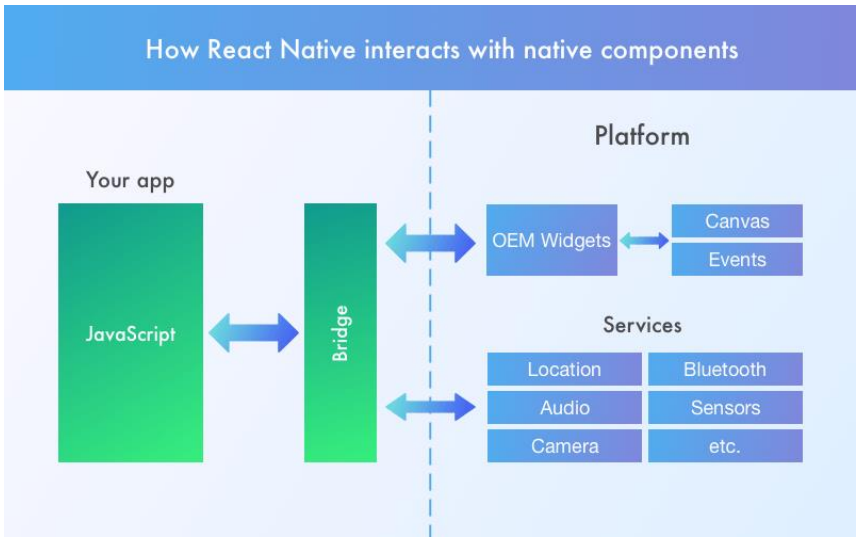

Рисунок 2.7 – Архітектура React Native [30]

React Native на даний момент активно розвивається, релізи з'являються кожні 3-6 місяців. Його особливістю є те, що іноді зустрічаються відмінності при відображенні в iOS та Android. Тому перед завершенням розробки додатка варто протестувати його роботу на обох платформах. Віджети, що надаються платформою, іноді сильно різняться [29].

Наступним на черзі йде один з найпопулярніших кросплатформених фреймворків – Flutter. Перша його демонстрація відбулась наприкінці 2018 року. Платформа досить молода, але вона активно розвивається: щоквартальні релізи, активний розвиток мови Dart, багатий інструментарій розробника, онлайн-редактор для анімації. Особливістю архітектури Flutter є те, що він сам малює кожен піксель, контролює жести і анімацію. Flutter не використовує OEM-віджети, замість цього команда Flutter створила два набори віджетів для основних мобільних платформ: Material для Android і Cupertino для iOS. Таким чином, вони заново створили всі UI-компоненти з обох мобільних платформ. Безпосередньо з мобільною платформою (геолокація, звук, Bluetooth) взаємодія відбувається через Platform Channels [29, 31].

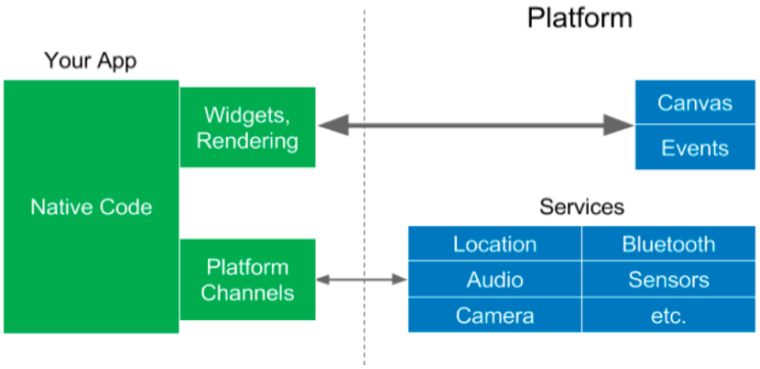

Рисунок 2.8 – Архітектура фреймворка Flutter [30]

Насамкінець, розглянемо Qt Framework. Він надає широкий спектр можливостей, достатніх для створення додатків практично будь-якої спрямованості на різних платформах. Даний фреймворк розроблено двома компаніями: Qt Company і Qt Project. Розповсюджується по комерційній та відкритій (з деякими обмеженнями) ліцензіями. Основною мовою програмування є C ++. Також є прив'язки до інших мов [32].

Qt дозволяє запускати програмне забезпечення, написане з його допомогою в більшості сучасних операційних систем, просто компілюючи програму для кожної системи без зміни вихідного коду. Він повністю об'єктно-орієнтований, розширюється і підтримує компонентне програмування. Даний фреймворк пропонує розробникам модульну бібліотеку, яка налічує понад 700 C++ класів, технологію QtQuick для створення UI за допомогою декларативної мови QML і інструментарій професійного рівня.

QML (Qt Meta Language або Qt Modeling Language) – декларативна мова програмування, яка використовується для розробки додатків з складним інтерфейсом. QML, з вбудованим в нього рушієм JavaScript, є засобом розробки зовнішнього представлення – інтерфейсу (front-end), а Qt C++ використовується для внутрішньої реалізації (back-end). У мовній інфраструктурі QML немає конструкцій програмування – циклів, умовних операторів тощо. Для їх реалізації як раз і призначений вбудований рушій JavaScript [32]. Структура QML-коду схожа на HTML і в першому наближенні нагадує зв'язку технологій HTML, CSS і JavaScript. Але, на відміну від мови розмітки HAML в C# і XML в Java, QML є описовою мовою програмування, в якій задається не тільки

розмітка і зовнішній вигляд елементів, а й описується їх поведінка, а також реакція на події. Це зроблено для більш зручної взаємодії між дизайнерами і програмістами, яким пропонується єдиний і простий інструмент для спільної розробки діючого прототипу.

Qt дозволяє прискорити процес створення ПЗ, підвищити його ефективність, тим самим значно скорочуючи час виходу готового продукту на ринок.

Фреймворк дозволяє створювати програми для наступних платформ:

– Мобільні: Android, iOS, WindowsPhone / WindowsRTi;
– Desktop: Windows, Mac OS, Linux;
– Вбудовані: Embedded Linux, Embedded Android, Windows Embedded, QNX, VxWorks, INTEGRITY.

Ключовою перевагою Qt є його кросплатформеність, тобто можливість застосування одного коду для різних платформ, операційних систем і пристроїв

Крім того можна виділити ще такі переваги:

– використання C ++ і QML;
– нативність створеного ПЗ з точки зору відповідної платформи;
– висока продуктивність створеної програми;
– середовище розробки QtCreator з можливістю установки і налагодження безпосередньо на пристрої та додатковими інструментами;
– детальна документація, приклади використання, якісна технічна підтримка.

Однак одним з основних недоліків робочого середовища Qt є великий розмір готового програмного продукту, що в ряді випадків є недопустимим [32].

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ «UKRAINE GUIDE»

3.1. Формування специфікації для додатку

Ціль та призначення мобільного додатку – надати користувачеві довідкову інформацію щодо українських міст та селищ. Такі дані нададуть йому можливість організувати свою подорож у відповідності з уподобаннями.

Всі вимоги до додатків поділяють на функціональні і нефункціональні. Функціональні вимоги визначають ту функціональність системи, яку розробники повинні створити, щоб користувачі змогли задовольнити свої потреби та вирішити певні проблеми.

Програма повинна мати такі функціональні можливості:

- Пошук населеного пункту по імені;
- Отримання довідкової інформації щодо обраного населеного пункту;
- Перегляд карти місцевості населеного пункту за наявності підключення до мережі інтернет;
- Перегляд списку найпопулярніших закладів в населеному пункті та можливість переходу на веб-сторінку даного за наявності підключення до мережі інтернет;
- Створення списку улюблених місць для подорожей;
- Створення та редагування власних маршрутів;
- Бронювання житла та замовлення залізничних квитків за допомогою відповідних сервісів;
- Перегляд основної інформації без підключення до мережі інтернет.

Нефункціональні вимоги являють собою опис характеристик додатку, важливих для користувача при роботі з системою. Сформуємо загальні нефункціональні вимоги до додатку.

Додаток повинен бути кросплатформеним, реалізованим під ОС Android та iOS.

Вимоги до надійності.

Додаток повинен передбачати базовий захист від несанкціонованого доступу. Система повинна забезпечувати валідність даних, що вводяться користувачем.

Вимоги до зручності використання.

Інтерфейс додатку повинен бути зручним та зрозумілим, відповідати базовим принципам побудови UI (User Interface), що висуваються до мобільних програм. Додаток повинен підтримувати роботу в портретній та ландшафтній орієнтації екрану.

Зручність навігації.

Одним з критеріїв розробки додатку повинен стати принцип простоти навігації. Користувач повинен інтуїтивно і безперешкодно переміщатися для отримання необхідної інформації та послуг.

Початкова версія додатку передбачає українську локалізацію користувацького інтерфейсу.

Перейдемо до проектування архітектури та роботи додатку. Для цього скористаємося об’єктно-орієнтованим підходом, використовуючи нотацію UML. Концептуальна модель розроблювальної системи виражається у вигляді «діаграм прецедентів» (Use Case diagram). Прецедент (англ. – use case) є документом, що описує послідовність подій, пов’язаних з виконавцем (якого називають актором), який для досягнення певного результату використовує створювану систему. За допомогою прецеденту описується взаємодія з системою [33; 34; 35].

Створення діаграми варіантів використання будемо здійснювати на основі аналізу функціональних вимог. З додатком взаємодіє один актор – користувач, на мобільний пристрій якого встановлений додаток.

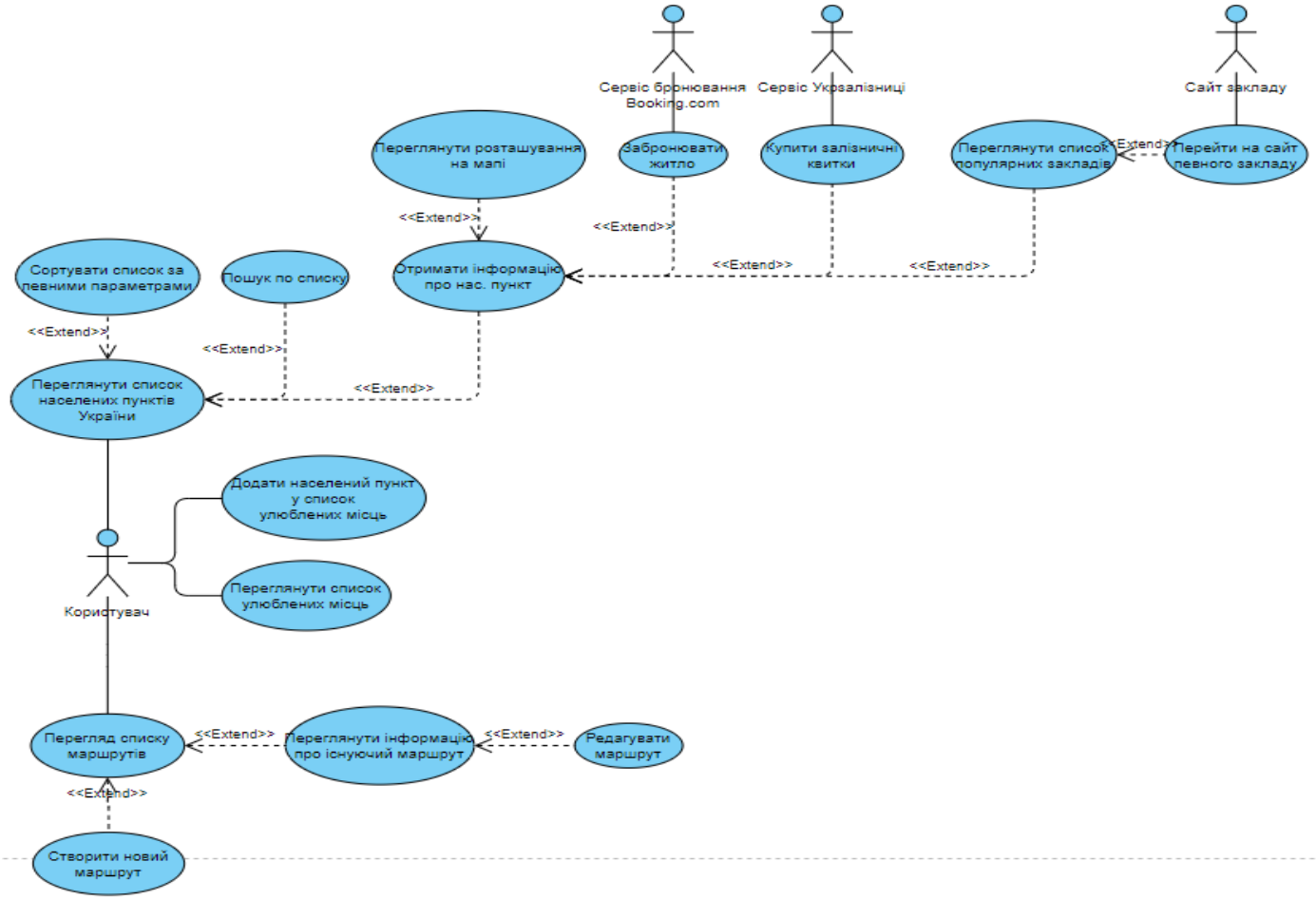

Рисунок 3.1 – Діаграма прецедентів мобільного додатку «Ukraine Guide»

Користувач може переглядати список населених пунктів (НП) України. Кожен з них він може відкрити на карті, яка є вбудованою у додаток і буде доступна для перегляду на окремій сторінці. Також доступний функціонал перегляду найкращих закладів, які наявні в певному населеному пункті та відвідування їх офіційної веб-сторінки. Крім того користувач може скористатись функціоналом бронювання житла та купівлі квитків на потяг з місцем призначення у даному НП. В цьому йому допоможуть сторонні спеціалізовані сервіси.

Користувач здатен додавати якийсь населений пункт до списку улюблених НП, які він вже відвідав чи тільки бажає відвідати.

Також він має можливість створювати власний маршрут, додаючи до нього різні точки зупинки. Створений маршрут можна редагувати та видаляти.

Діаграма діяльності зі створення та редагування маршруту наведена на

рис. 3.2.

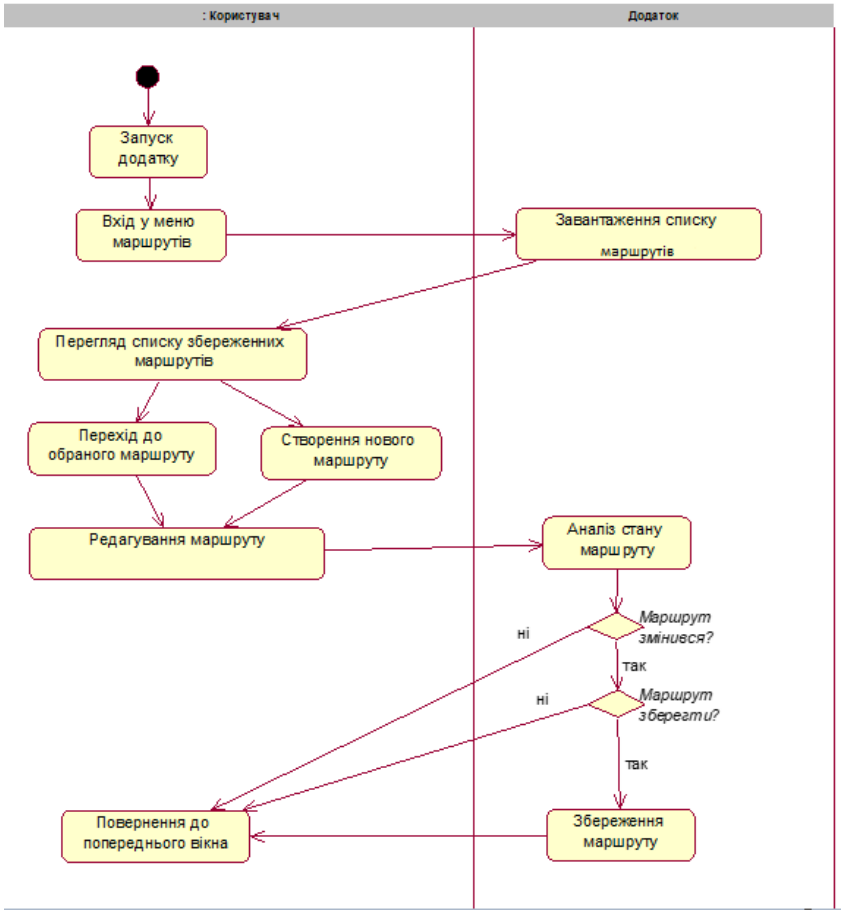

Рисунок 3.2 – Діаграма діяльності роботи з маршрутами

Користувач переглядає список існуючих маршрутів, які є в системі. Він має можливість або створити новий маршрут або відредагувати вже існуючий. Редагування полягає в додаванні нових зупинок до маршруту, їх видаленні, зміні порядку та перейменуванні маршруту. Додаток аналізує стан маршруту та визначає необхідність його збереження. За результатами аналізу стану маршруту та реакції користувача, додаток зберігає даний маршрут або повертається до попереднього вікна без змін.

Діяльність користувача для перегляду списку населених пунктів та інформації про конкретний з них (рис. 3.3) складається з таких етапів:

– Перехід на сторінку списку населених пунктів України;

– Сортування та фільтрування списку за бажаними доступними критеріями;

– Вибір конкретного населеного пункту та перехід на сторінку

відображення інформації про нього;

– Перегляд населеного пункту на карті за наявності підключення до мережі інтернет;

– Перегляд списку популярних закладів (культури, готельно-ресторанного бізнесу, мистецтва, спорту тощо) та можливість переходу на веб-сторінку якогось з них;

– Перегляд детальної інформації про населений пункт за наявності підключення до мережі інтернет;

– Бронювання житла в певному населеному пункті за наявності підключення до мережі інтернет;

– Купівля залізничних квитків до населеного пункту за наявності підключення до мережі інтернет.

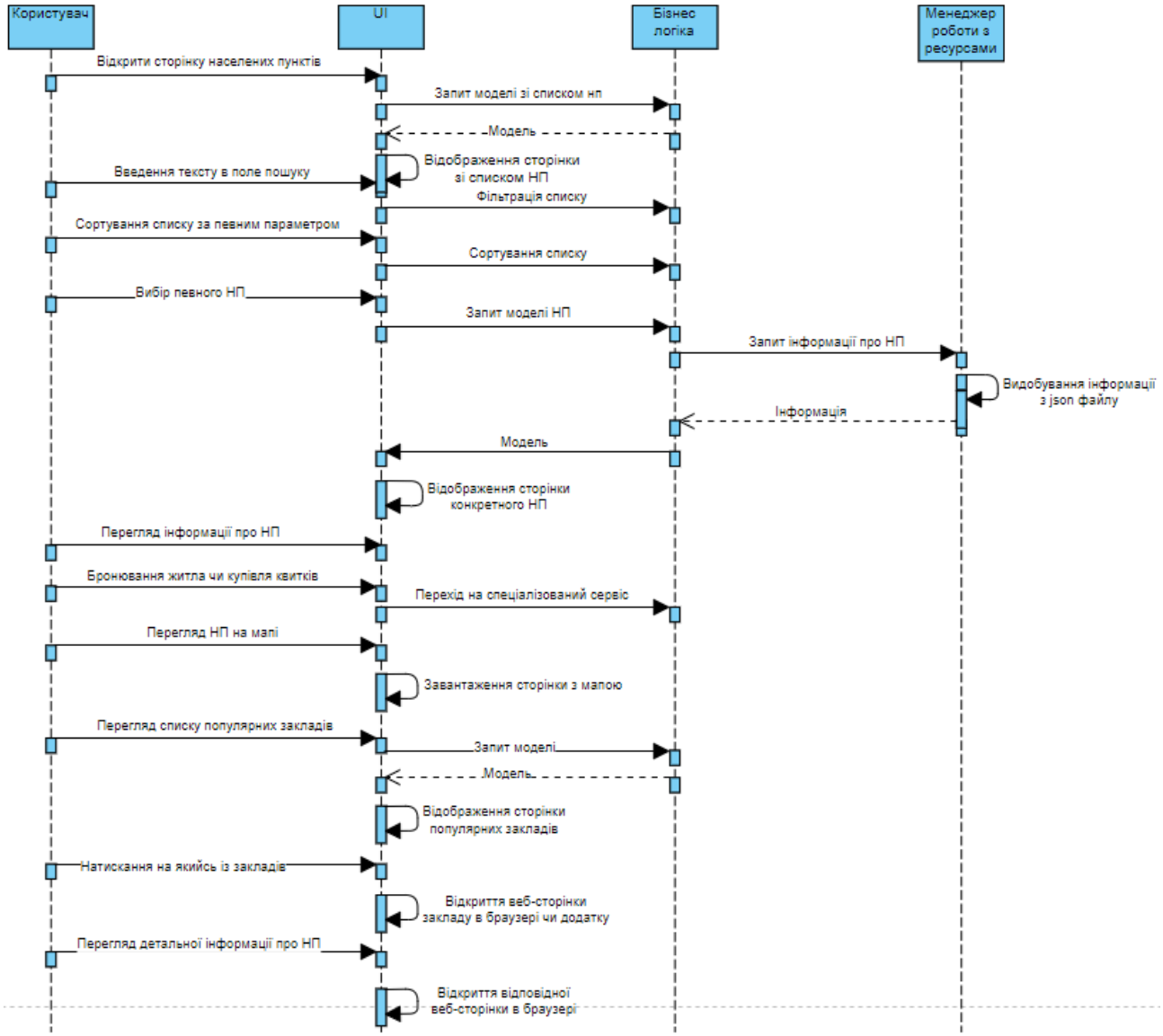

Рисунок 3.3 – Діаграма послідовностей для перегляду списку населених пунктів та інформації про конкретний з них

Процес роботи зі списком улюблених населених пунктів (рис. 3.4) складається з таких етапів:

– Перехід на сторінку списку населених пунктів України;

– Вибір конкретного НП та перехід на сторінку відображення інформації про нього;

– Додавання або видалення НП до / зі списку улюблених;

– Повернення до головного меню;

– Перехід на сторінку улюблених населених пунктів;

– Вибір конкретного населеного пункту та перехід на сторінку

відображення інформації про нього;

– Видалення населеного пункту зі списку улюблених.

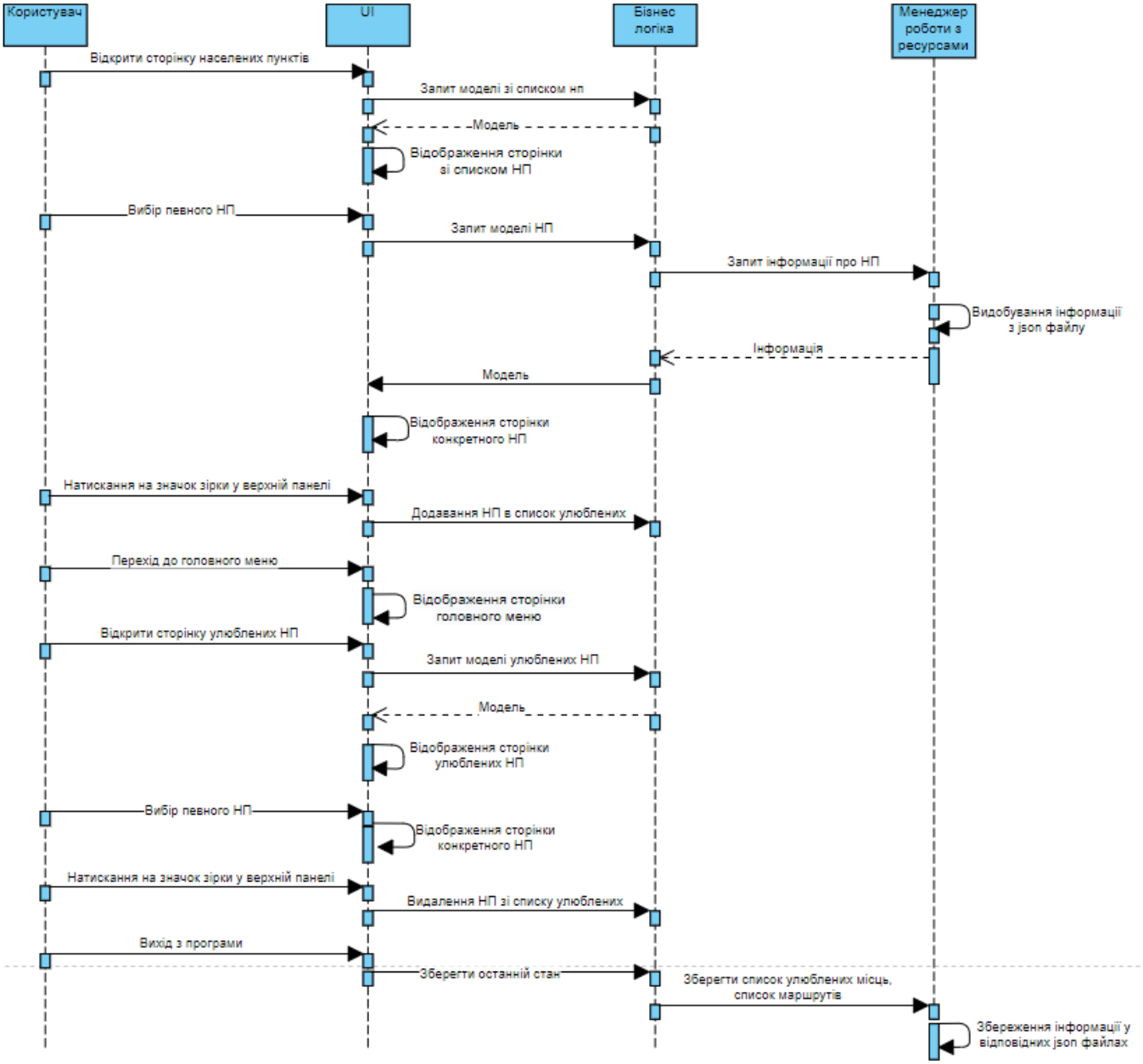

Рисунок 3.4 – Діаграма послідовності до процесу перегляду та роботи зі списком улюблених населених пункті

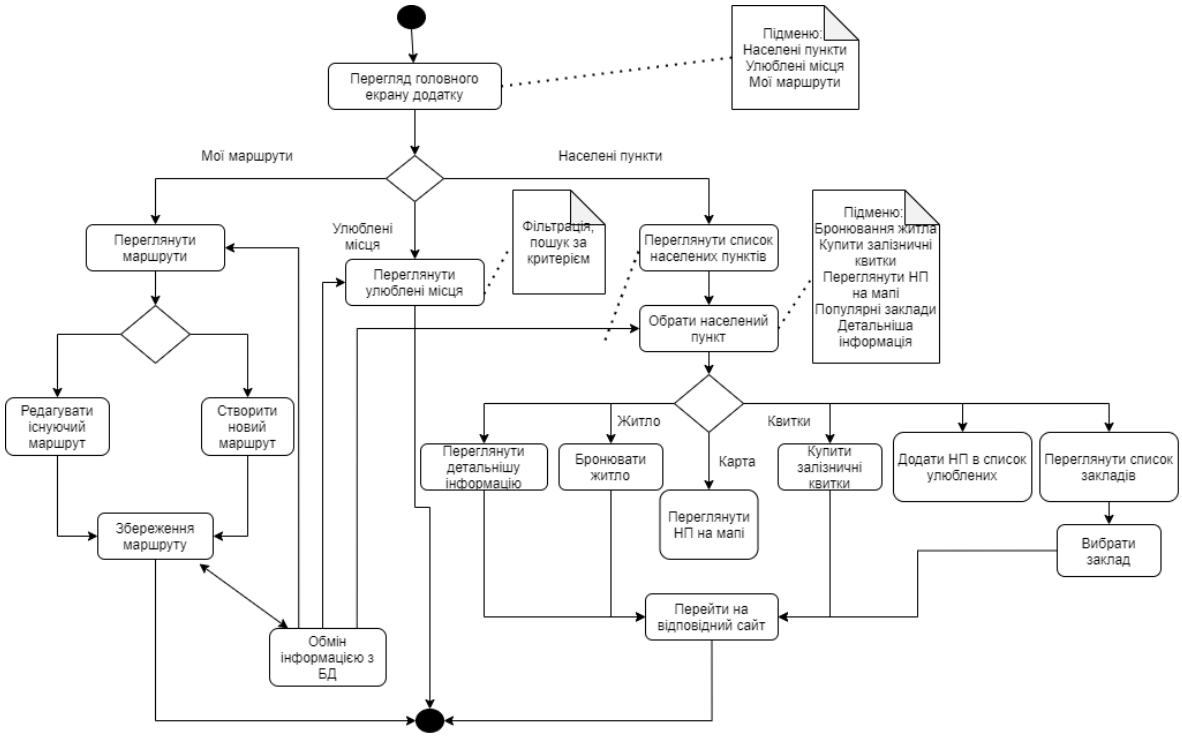

Рисунок 3.5 – Загальна схема роботи користувача з мобільним додатком «Ukraine Guide»

На основі розробленої концептуальної моделі додатку визначимо взаємодію між його компонентами (рис. 3.6).

Компонент «Інтерфейс користувача» призначений для взаємодії користувача з додатком.

«Компонент роботи з населеними пунктами» надає інформацію про населені пункти України і відповідає за пошук населених пунктів по назві та їх сортування за наступними критеріями: алфавіт, площа, населення. Також за надання інформації про НП, бронювання житла, замовлення залізничних квитків, надання інформації про популярні заклади та відображення НП на мапі.

«Компонент роботи з маршрутами» відповідає за створення та редагування маршрутів, при цьому звертається до компонента населених пунктів для отримання списку міст та селищ.

«Компонент роботи з улюбленими НП» надає інформацію про улюблені місця користувача та дає змогу редагувати даний список. Для роботи йому потрібно використовувати компонент населених пунктів.

«Менеджер ресурсів» видобуває необхідну інформацію з бази даних і вносить її туди ж [35, 36].

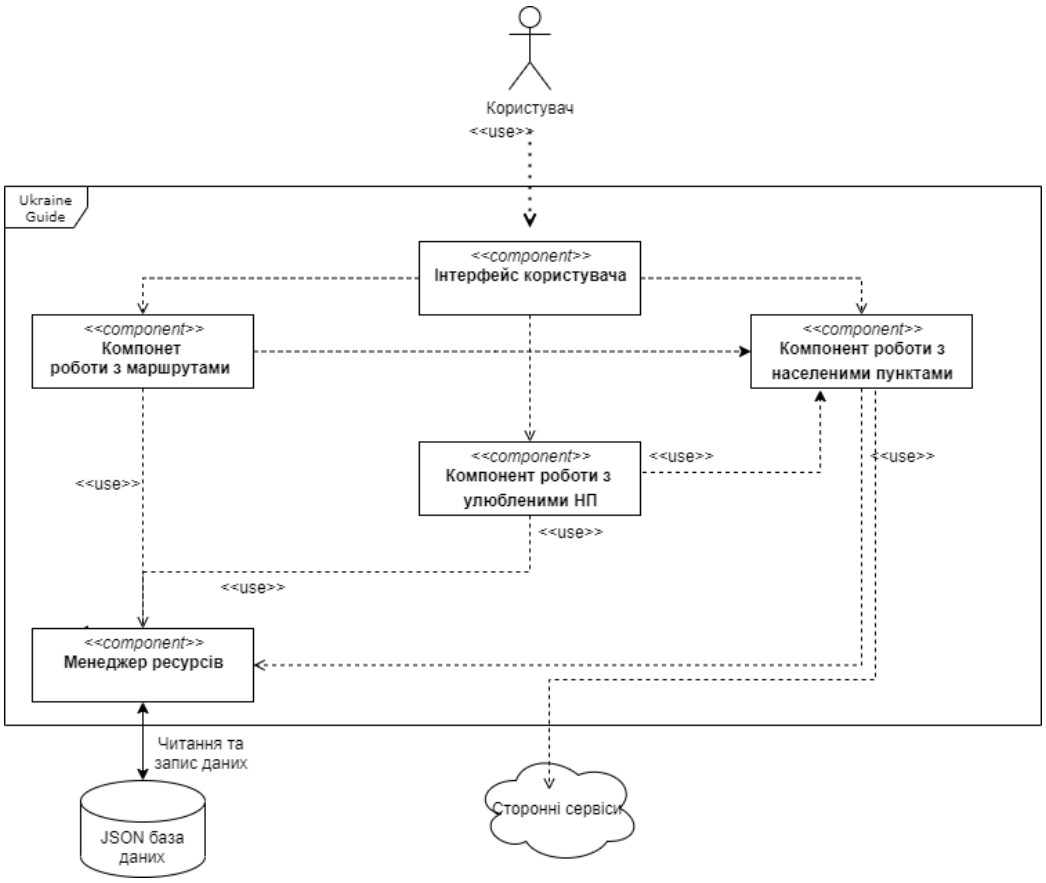

Рисунок 3.6 – Взаємодія компонентів мобільного додатку «Ukraine Guide»

Для забезпечення функціональних вимог мобільний додаток повинен використовувати базу даних. Постає вибір: використовувати у якості БД реляційну базу даних або звичайний текстовий файл. Проаналізувавши інформацію, яка повинна зберігатись в сховищі, дійшли висновку, що використання реляційної бази даних буде надлишковим, оскільки кількість даних є невеликою і наявна тільки одна сутність – населений пункт. Тож, було обрано JSON, як текстовий формат для збереження даних. JSON є абревіатурою, яка розшифровується як JavaScript Object Notation. Він являє собою певний текстовий формат обміну даними, де інформація розташовується компактно, завдяки чому її легко прочитати, навіть якщо цей текст читатиме людина. JSON зберігає дані об’єкта у вигляді множини «name»:«value», де «name» – це рядок, а «value» може набувати різних форм, таких як число, рядок, масив, інший об’єкт, логічний літерал. Сам JSON об’єкт починається з символа

«{» та закінчується символом «}». Не дивлячись на те, що цей формат утворився від мови програмування JavaScript, він є незалежним і може використовуватись практично будь-якою мовою програмування [37, 38]. На рис. 3.7 зображена структура json файлу, в якому зберігається інформація про населені пункти.

```
{
  "унікальний_ідентифікатор_населеного_пункту": {
    "engName": "",
    "ukrName": "",
    "population": 2000,
    "area": 0.0,
    "oblast": "",
    "region": "",
    "type": "",
    "foundationDate": "",
    "coordinates": [],
    "images": [ "1.jpg" ],
    "moreInfoUrl": "",
    "topEspablishments": [
        {
            "ukrName": "",
            "engName": "",
            "type": "",
            "moreInfoUrl": ""
        }
    ]
  },
  ...
}
```

Рисунок 3.7 – Структура файлу, в якому зберігається інформація про населені пункти

З рис. 3.7 ми можемо побачити, що кожен населений пункт повинен бути представлений унікальним ідентифікатором. Це необхідно для коректного видобування інформації з файлу. Розглянемо поля в об’єкті населеного пункту.

а) «engName» та «ukrName» – англійська та українська назва НП;

б) «population» – населення НП;

в) «area» – площа НП;

г) «oblast» – область;

д) «region» – район;

е) «type» – тип НП (село, село міського типу або місто);

ж) «foundationDate» – рік заснування НП;

з) «coordinates» – десяткові географічні координати НП;

и) «images» – назви зображень, які відображаються на сторінці НП;

к) «moreInfoUrl» – посилання на веб-сторінку з детальною інформацією про НП;

л) «topEstablishments» – об’єкт, який представляє список популярних закладів в НП. Він представлений наступними полями:

1) «engName» та «ukrName» – англійська та українська назва закладу;
2) «type» – тип закладу (кафе, ресторан, піцерія і тд.);
3) «moreInfoUrl» – посилання на веб-сторінку закладу.

Даний файл надає всю інформацію про населені пункти.

Список улюблених місць також зберігається у файлі з форматом json і має наступну структуру:

```
{
  “favoriteLocalities”: [“унікальний_ідентифікатор_населеного_пункту”, … ]
}
```

Дані з цього файлу дістаються на початку роботи програми й оновлюються при її завершені.

І, насамкінець, перейдемо до маршрутів, які зберігаються у json файлі з аналогічною структурою до списку улюблених місць, тільки замість унікальних ідентифікаторів населених пунктів у даному у ньому зберігаються назви файлів з конкретними маршрутами. А вже сам файл з певним маршрутом повністю ідентичний за структурою файлу, в якому зберігаються улюблені населені пункти. Файли з маршрутами можуть модифікуватися як під час роботи програм – так і після її завершення.

3.2. Розробка користувацького інтерфейсу

Одним з основних етапів розробки будь-якого мобільного додатку є проектування користувацького інтерфейсу.

І Google Play Market, і Apple App Store висувають досить суворі рекомендації щодо інтерфейсу додатку для запобігання появі погано розроблених програм у їх магазинах додатків. Основні вимоги до дизайну

інтерфейсу такі:

1. Забезпечення інтуїтивного розуміння роботи додатку. Як тільки користувач завантажує додаток, він повинен інтуїтивно розуміти, куди йти і що робити для досягнення бажаних результатів. Це можна здійснити, створивши вікно з описом функціональних можливостей програми, а також, використовуючи відповідні підказки в потрібному місці додатку.

2. Забезпечення користувачу легкого доступу до необхідної інформації. Для цього потрібно дотримуватись простої та швидкої навігації.

3. Мінімалістичний, очищений від зайвих графічних навантажень, дизайн. Зараз це основна філософія дизайну. Одним зі способів досягнення цієї вимоги є використання незайнятого простору навколо основних елементів сторінки, зведення кількості елементів на сторінці до мінімуму.

4. Орієнтованість на користувача вимагає застосування правила «великого пальця» , тобто у додатку всі елементи та кнопки, на які можна натиснути, повинні бути в межах досяжності великого пальця користувача [39, 40, 41].

Отже, при розробці інтерфейсу мобільного додатку потрібно враховувати наведені вище рекомендації. Це забезпечить достатню привабливість продукту серед користувачів. Тому кількість робочих екранів повинна бути мінімально необхідною і відповідати кількості окремих режимів роботи системи, кнопки, що дають виконати певні дії, повинні мати простий та зрозумілий дизайн.

На рис. 3.8 наведений список вікон розроблюваного додатку та порядок навігації між ними.

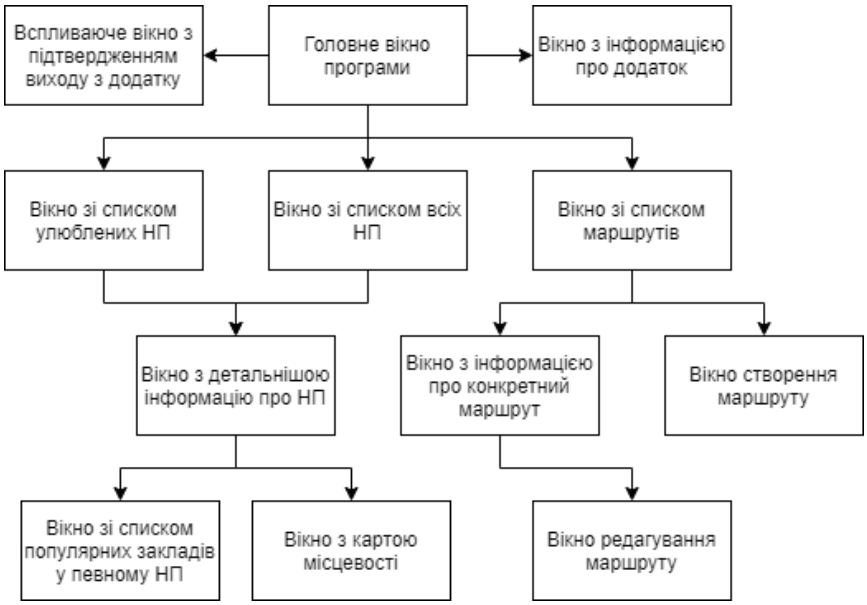

Рисунок 3.8 – Навігація між екранними формами в додатку «Ukraine Guide»

Ескізи екранних форм мобільного додатку наведені на рис. 3.9-3.12.

Всі вікна, які використовуються для роботи з додатком, мають однакову структуру та розробляються в однаковій кольоровій схемі. Зазвичай з кожним кольором у людини є певні асоціації, що пояснюються фізіологічним сприйняттям кольору. Для додатку була використана коричнева гама, оскільки вона сприяє зняттю стресу, додає впевненості, стабільності, що гармонує з областю застосування додатку [39, 40, 41].

Більшість користувачів, що використовують додаток, звикли переглядати екран зліва направо і зверху вниз [42, 43]. З огляду на цей фактор, при розробці дизайн додатку варто елементи управління розташовувати так, щоб користувач зміг їх знайти, тобто у верхній частині вікна, але спочатку повинно йти текстове поле, а потім елементи управління для нього.

Перейдемо до створення ескізів вікон, для того, щоб в подальшому приблизно розуміти як буде виглядати UI та для спрощення його реалізації.

На головній екранній формі передбачена наявність кнопки виходу з програми (правий верхній кут вікна) та кнопки отримання інформації про програму (лівий верхній кут вікна). На рисунку 3.9 зображений ескіз головного вікна програми, яке містить варіанти переходу у відповідні режими роботи програми.

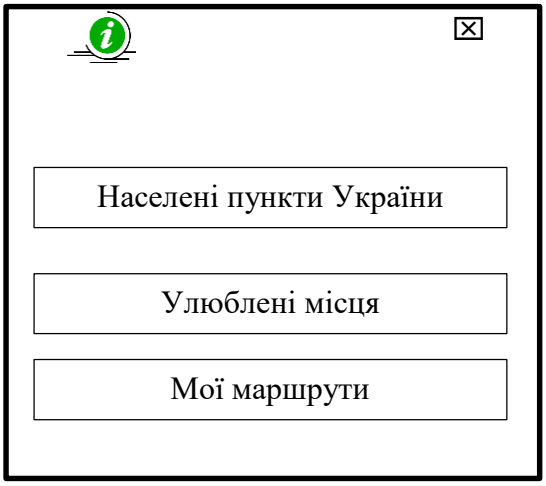

Рисунок 3.9 – Ескіз головного вікна додатку

Ескіз вікна режиму перегляду списку населених пунктів зображено на рис. 3.10.

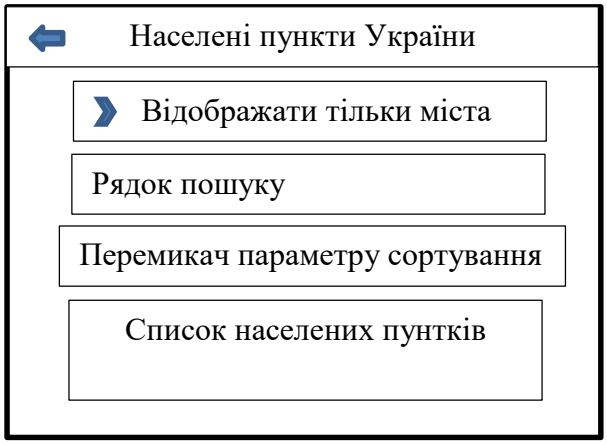

Рисунок 3.10 – Ескіз вікна додатку в режимі Населені пункти

Ескіз вікна перегляду інформації про населений пункт наведений на рис. 3.11.

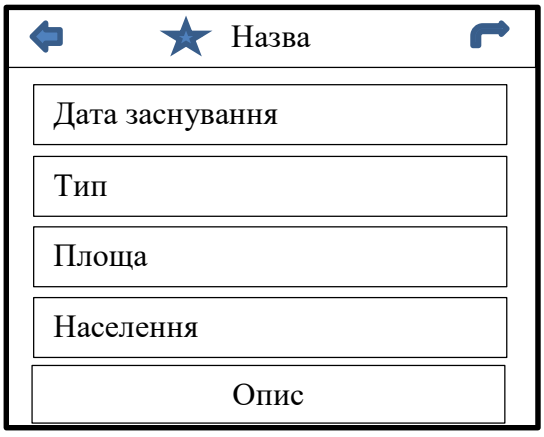

Рисунок 3.11 – Ескіз вікна додатку при перегляді інформації про обраний населений пункт

Ескіз меню, що доступне у вікні перегляду населеного пункту після натискання на кнопку у правому верхньому куті наведене на рис. 3.12. Це меню буде зображене зліва поверх вікна з інформацією про НП і з’являтиметься з певною анімацією.

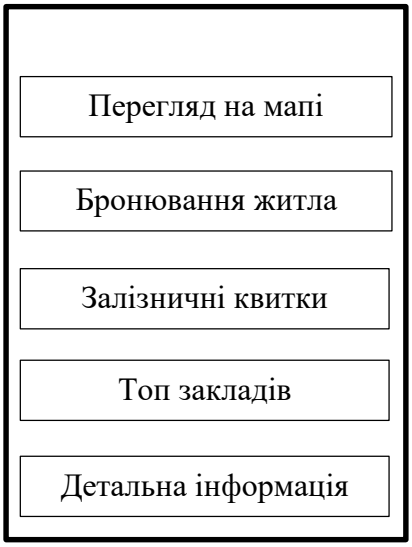

Рисунок 3.12 – Ескіз бокової панелі, що надає додатковий функціонал для роботи із населеними пунктами

Ескіз вікна улюблених населених пунктів буде аналогічним до вікна із відображенням всіх НП України. Єдиною відмінністю даних вікон буде їхній вміст.

Ескізи вікон по роботі з маршрутами наведені на рис. 3.13-3.14.

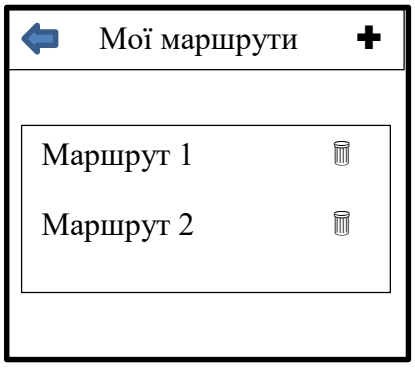

Рисунок 3.13 – Ескіз вікна переліку маршрутів

На рис. 3.14 зображено ескіз вікна перегляду конкретного маршруту (зліва) та режим редагування, який вмикається та вимикається при натисканні на кнопку у правому верхньому куті вікна (ескіз справа)

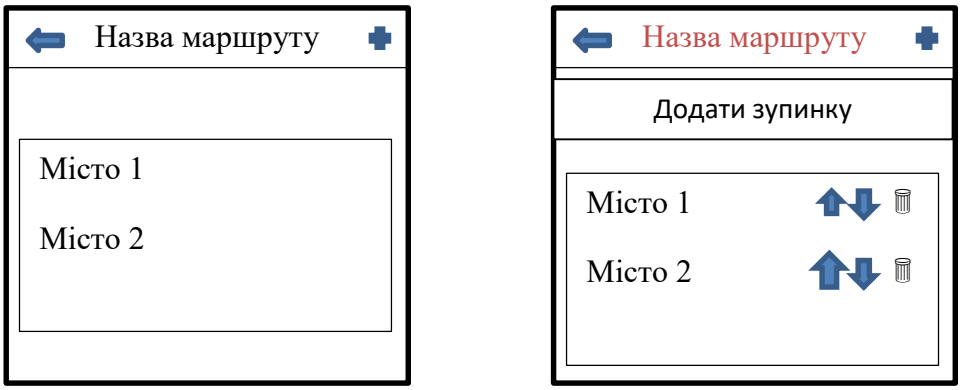

Рисунок 3.14 – Ескіз вікна перегляду конкретного маршруту (зліва) та режим редагування цього ж маршруту (справа)

3.3. Вибір технологій для розроблюваного додатку, їхня характеристика

На основі проведеного проектування мобільного додатку перейдемо до вибору та обґрунтування технологій розробки. Перш за все визначимось з мовою програмування. Оскільки передбачається розробка кросплатформеного додатку, який повинен працювати під управлінням операційних систем Android та iOS, то вибір мови C++ є досить обґрунтованим.

Основними перевагами мови C++ стали:

– Кросплатформеність. Програми, написані на мові C++, можуть бути розроблені під більшість сучасних операційних систем;
– Під дану мову розроблені відкриті бібліотеки і модулі, що також дає змогу суттєво підвищити ефективність розробки додатків та простоту їх розробки;
– Код, написаний на мові C++, має високі швидкісні характеристики, оскільки в цих програмах управління ресурсами здійснюється розробником напряму [44, 45];
– В мене наявний досвід програмування на даній мові і я б хотів покращити свої знання під час виконання даної роботи.

Допоміжним інструментарієм для створення додатку був обраний Qt – один з найпопулярніших кросплатформених фреймворків для розробки ПЗ на мові програмування C++. Він дає змогу запускати написане ПЗ в більшості сучасних операційних систем шляхом простої компіляції програми для кожної ОС без зміни вихідного коду.

Середовище включає в себе всі основні класи, які можуть знадобитися при розробці прикладного програмного забезпечення, починаючи від елементів графічного інтерфейсу і закінчуючи класами для роботи з мережею, базами даних і XML. Фреймворк надає дуже зручний та потужний функціонал для роботи з json файлами, що дуже стає в нагоді, оскільки вся інформація про населені пункти буде зберігатись саме в таких файлах і простота роботи з ними буде корисною для розробника. Qt є повністю об’єктно-орієнтованим [32]. Перевагою Qt є те, що з версії 4.5 він поширюється в 2 редакціях:

– Qt Commercial – для розробки комерційних додатків;

– Qt Open Source – поширюється під ліцензією GNU GPL і GNU LGPL, що є важливим для самостійних розробників. Якраз цю версію буде використана, оскільки на даний момент додаток буде не комерційним, а навчальним продуктом.

Qt надає 2 способи створення графічного інтерфейсу: Qt Widgets та QML (Qt Modeling Language). Перший з них надає велике різноманіття C++ класів для побудови різних UI елементів. Проблема цього підходу полягає у його складності, оскільки всі елементи графічного інтерфейсу потрібно вручну створювати в C++ коді, що є дуже незручно. Також цей підхід вважають застарілим і використовують тільки у застарілих проектах. На щастя, в 2009 році на заміну Qt Widgets прийшов QML – декларативна мова програмування, заснована на JavaScript, яка частиною Qt Quick, середовища розробки UI, поширюваного разом із Qt. QML код являє собою дерево елементів. QML компонент, так само, як і компонент Qt, являє собою сукупність блоків: графічних (таких, як rectangle, image) і поведінкових (таких, як state, transition, animation). Ці елементи можуть бути об'єднані, щоб побудувати комплексні

компоненти, починаючи від простих кнопок і повзунків та закінчуючи повноцінними складними застосунками. QML компоненти можуть бути доповнені стандартними JavaScript-вставками шляхом вбудовування блоків коду на цій мові. Також вони можуть бути розширені C++ компонентами через Qt framework.

Враховуючи великі переваги QML було вирішено обрати саме цей підхід до створення графічного інтерфейсу користувача, оскільки він сучасніший і гнучкіший.

3.4. Наповнення інформацією та тестування готового додатку

В результаті проведеного проектування був розроблений функціонал додатку «Ukraine Guide».

Для початку додамо підтримку двадцяти чотирьох обласних обласних центрів України та заповнимо інформацію про них. Основним джерелом інформації про населення в містах слугував ресурс [46], оскільки є офіційним джерелом, дані якого повинні бути найбільш точні. Всі інші дані збирались з різноманітних відкритих джерел в мережі інтернет, в тому числі з офіційних веб-сторінок міст. Зображення для елементів додатку були взяті з сайту [47], який надає близько 120 тисяч різноманітних ілюстрацій. Даний ресурс був вибраний, тому що він являється безплатним і надає широкий асортимент картинок для можливості задоволення потреб розробки UI.

Проведемо ручне тестування даного додатку.

При запуску програми на екрані мобільного пристрою з’являється головне меню (рис. 3.15).

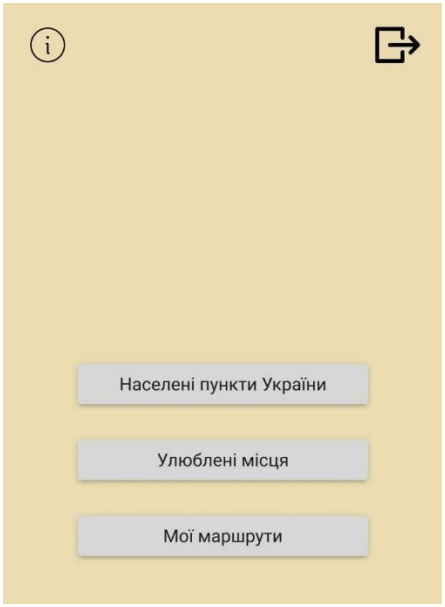

Рисунок 3.15 – Головне меню додатку «Ukraine Guide»

Розглянемо роботу з населеними пунктами. Після вибору даного пункту меню на екрані з’являється список населених пунктів України (рис. 3.16). На початковому етапі повинні бути доступні для перегляду та роботи 24 обласні центри та одне село, яке було добавлене для перевірки працездатності деяких функцій, а саме – працездатність перемикача “Показувати тільки міста” (рис. 3.17).

Рисунок 3.16 – Список населених пунктів України

Розроблений додаток дає можливість здійснювати пошук населених пунктів за назвою, сортувати їх за алфавітом, площею та населенням (рис. 3.17).

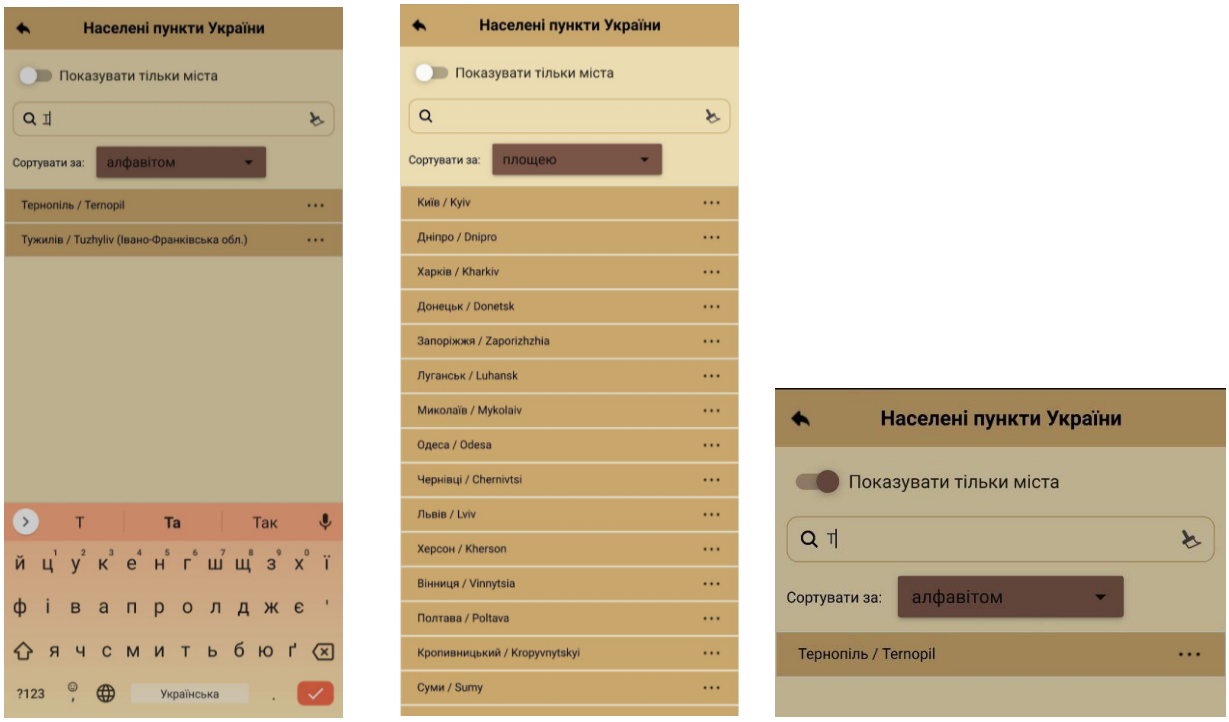

Рисунок 3.17 – Демонстрація функціоналу пошуку та фільтрації списку населених пунктів

Серед знайдених населених пунктів можна обрати однин населений пункт та переглянути структуровану за категоріями інформацію про нього (рис. 3.18). На початку сторінки нас зустрічають зображення міста, які можуть змінюватись за допомогою натискання на кнопку, що знаходиться в правому верхньому куті зображення.

Рисунок 3.18 – Перегляд інформації про населений пункт

Для обраного населеного пункту можна викликати бокову панель, що дає змогу відкрити карту та переглянути місто на ній, забронювати житло в даному місці, купити квитки на потяг або отримати більш детальну інформацію про даний населений пункт (рис. 3.19). Також є можливість переглянути список найкращих закладів даного НП (рис. 3.19), натиснувши на опцію «Топ закладів». Після натискання на один з них – відкривається веб-сторінка даного закладу в мережі інтернет.

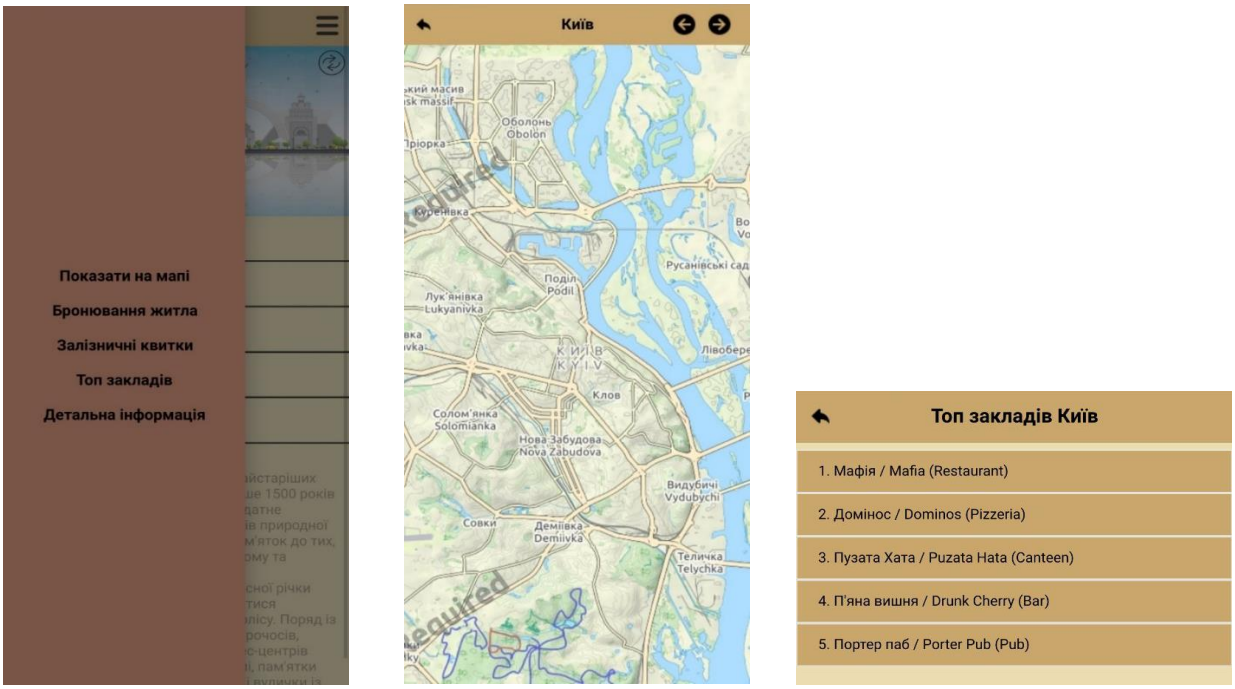

Рисунок 3.19 – Додатковий функціонал при роботі з населеним пунктом

Для роботи з маршрутами користувачу необхідно скористатися меню «Мої маршрути». Після його відкриття на екран виводиться список створених маршрутів (рис. 3.20).

Рисунок 3.20 – Список доступних маршрутів

Після вибору певного маршруту користувач може побачити список зупинок в даному маршруті (рис. 3.21). Побудований маршрут користувач може відредагувати, змінивши назву, додавши додаткові зупинки, змінивши порядок слідування (рис. 3.21). Режим редагування розблоковується після натискання на кнопку із зображенням олівця в правому верхньому куті. Після внесення змін у маршрут його назва змінює колір на червоний (рис. 3.21). При поверненні до попереднього вікна додаток запитає щодо збереження змін (рис. 3.21), що буде зроблено після згоди користувача. Вихід з програми здійснюється з головного меню системи, натискаючи на кнопку в правому

верхньому куті і супроводжується вікном з підтвердженням цієї дії, схожим на те, що зображено на рис. 3.21, тільки з іншим повідомленням.

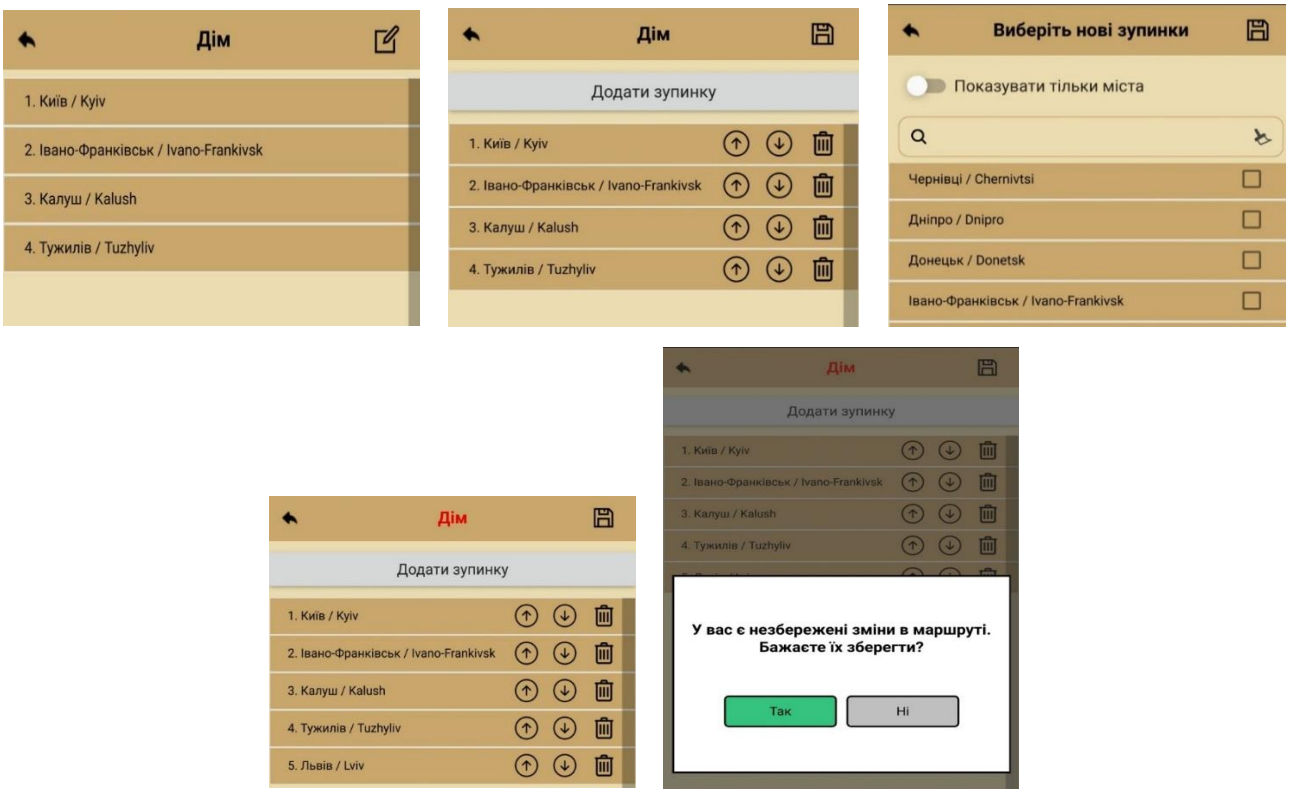

Рисунок 3.21 – Демонстрація можливості роботи з маршрутами

Після натискання на ліву верхню кнопку в головному вікні додатку, перед нами з’являється сторінка з деякою довідковою інформацією про додаток (рис. 3.22).

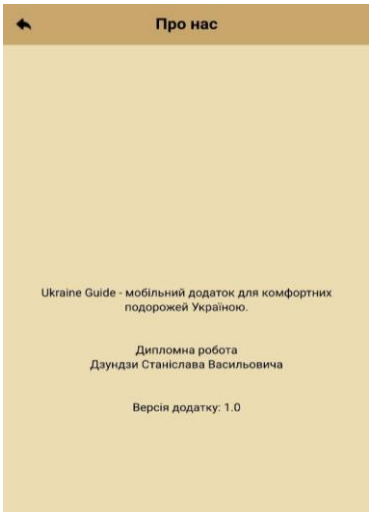

Рисунок 3.22 – Сторінка «Про нас»

Таким чином, в результаті ручного тестування, яке здійснювалось розробником даного додатку та ще двома незацікавленими особами, було вирішено, що розроблювальний додаток відповідає всім поставленим перед ним функціональним та нефункціональним вимогам, забезпечує ефективну діяльність користувача з організації власного туристичного дозвілля. Критичних помилок, які призводять до непередбачуваного виходу з програми чи якихось збоїв не виявлено. Додаток працює добре як на операційній системі Android, так і на операційній системі iOS. Всі рекомендацій від Google та Apple з приводу побудови користувацького інтерфейсу були дотримані. Тестування, яке проводилось розробником відбувалось на таких пристроях: Xiaomi Redmi Note 8 Pro 6/128 (Android), iPhone 11 128Gb (iOS).

3.5. Порівняння розробленого додатку з існуючими аналогами

Після тестування проведемо порівняння функціоналу додатку «Ukraine Guide» з іншими наявними рішеннями, які представлені в спеціалізованих магазинах додатків Play Market та App Store.

Після детального аналізу магазину Play Market від Google був знайдений тільки один працездатний додаток, який хоча б частково задовольняв функціоналу, який необхідний для комфортної та легкої організації подорожей по Україні. Цим додатком виявився «Ukraine Travel & Explore, Offline Tourist Guide», який на даний момент має лише 500+ завантажень. Після його ретельного вивчення було виявлено, що додаток не задовольняє рекомендаціям від Google та Apple з приводу побудови користувацького інтерфейсу. Його інтерфейс нагромаджений різноманітними кнопками, меню, що не дозволяє зразу інтуїтивно розібратись, як користуватись даним додатком. Локалізація додатку тільки англійська, що здається дивним та нелогічним рішенням, оскільки цільова аудиторія такого продукту – українці. Також було помічено, що деякі ресурси, такі як зображення, не відображаються, якщо пристрій не підключено до мережі інтернет в момент використання програми. У даному продукті також наявний функціонал роботи з маршрутами, але дещо в іншому

вигляді ніж у «Ukraine Guide». «Ukraine Travel & Explore, Offline Tourist Guide» дозволяє переглядати наперед заготовлені маршрути, включаючи відображення зупинок на карті, проте не надає можливість створення власних. Якась навіть коротка інформація про населені пункти України в даному додатку відсутня. «Ukraine Travel & Explore, Offline Tourist Guide» не був знайдений у магазині App Store – це говорить про те, що він може працювати тільки на пристроях з ОС Android. Але цей продукт має й свої переваги: зручне структурування закладів міст по найрізноманітнішим категоріям (об’єкти мистецтва, ресторани, парки, спортивні заклади тощо), надання інформації про їхнє розташування. Також цей додаток поширюється у двох версіях: звичайна та PRO. Вони відрізняються між собою тільки наявністю реклами, хоча в звичайній версії її не так багато і вона не дуже нав’язлива.

Після аналізу магазину App Store був знайдений додаток «WalQlike», який має схожу спеціалізацію з «Ukraine Guide». Зразу відмітимо, що він відсутній у магазині Play Market та не може бути запущений на ОС Android. На перший погляд продукт виглядає якісно і локалізований для трьох мов: української, англійської та російської. «WalQlike» надає користувачу вибір з наперед ретельно заготовлених туристичних маршрутів і може створити персоналізований маршрут, зважаючи на інтереси певної особи. Всі маршрути, які були завершені, зберігаються в історії – окремому пункті меню. Недоліками цієї програми є відсутність будь-якої інформації про населені пункти України, обмеження на запуск додатку при відключеному інтернеті, неможливість створення власних маршрутів та обмеження їх зупинками у межах певного міста.

Детальне порівняння додатків, які спеціалізуються на організації подорожей по Україні, наведено у таблиці 3.1.

Характеристики	Ukraine Guide	Ukraine Travel & Explore, Offline Tourist Guide	WalQlike
Кросплатформеність	+	-	-

Підтримка укр. мови	+	-	+
Підтримка англ. мови	-	+	+
Можливість створення та редагування маршрутів	+	-	-
Наявність вже готових туристичних маршрутів	-	+	+
Можливість перегляду маршруту на карті	-	+	+
Наявність детальної інформації про НП України	+	-	-
Можливість бронювання житла з меню додатку	+	-	-
Можливість купівлі квитків на транспорт з меню додатку	+	-	-
Можливість збереження улюблених місць	+	-	-
Наявність списку цікавих місць	-	+	-
Наявність списку популярних закладів	+	+	-
Можливість ручного розширення списку цікавих місць	-	+	-
Можливість пошуку закладів по імені	-	+	-
Можливість пошуку та фільтрації НП України	+	-	-
Можливість індивідуального автоматичного підбирання	-	-	+

турів, зважаючи на інтереси користувача			
Можливість роботи у режимі offline	+	+	-

Таблиця 3.1 – Порівняння мобільних додатків «Ukraine Guide», «Ukraine Travel & Explore, Offline Tourist Guide» та «WalQlike»

В результаті порівняння було виділено наступні переваги додатку «Ukraine Guide»:

- кросплатформеність;
- наявність довідкової інформації про населені пункти;
- можливість бронювання житла;
- можливість купівлі залізничних квитків;
- можливість створення власних маршрутів;
- працездатність більшості функцій без постійного підключення до інтернету.

Але також були знайдені і певні недоліки:

- відсутність англійської локалізації;
- відсутність наперед заготованих туристичних маршрутів;
- відсутність списку популярних місць , посортованих за категоріями, та їх адрес.

Тож, підсумовуючи, можна сказати, що на даному етапі додаток «Ukraine Guide» вже виглядає не гірше, ніж його конкуренти, а при певних допрацюваннях та публікації у магазинах, має всі шанси зайняти впевнене лідируюче місце серед них.

ВИСНОВКИ

Розвиток мобільних технологій збільшив популярність додатків для смартфонів. Сучасний туризм стає все більш технологічним, і використання смартфонів під час та для організації подорожей зростає. Сьогодні туристи можуть використовувати свої мобільні пристрої під час подорожі для спілкування, пошуку інформації та розваг. Фактично, смартфони стають супутниками подорожей.

В результаті виконання даної роботи були отримані такі результати:

– Проаналізовано вплив мобільних технологій на туристичну галузь. Було з'ясовано, що з кожним роком відбувається зростання кількості користувачів мобільних пристроїв. Час, що користувачі щоденно проводять зі своїми гаджетами, також постійно зростає;

– В сучасному світі існують різні операційні системи, під управлінням яких працюють додатки. Архітектура даних операційних систем відрізняється одна від одної принципами організації роботи та управління ресурсами. Однак зараз найбільше розповсюдження отримали ОС Android та iOS;

– Підходи та інструментарії для розробки мобільних додатків суттєво залежать від типу програмного забезпечення. Якщо для нативних додатків найкращим варіантом є використання своїх специфічних платформ, то розробка кросплатформених додатків потребує використання спеціального інструментарію, що дозволяє переносити код з платформи на платформу без змін;

– Сформовані вимоги до мобільного додатку організації подорожей по Україні. На основі визначених вимог було проведене об'єктно-орієнтоване проектування мобільного додатку, створена його архітектура;

– Проведене проектування графічного інтерфейсу мобільного додатку з урахуванням особливостей його технічної реалізації;

– Проведений аналіз підходів до розробки мобільних додатків та обрані мова C++ та Qt як інструментальні засоби реалізації кросплатформеного додатка;

– На основі отриманих моделей був розроблений мобільний додаток для організації подорожей, проведене його ручне тестування та порівняння з іншими програмами, що спеціалізуються в сфері подорожей по Україні.

Дана програмна розробка має практичну значущість, оскільки надає українським користувачам мобільних телефонів планувати власні подорожі.

Зараз робота над додатком знаходиться в стадії активної розробки та наповнення контентом. Безсумнівно, першим ділом потрібно опублікувати додаток в Play Market та App Store для того, щоб користувачі змогли його завантажити, покористуватись та залишити свої відгуки. Після цього розробка може бути продовжена за рахунок розширення функціональності програми, наприклад, зберігання запланованих подорожей в хмарних сховищах. Створення спільноти, де люди зможуть ділитись своїми маршрутами, оцінювати їх, коментувати та залишати відгуки було б корисним доповненням, враховуючи сучасні тенденції. Також, безсумнівно, необхідно зробити локалізацію додатку на англійській мові для розширення цільової аудиторії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Артеменко О.І., Пасічник В.В., Єгорова В.В. Інформаційні технології в галузі туризму. Аналіз застосувань та результати досліджень [Електронний ресурс] – Режим доступу : http://science.lpnu.ua/sites/default/files/journal-paper/2017/may/2615/814ism2015 min3-22_0.pdf
2. Wang D., Park S., Fesenmaier Daniel R. Mobile Technology, Everyday Experience and Travel. Travel and Tourism Research Association: Advancing Tourism Research Globally. 28. – 2016.
3. Пантилейкин Н.В. Мобильные приложения и их виды // Научно-методический электронный журнал Концепт. – 2016. – Т. 26. – С. 776-780.
4. Современные операционные системы / Эндрю Таненбаум. – СПб.: Питер, 2016. – 1120с.
5. Sundara Krishna Y.K., Mohan Deverkonda G.K. A Survey on Architectures of Mobile Operating Systems: Challenges and Issues. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE). Volume 2, Issue 3, March 2015, PP 73-76
6. Доля рынка мобильных операционных систем по всему миру [Електронний ресурс] – Режим доступу: http://gs.statcounter.com/os-market-share/mobile/worldwide
7. Android Development Tool [Електронний ресурс] – Режим доступу: https://developer.android.com
8. Clean Architecture Tutorial for Android: Getting Started [Електронний ресурс] – Режим доступу: https://www.raywenderlich.com/3595916-clean-architecture-tutorial-for-android-getting-started
9. The application runtime environment [Електронний ресурс] – Режим доступу: http://developer.apple.com/library/ios/#documentation/iphone/conceptual/iphoneosprogrammingguide/RuntimeEnvironment/RuntimeEnvironment.html

10. Здзиарски Дж. iPhone. Разработка приложений с открытым кодом / Здзиарски Дж.; пер. с англ. – 2-е изд., перераб. и доп. – СПб.: БХВПетербург, 2009. – 368 с.
11. Mobile Apps: User Growth, Engagement, & Retention [Електронний ресурс] – Режим доступу: https://www.kleinerperkins.com/perspectives/google-play-workshop
12. The State of Mobile 2020 [Електронний ресурс] – Режим доступу: https://www.appannie.com/ru/go/state-of-mobile-2020/
13. The Global State of Digital 2021 [Електронний ресурс] – Режим доступу: https://www.hootsuite.com/resources/digital-trends
14. Амелин К.С., Граничин О.Н., Кияев В.И., Корявко А.В. Введение в разработку приложений для мобильных платформ. – СПб. : Издательство BBM, 2011. – 511 с.
15. Соколова В.В. Разработка мобильных приложений: учебное пособие – Томск: Изд-во Томского политехнического университета, 2011. – 175 с.
16. Discover Hidden Technology That Drives Successful Apps [Електронний ресурс] – Режим доступу: https://sensortower.com/solutions/app-teardown
17. Gunavan H. The use of mobile technology for tourism development (case study: banyumas regency)// IOP Conf. Series: Journal of Physics: Conf. Series 953 – 2018.
18. Місцеві експерти [Електронний ресурс] – Режим доступу: https://maps.google.com/intl/uk/localguides/
19. Усов В. Swift. Основы разработки приложений под iOS, iPadOS и macOS. 6-е изд. дополненное и переработанное. – СПб.: Питер, 2021. – 544 с.
20. Льюис Ш., Данн М. Нативная разработка мобильных приложений / пер. с англ. А. Н. Киселева. – Москва: ДМК Пресс, 2020. – 376 с.
21. Кросплатформеність [Електронний ресурс] – Режим доступу: http://lib.mdpu.org.ua/ebook/vstup/L4.htm
22. Кроссплатформенная и нативная разработка [Електронний ресурс] – Режим доступу: http://appcraft.pro/portal/cross_platform_vs_native_app/

23. Харди Б., Филлипс Б. Программирование под Android. – СПб.: Питер, 2014. – 592 с.
24. Клифтон Я. Проектирование пользовательского интерфейса в Android. – М: ДМК, Пресс. 2017. – 452 с.
25. Meier RK. Professional Android 4 Application Development. Wrox, 2012.
26. Google розкрила сумну статистику Android 2020 року [Електронний ресурс] – Режим доступу: https://lifestyle.segodnya.ua/ua/lifestyle/science/google-raskryla-glavnyy-sekret-android-2020-goda-1433372.html
27. Румянцев Е.П. Обзор инструментальных средств для разработки мобильных приложений // Научное обозрение. Технические науки. – 2017. – № 1. – С. 88-92
28. Черников В.Н. Разработка мобильных приложений на C# для iOS и Android. – М.: ДМК Пресс, 2020. – 188 с.
29. Карпюк И. А. Сравнительный анализ мобильных приложений и инструментальных средств их разработки // Научно-методический электронный журнал «Концепт». – 2017. – Т. 31. – С. 826–830.
30. Зарицкий Д. Обзор кроссплатформенных решений для разработки мобильных приложений. – Режим доступу: https://habrahabr.ru/post/319348/
31. Інформаційний сайт «Open Standards» [Електронний ресурс] – Режим доступу: http://www.open-std.org
32. Qt Charts [Електронний ресурс] – Режим доступу: http://doc.qt.io/QtCharts/.
33. Основні етапи розробки мобільних додатків [Електронний ресурс] – Режим доступу: https://spark.ru/startup/componentix/blog/4499/osnovnie-etapi-razrabotki-mobilnih-prilozhenij
34. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language (Object Technology Series). Addison Wesley, 2003.
35. Маклафлин Б. Объектно-ориентированный анализ и проектирование. – СПб.: Питер, 2013. – 608 с.

36. Пайлон Д. UML 2 для программистов. – СПб.: Издательство Питер, 2012. – 464 с.
37. Introducing JSON [Електронний ресурс] – режим доступу: https://www.json.org/json-en.html
38. Ответы на вопросы на собеседование JSON [Електронний ресурс] – режим доступу: https://jsehelper.blogspot.com/2016/04/json.html
39. Usability Heuristics for User Interface Design [Електронний ресурс] – Режим доступу: https://www.nngroup.com/articles/tenusability-heuristics/
40. Ким В.Ю. Особенности разработки дизайна пользовательского интерфейса для мобильного приложения // Новые информационные технологии в автоматизированных системах. 2015. – № 18. – С. 479–481.
41. Клифтон Я. Проектирование пользовательского интерфейса в Android. – М.: ДМК, Пресс. 2017. – 452 с.
42. Что такое UI, прямым текстом [Електронний ресурс] – Режим доступу: https://habr.com/ru/post/321312/
43. Что такое UX дизайн? [Електронний ресурс] – Режим доступу: http://tilda.education/articles-what-is-uxdesign
44. Лафоре Р. Объектно-ориентированное программирование в C++. – СПб.: Изд. «Питер», 2015. – 928 с.
45. Шилдт Г. C++. Базовый курс / Пер. с англ. Н. Ручко. – М.: Изд.: «Вильямс», 2014. – 624 с.
46. Населення по містах України [Електронний ресурс] – Режим доступу: https://index.minfin.com.ua/ua/reference/people/town/
47. Free open source icons & illustrations [Електронний ресурс] – Режим доступу: https://iconduck.com/