

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних технологій

**Бакалаврська кваліфікаційна робота**

на тему: Розробка веб-системи для туристичних маршрутів України. Ство-  
рення мобільного додатку з пошуковими функціями

Виконав студент 4 курсу групи К-18  
Спеціальність 122 Комп'ютерні науки  
Бойко Віталій Вікторович

Керівник асистент  
Штефан Наталія Зінов'ївна

Консультант д.т.н., проф.  
Казакова Надія Феліксівна

Рецензент к.техн.н., доцент  
Гнатівська Ганна Арнольдівна

## ЗМІСТ

Перелік скорочень, умовних позначень і термінів.....	5
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Характеристика об'єкта розробки.....	11
1.2 Опис предметної області.....	11
1.3 Аналіз мобільних додатків аналогів.....	13
2 Вибір програмних засобів розробки.....	18
2.1 Функціональні та нефункціональні вимоги.....	18
2.2 Вибір операційної системи.....	18
2.3 Загальна схема роботи програми Android.....	21
2.4 Вибір інтегрованого середовища розробки.....	22
2.5 Вибір мови програмування.....	25
2.6 Мова розмітки XML.....	28
2.7 Система автоматичного збирання Gradle.....	30
2.8 Файл маніфесту.....	31
3 Проектування та розробка мобільного додатку.....	33
3.1 Моделювання процесу розробки.....	33
3.2 Проектування діаграми варіантів використання.....	35
3.3 Проектування інтерфейсу.....	39
3.4 Установка Android Studio та створення проекту.....	42
3.5 Реалізація функції пошуку.....	42
3.6 Керівництво користувача.....	43
Висновки.....	48
Перелік джерел посилання.....	50
Додаток А Лістинг основних класів.....	53

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

ОС – операційна система.

ПЗ – програмне забезпечення.

ПК – персональний комп'ютер.

Тег – елемент стандартного мови розмітки тексту в Інтернеті.

Activity – окреме вікно в Android.

ADT – Android Development Tools, плагін для Eclipse IDE, що дозволяє створювати програми під операційною системою Android.

Android – операційна система для мобільних телефонів та планшетних комп'ютерів.

AndroidManifest.xml – файл маніфесту.

Apache Ant – java-утиліта для автоматизації процесу збирання програмного продукту.

Apache Ivy – менеджер залежностей для програмних проєктів.

Apache License 2.0 – ліцензія на вільне програмне забезпечення Apache Software Foundation.

Apache Maven – це фреймворк з автоматизації та управління складанням Java-додатків.

АРК – Android Package, формат архівних файлів-додатків для Android.

Apple – американська корпорація, виробник персональної комп'ютерної техніки та програмного забезпечення.

App Store – онлайн-магазин додатків Apple для фірмових мобільних пристроїв.

C++ – універсальна мова програмування високого рівня з підтримкою декількох парадигм програмування.

C# – об'єктно-орієнтована мова програмування.

Checkstyle – інструмент статичного аналізу коду, який використовується в розробці програмного забезпечення для перевірки того, чи Java код відповідає правилам кодування.

ChromeOS – операційна система компанії Google, в основі якої лежить ядро Linux.

CMS – Content Management System, програмне забезпечення для організації веб-сайтів чи інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах.

CommerceML – єдиний стандарт обміну комерційною інформацією у форматі XML.

cXML – commerce eXtensible Markup Language, це стандартний протокол для ділових транзакцій, який призначений для передачі ділових документів серед постачальників, центрів електронної комерції та програм для закупівель.

Google Play Store – магазин додатків для мобільних пристроїв з операційною системою Android.

Groovy – об'єктно-орієнтована динамічна мова програмування, розроблена для платформи Java.

HTML – Hypertext Markup Language, стандартизована мова гіпертекстової розмітки документів для перегляду веб-сторінок у браузері.

IDE – Integrated Development Environment, інтегроване середовище розробки.

IDEF0 – методологія функціонального моделювання та графічного опису процесів, призначена для формалізації та опису бізнес-процесів.

iOS – мобільна операційна система для смартфонів, електронних планшетів.

IPA – App Store Package, формат архівних файлів програм від Apple.

Java – об'єктно-орієнтована мова програмування.

JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування.

JVM – Java virtual machine, це програма, яка розроблена для виконання та запуску інших програм на основі Java.

Kotlin – функціональна об'єктно-орієнтована мова програмування зі статичним написанням.

Linux – операційна система з відкритим вихідним кодом.

MacOS – операційна система корпорації Apple.

Manifest Editor – редактор файлу маніфесту.

Moqups – онлайн-платформа для створення дизайну прототипів додатків, сайтів діаграм та макетів.

NET – система розробки програмного забезпечення від Microsoft.

RDF – Resource Description Framework, це розроблена Консорціумом Всесвітньої павутини модель для представлення даних, особливо метаданих.

Scala – мультипарадигмова мова програмування, що поєднує властивості об'єктно-орієнтованого та функційного програмування.

SDK – Software Development Kit, набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи.

SOAP – Simple Object Access Protocol, протокол обміну структурованими повідомленнями в розподілених обчислювальних системах, базується на форматі XML.

SonarQube – платформа для перевірки коду на якість за правилами, що базуються на угодах та стандартах.

StatCounter – ірландський веб-сайт, який є інструментом для аналізу веб-трафіку.

UML – Unified Modeling Language, уніфікована мова моделювання.

URL – Uniform Resource Locator, стандартизована адреса певного ресурсу.

Use-case diagram – діаграма варіантів використання.

Web – інтернет-простір.

Windows – операційна система, яка забезпечує одночасно роботу кількох додатків (програм).

xSBL – це колекція XML технічних характеристик для використання в електронному бізнесі.

XHTML – Extensible Hypertext Markup Language, мова розмітки, яка базується на основі XML.

XLink – XML Linking Language, розроблена в консорціумі W3C мова розмітки, що дозволяє вставляти в XML документи елементи, щоб створити і описати посилання між ресурсами.

XPointer – XML Pointer Language, розширювана специфікація, що визначає способи адресації структурних елементів і фрагментів документів у форматі XML.

XSL – eXtensible Stylesheet Language, розширювана мова таблиць стилів.

## ВСТУП

В наш час через те, що дикої природи з кожним роком стає все менше, розвивається напрям туризму під назвою зелений туризм або екотуризм – це подорожі місцями, де довкілля збереглося практично у первозданному вигляді.

Основна спрямованість екотуризму – це знайомство з місцями, де рідко ступає нога людини. Кількість екологічно чистих місць зменшується через вирубку лісів, будівництво заводів і т.д., але на щастя, у світі та зокрема в Україні ще залишилися заповідні куточки, де можна відчувати себе у повній гармонії із природою.

Дуже велике значення при організації екотуризму приділяється збереженню та підтримці цілісності екосистеми. Все просто: приплив мандрівників – це приплив грошей до регіону (завдяки покупці місцевих продуктів, сувенірів, користування послугами кафе, локальних туроператорів з екотуризму). Місцевому населенню стає вигідно підтримувати екотуризм – у результаті місцевість розвивається, але природне середовище не порушується.

Екотуризм дозволяє відкривати заповідні куточки нашої країни та захоплювати красою живої природи. Екотуризм – це тури та екскурсії для любителів пригод та поціновувачів природності. При цьому необов'язково мати гарну фізичну форму. Можна підібрати відповідний тур чи екскурсію, як для екотуриста-початківця, так і для досвідченого і добре підготовленого для перебування в дикій природі мандрівника. В турі або екскурсії вам допоможуть зрозуміти всю велич нашої чудової країни. З таких поїздок можна привезти дивовижні фото заповідних та екологічно чистих місць, де ви побували.

Щоб полегшити пошук заповідних місць України, актуальним завданням є розробка мобільного додатка, оскільки майже кожна людина має смартфон, яким він може користуватися у будь-який час протягом дня. Також завдяки тому що цей мобільний додаток може працювати й без підключення до інтернету, деяку інформацію про тури та екскурсії в ньому можна дивитись у

віддалених від цивілізації природних місцях України де зовсім немає інтернету або він дуже повільно працює.

Метою бакалаврської кваліфікаційної роботи є розробка мобільного додатку для пошуку турів, екскурсій та туристичних маршрутів по біосферних та природних заповідниках України.

Загальні характеристики кваліфікаційної роботи:

- повний обсяг сторінок пояснювальної записки – 57;
- кількість рисунків – 24;
- кількість таблиць – 1;
- кількість посилань – 13.



## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **1.1 Характеристика об'єкта розробки**

Об'єктом розробки є мобільний додаток для пошуку турів, екскурсій та туристичних маршрутів по заповідниках України. Цей додаток допоможе усім любителям екотуризму зручно та швидко знайти інформацію про заповідник який вони бажають відвідати.

Під час розробки мобільного додатку буде зроблено акцент на зручність, дизайн та розмір додатка, щоб його можна було встановити навіть на гаджети з мінімальними технічними характеристиками.

Для створення мобільного додатку були сформульовані наступні завдання:

- провести характеристику об'єкта розробки;
- провести опис предметної області;
- провести порівняльний аналіз наявних мобільних додатків аналогів;
- обґрунтувати вибір програмних засобів розробки та технологій;
- провести проектування та розробку;
- підготувати керівництво користувача.

### **1.2 Опис предметної області**

Екологічний туризм – це напрямок туризму, що передбачає відвідування територій, не порушених антропогенним впливом. Метою поїздок є отримання інформації про природу регіону, знайомство з культурою, етнографією, археологічними та історичними пам'ятками. Основне завдання екотуризму полягає у збереженні у первозданному вигляді унікальних ландшафтів та вихованні у людей дбайливого ставлення до природи.

Екологічний туризм охоплює великий сектор галузі від одноденного відпочинку за містом та невеликих екскурсій для школярів до багатоденних

турів великими групами по заповідниках. За родом занять виділяють такі види екотуризму:

- науковий екологічний туризм. Експедиції вчених, практика студентів у заповідниках, польові дослідження (наприклад, врахування чисельності тварин) із залученням волонтерів;
- пізнавальний екотуризм. Спостереження за ссавцями, птахами, рідкісними рослинами, метеликами. Тури до палеонтологічних пам'яток, етнографічні поїздки, фотополювання;
- рекреаційний туризм. Активний відпочинок: кінні, піші, велосипедні походи, альпінізм, сплав річками. Пасивний відпочинок: пікніки, перебування на базах та наметових таборах з вивченням околиць [1].

В Україні екотуризм займає важливе місце серед інших видів туризму, і важливе тому, що він пов'язаний з природою. Україна багата на чудові природні куточки, м'який та унікальний клімат, природні заповідники, унікальну різноманітність ландшафтів та природних багатств. Гори, ліси, річки, моря – це унікальна природна спадщина, яка дарує Україні величезний туристично-рекреаційний потенціал. Зберігаючи природу країни та розвиваючи зелений туризм, ми робимо неоціненний внесок у здоров'я нації. Адже відпочинок на лоні природи, в екологічно чистому середовищі приносить величезну користь організму людини.

Розвиток екотуризму в Україні дозволяє не лише розвивати рекреаційно-оздоровчі комплекси, а й залучаючи туристів з усього світу вкладати кошти у природоохоронні заходи, збереження рідкісних видів рослин, тварин та природних заповідників, щоб кожне покоління українців насолоджувалося унікальною спадщиною.

Екотуризм в Україні може сприяти розв'язанню багатьох проблем, пов'язаних із частковим занепадом навколишнього середовища, тому що природні багатства нашої країни потребують догляду та шанобливого ставлення задля їх збереження. Також це збільшення зайнятості серед сільського населення, враховуючи, що народ працьовитий, гостинний та самостійно розвиває

туристичну сферу. Важливо сприяти розвитку екотуризму, проводити освітні роз'яснення щодо важливості збереження природних багатств України, збільшувати інформативність у суспільстві.

Вкладаючи кошти у такий вид відпочинку, ми не лише розвиваємо природоохоронний комплекс держави, а й сприяємо поглибленню людей у культурно-природну спадщину, а також знайомству з культурою та побутом різних регіонів нашої прекрасної країни [2].

### 1.3 Аналіз мобільних додатків аналогів

Нині створюються різноманітні мобільні додатки для пошуку турів чи екскурсій по місцях з відносно незайманою природою.

Мобільний додаток «Ecotourism» (рис. 1.1) дозволяє ознайомитися з природними місцями які знаходяться на границі між Болгарією та Румунією.

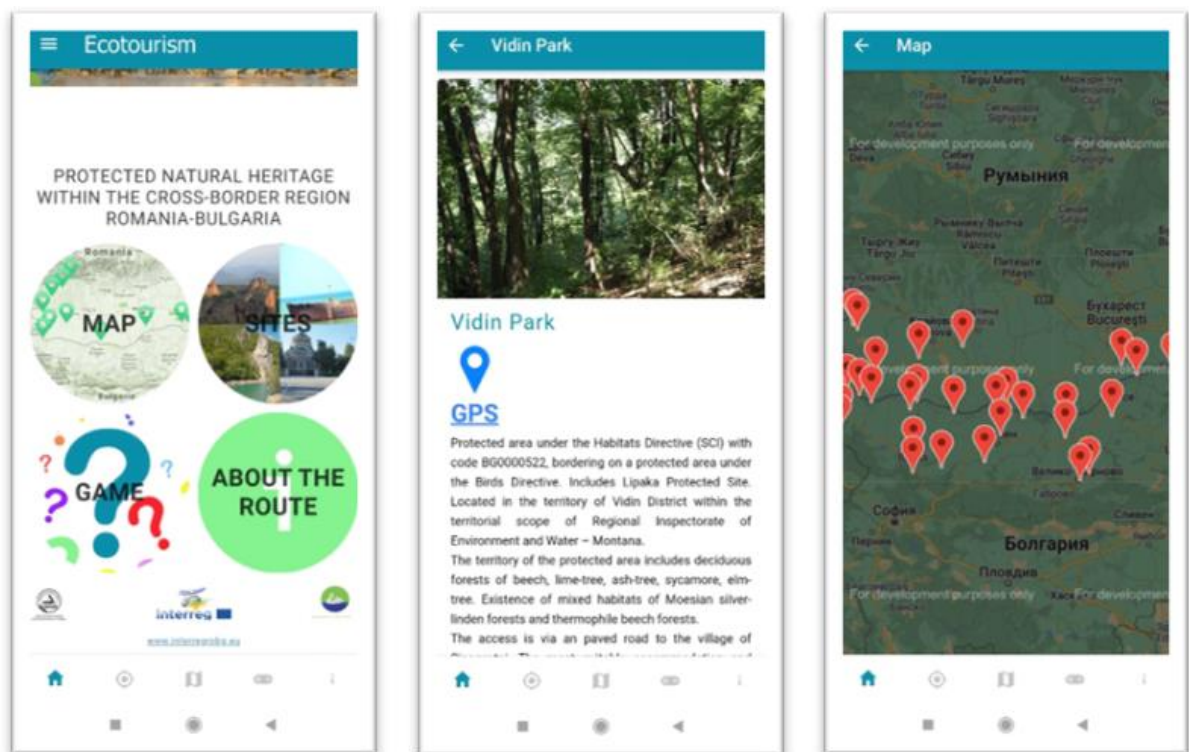


Рисунок 1.1 – Інтерфейс мобільного додатку «Ecotourism»

У цьому мобільному додатку можна подивитись карту з розташуванням природних місць Болгарії та Румунії, дізнатися коротку інформацію про ці місця, а також пограти у вікторину про них. Крім того, плюсом цього додатку є те ще можна вибрати одну з трьох мов для перегляду інформації: англійську, болгарську або румунську.

Мобільний додаток «Ecotourism and Adventures Fidesur» (рис. 1.2) використовується для пошуку природних місць штатів Мексики.

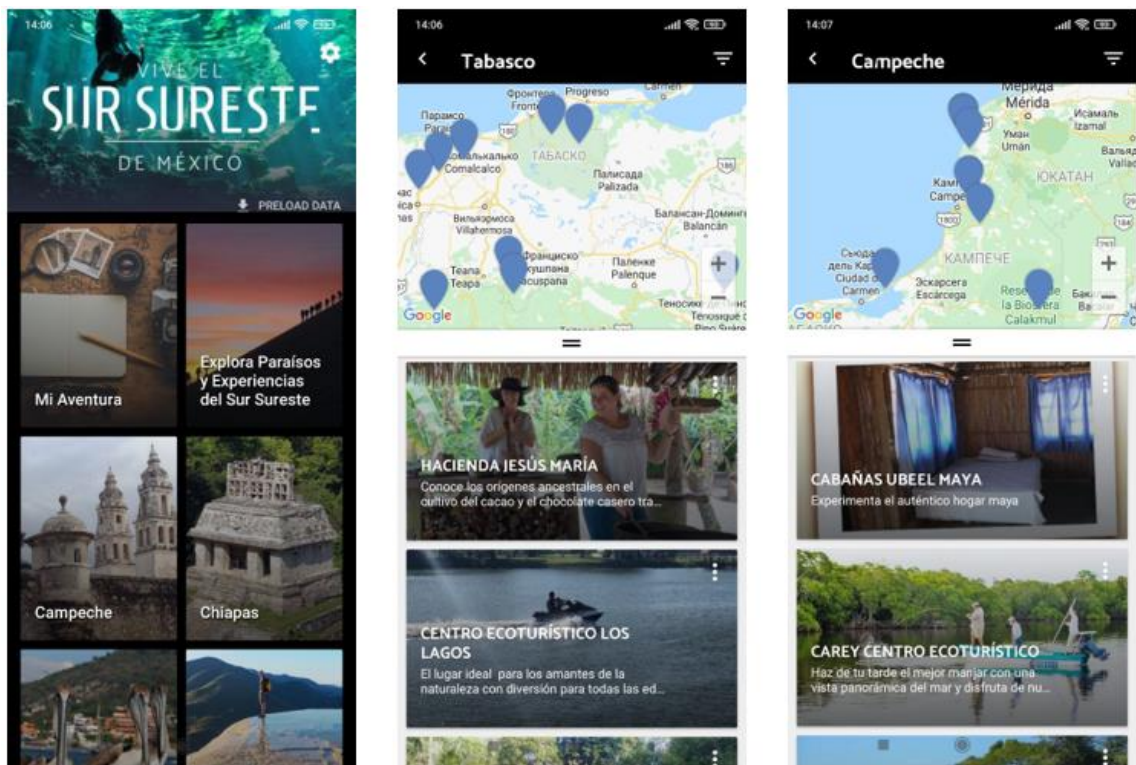


Рисунок 1.2 – Інтерфейс мобільного додатку «Ecotourism and Adventures Fidesur»

Цей мобільний додаток дозволяє вибрати штат Мексики та подивитись на карті розташування природних місць цього штату. Також у цьому додатку можна дізнатися інформацію про ці природні місця та тури чи екскурсії до них.

Мобільний додаток «Meet Nature» (рис. 1.3) є аудіогідом для подорожей природними місцями Швеції.

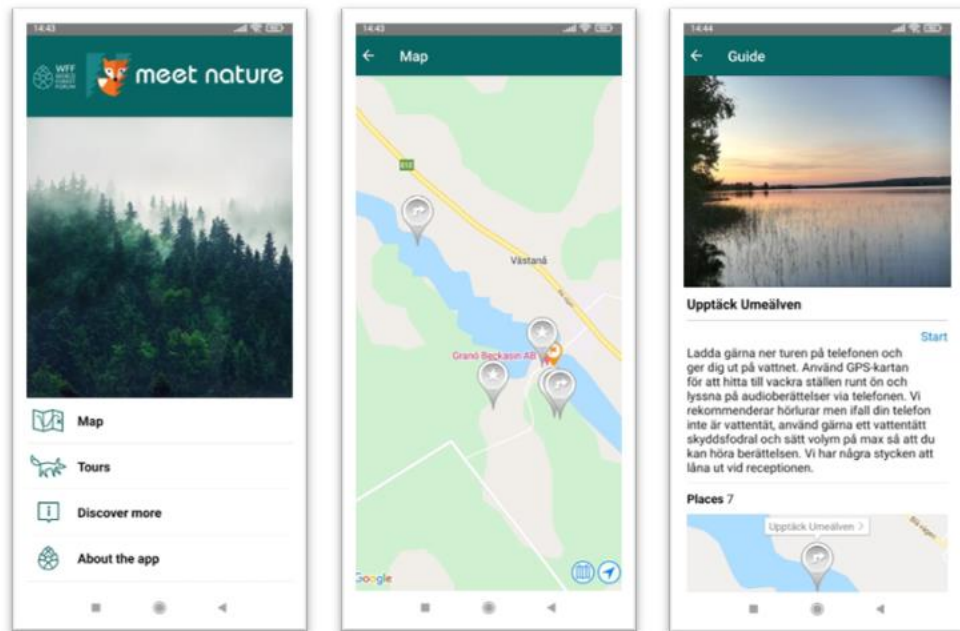


Рисунок 1.3 – Інтерфейс мобільного додатку «Meet Nature»

Завдяки цьому мобільному додатку можна подивитися на карті природні місця Швеції та за допомогою аудіогіду дізнатися більше про них.

Мобільний додаток «CapeNature» (рис. 1.4) дозволяє дізнатися інформацію про природні заповідники Південно-Африканської Республіки.



Рисунок 1.4 – Інтерфейс мобільного додатку «CapeNature»



За допомогою цього мобільного додатку можна прочитати інформацію про заповідники різних регіонів Південно-Африканської Республіки та перейти на офіційні сторінки цих заповідників.

Мобільний додаток «Benim Ormanım» (рис. 1.5) дозволяє ознайомитися з лісами та парками Туреччини.

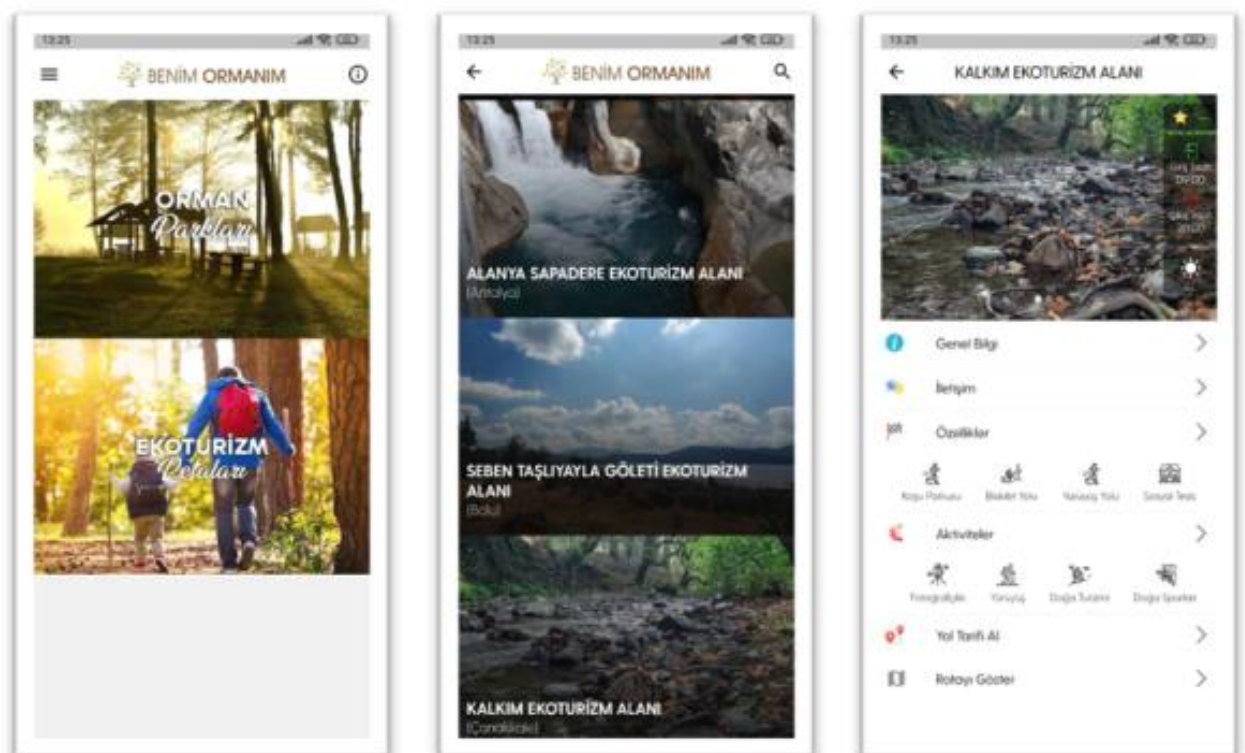


Рисунок 1.5 – Інтерфейс мобільного додатку «Benim Ormanım»

У цьому мобільному додатку присутня інформація про ліси та парки Туреччини. Також у додатку можна подивитись розташування цих лісів і парків на карті.

Мобільний додаток «Eco Tourism UAE» (рис. 1.6) дозволяє ознайомитися з екотуристичними та агротуристичними місцями Об'єднаних Арабських Еміратів.

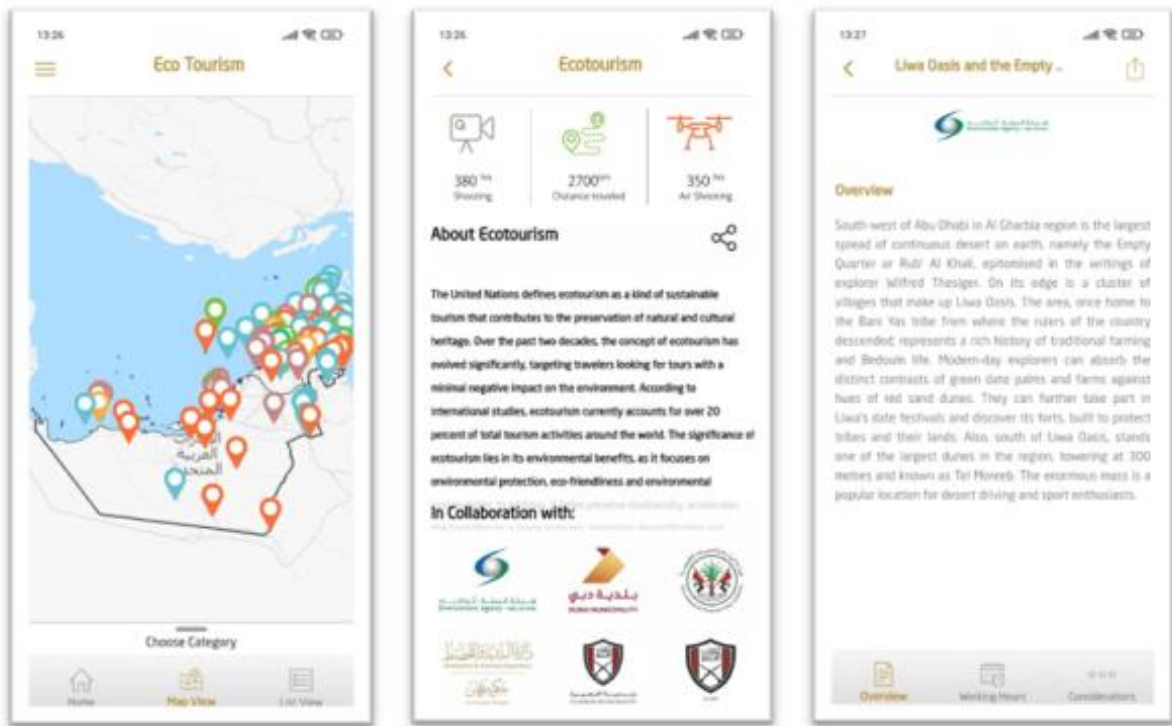


Рисунок 1.6 – Інтерфейс мобільного додатку «Eco Tourism UAE»

За допомогою цього додатку можна подивитися на карті розташування екотуристичних та агротуристичних місць Об'єднаних Арабських Еміратів та дізнатися коротку інформацію про ці місця і про екотуризм та агротуризм в цілому.

## **2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ**

### **2.1 Функціональні та нефункціональні вимоги**

Перед початком розробки мобільного додатку були визначені функціональні та нефункціональні вимоги до нього.

До функціональних вимог належить:

- відображення інформації про додаток;
- відображення списку заповідників;
- функція пошуку по списку заповідників;
- відображення інформації про заповідники;
- відображення картинок природи заповідників;
- відображення карт заповідників;
- відображення інформації про тури, екскурсії та туристичні маршрути по заповідниках.

До нефункціональних вимог належить:

- смартфон з ОС Android версією 4.1+.

### **2.2 Вибір операційної системи**

Найбільш популярними операційними система для мобільних додатків є Android та iOS.

Android лідирує серед операційних систем для мобільних пристроїв у всьому світі. За даними сервісу StatCounter на січень 2022 року, частка Android серед мобільних пристроїв становить 69,7%, iOS займає друге місце з 29,4% мобільних пристроїв [3].

Основні плюси розробки мобільних програм з ОС Android:

- невеликий бюджет для "входу". Щоб почати програмувати, достатньо бюджетного ноутбука із системою Linux або Windows. Не обов'язково купувати ноутбук вартістю тисячі доларів, вистачить навіть домашнього ПК. Запускати та тестувати готову програму можна на емуляторі;



- порівняно невелика плата за реєстрацію облікового запису розробника в Google Play: реєстрація в Google Play Store коштує \$25, внесок одиноразовий;
- Java – популярна мова, яка використовується не лише у мобільній розробці. А значить, якщо розробнику не сподобається розробка мобільних програм, він може знайти застосування своїм знанням в іншій сфері;
- велика спільнота. У мережі багато корисної інформації для новачків: статті, відео та конференції;
- Android постійно вдосконалює безпеку, розробляючи приватні місця для зберігання даних;
- Android дає розробникам більше простору: можна підключати сторонні сервіси, тобто додавати до програм свої служби, які працюють у фоновому режимі й не вимагають участі користувача. Такі послуги обробляють інформацію, запускають повідомлення та інше;
- можливість підключати сторонні бібліотеки – це ще одна вагома перевага. Наприклад, до Java можна приєднати «готові вихідні дані» – бібліотеки C++;
- різноманітність гаджетів на базі Android – це плюс, і мінус для розробника. Додаток можна тестувати на пристрої будь-якої цінової категорії, з будь-якими характеристиками – це перевага, оскільки не обов'язково мати в наявності новенький флагман. Але з іншого боку, через велику різноманітність пристроїв, процесорів, діагоналей екранів та версій операційних систем деякі програми можуть зображатися некоректно.

Основні плюси розробки мобільних програм з ОС iOS:

- єдина екосистема гарантує плавний перехід між пристроями. На нових пристроях додаток не зависатиме;
- єдина для всіх пристроїв безпека та шифрування – турбота і про розробників, і користувачів;

- використання директорій, які обмежують роботу інших додатків та дозволяють збалансовано використовувати ресурси пристрою. Не потрібно нарощувати потужність системи, щоб запустити важку програму, оскільки можна грамотно розпорядитися вже наявною;
- регулювання вимог до використання операційної системи, залученим обсягам пам'яті та інтерфейсу. Це забезпечує високу якість програм в App Store;
- інтуїтивність керування операційною системою: програми мають єдину систему управління, тому після установки у користувача не виникає питань, як повернутися на крок назад або вийти із програми;
- середній цикл життя пристроїв довше, ніж Android, тому вони отримують оновлення в середньому протягом 5-7 років після виходу ринку;
- жорстка модерація в App Store – висока якість сервісів.

Мінусів у кожній з операційних систем не так багато, як плюсів. Але вони все ж таки є.

Недоліки операційної системи Android:

- в Android через велику кількість пристроїв з різними технічними характеристиками можуть виникати проблеми з роботою та зображенням однієї й тієї ж програми. На різних пристроях навіть у межах однієї версії операційної системи програма може виглядати по-різному;
- через простий "вхід" для розробників і відсутність жорсткої модерації в Google Play Store багато "сміття";
- ОС оновлюється настільки часто і змінюється так швидко, що розробники не завжди встигають приділяти увагу постійній підтримці продуктів.

Недоліки операційної системи iOS:

- висока ціна входу. Техніка Apple не найдешевша і на старті вона обов'язково знадобиться;

- закритість системи. Розробники відносять це до суттєвого мінусу та бачать у цьому обмеження своїх можливостей;
- вартість облікового запису розробника в App Store (\$99 на рік) для багатьох теж стає мінусом;
- висока конкуренція на ринку праці. Початківцям знадобиться багато завзятості та практики, перш ніж вони зможуть по-справжньому яскраво та впевнено проявити себе [4].

З огляду на перераховані переваги та недоліки систем iOS та Android, перевага надається системі Android через багату кількість інформації про цю операційну систему та можливість тестування мобільних додатків навіть на домашньому ПК. Тож розробка застосунку буде виконуватися для пристроїв з ОС Android.

### 2.3 Загальна схема роботи програми Android

Програми для Android у своїй роботі використовують вікна (аналогічно Windows), проте в Android вищезазначені вікна мають іншу назву – Activity. Як і в Windows, кожне вікно повинно мати свій життєвий цикл та деякі свої особливості.

При створенні нового вікна викликається метод `onCreate()`, при розробці цей метод перевизначається і в ньому відбувається ініціалізація програми та її компонентів.

Далі викликаються методи `onStart()` та `onResume()`. Обидва методи викликаються перед відображенням вікна при його створенні або відновленні (при перемиканні з іншого додатка, при розгортанні згорнутого додатка тощо).

При згортанні викликаються методи `onPause()` та `onStop()`. При закритті програми та вікна викликається `onDestory()`, в даному методі можна зберегти власні дані та параметри. [5].

Загальна схема життєвого циклу програми Android представлена на рис. 2.1.

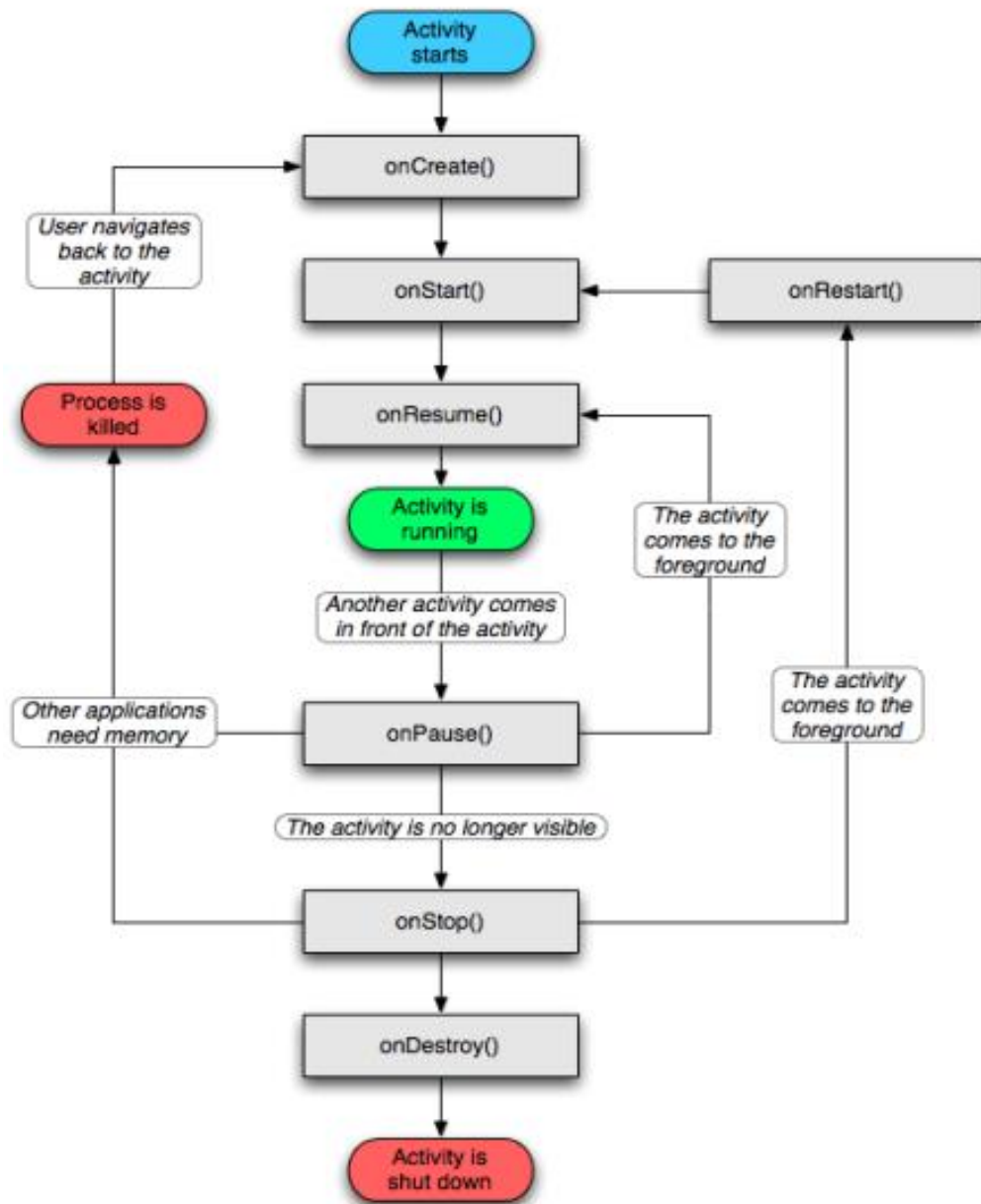


Рисунок 2.1 Схема життєвого циклу програми Android

## 2.4 Вибір інтегрованого середовища розробки

Популярними середовищами розробки під ОС Android є Xamarin, Eclipse, Android Studio.

Xamarin – це платформа з відкритим вихідним кодом для створення сучасних та продуктивних програм для iOS, Android та Windows за допомогою NET. Xamarin – це рівень абстракції, який керує взаємодією загального коду з

базовим кодом платформи. Xamarin працює в керованому середовищі, яке забезпечує такі зручності, як виділення пам'яті та складання сміття.

Xamarin підтримується компанією Microsoft та інтегрована у середовище Visual Studio. Мова програмування – C#. Середовище розробки Visual Studio має безліч компонентів для розробки та налагодження застосувань та при цьому зручний та не перевантажений інтерфейс.

Xamarin дозволяє розробникам спільно використовувати в середньому 90% своєї програми на різних платформах. Цей шаблон дозволяє розробникам писати всю свою бізнес-логіку однією мовою (або повторно використовувати наявний код програми), але при цьому досягати власної продуктивності, зовнішнього вигляду та функцій на кожній платформі.

Платформа Xamarin орієнтована на розробників, перед якими стоять такі завдання:

- спільне використання коду та тестів та бізнес-логіки на різних платформах;
- написання кросплатформових додатків мовою C# у Visual Studio.

Програми Xamarin можна писати на ПК і компілювати у власні пакети програм, такі як .apk на Android або .ipa на iOS [6].

Проте, через невисоку популярність даного середовища розробки та проблем з пошуком інформації від нього прийшлося відмовитися.

Eclipse – це інтегроване середовище розробки модульних кросплатформних додатків.

Для розробки додатків Android в Eclipse пропонується плагін для Eclipse IDE, званий ADT. Цей плагін є потужним, інтегрованим середовищем розробки додатків для Android. Це розширює можливості Eclipse, що дозволяє швидко створити нові Android проекти, побудувати інтерфейс програми, налагодити програму, а також підписувати та експортувати пакети програми (APK) для розповсюдження [7].

На жаль, цей плагін більше не підтримується з 2015 року, тому замість нього рекомендують застосовувати Android Studio.

Android Studio – це середовище розробки, ПЗ, в якому є інструменти та сервіси для створення нових додатків Android. Багато наявних операційних систем мають таке середовище розробки.

У цій програмі можна знайти інструменти, необхідні для створення програм від коду до дизайну інтерфейсу користувача.

Android Studio, можна завантажити на комп'ютер під керуванням Windows, MacOS, ChromeOS або Linux, щоб мати можливість працювати над програмою. Цей інструмент підтримується всіма популярними ОС, тому жоден розробник не зіткнеться з проблемами доступу до цього середовища. Крім того, завантаження безкоштовне, що, безперечно, є плюсом для тих, хто хоче використовувати цей інструмент на своєму комп'ютері.

Android Studio розроблена таким чином, щоб користувачі могли виконувати кілька дій, всі з яких пов'язані зі створенням програм для Android. Ось чому багато хто бачить у цій програмі важливий інструмент для розробників. Android Studio можна використовувати для різних речей, наприклад:

- створення програми. Це основна мета, запропонована цим програмним забезпеченням, оскільки завдяки простому і наочному інтерфейсу, користувачі можуть створювати власні додатки. Крім того, в залежності від типу програми надаються дизайни та інтерфейси, що допомагають у розробці;
- використання як емулятор Android. Android Studio має емулятор, який розробники можуть використовувати для тестування функцій операційної системи або створених програм, щоб побачити, чи є щось для зміни, чи все працює, як і очікувалося;
- аналіз інших програм. Однією з доступних функцій є аналіз або використання роботи інших розробників, оскільки можна використовувати програми з відкритим вихідним кодом, щоб дізнатися про те, як вони були розроблені або використовувати частину їхнього коду у своєму власному проєкті [8].

Більш детальне порівняння інтегрованих середовищ розробки наведено у табл. 1.

Таблиця 1 – Детальне порівняння інтегрованих середовищ розробки.

Технологія або характеристика	Android Studio	Xamarin (Visual Studio)	Eclipse IDE
Платформи	Android	Android, Windows, iOS	JVM
Підтримка нативних функцій	Максимальна для Android	Висока	Мінімальна
Інтерфейс	Зручний	Зручний	Задовільний
Алгоритми рефактору коду	Відмінні	Відмінні	Незадовільні
Мова програмування	Java, Kotlin	C#	Java
Підтримка	Висока	Задовільна	Висока
Інтеграція з засобами відладки	Максимальна для Android	Висока	Незадовільна

За підсумком аналізу наявних середовищ розробки, було обрано інтегроване середовище розробки Android Studio через зручність його інтерфейсу, та тому що в інтернеті є багато інформації про налаштування та роботу з Android Studio.

## 2.5 Вибір мови програмування

У інтегрованому середовищі Android Studio є можливість вибрати між мовою програмування Java та Kotlin. Розробка Android-програм на Java дає і розробникам, і користувачам широкий перелік переваг.

Із застосуванням платформи можна отримувати програми, які на 100% сумісні з різними типами операційних систем.

Мова програмування Java:

- універсальна;
- проста у вивченні;
- платформи-незалежна
- дозволяє писати програми-модулі;
- застосовується у мережевому програмуванні;
- динамічна, ефективна.

При написанні додатків код виходить багатослівним і довгим, що подобається не всім. Сам процес написання постійно спрощується, стає дедалі ясним і чітким. Це мінімізує можливість помилок.

Трактування конструкцій Java переважно прості, мають однозначне прочитання. Після оновлення старі коди не ламаються.

Java має плюси, які можна винести в ряд конкурентних переваг перед Kotlin. Перший – портативність. Із застосуванням віртуальної машини Java програми мовою запускаються у будь-якій системі. Максимальна сумісність спричиняє лідерство мови розробника в сегменті створення Android-додатків.

Спільнота Java величезна. Завдяки ресурсам типу GitHub, Stack Overflow, розробник може отримати швидку допомогу та підтримку з будь-якої проблеми. Доступний широкий перелік готових бібліотек SDK для спрощення розробки.

Відстає Java у плані швидкості реалізації поточних завдань. Потрібно більше пам'яті у порівнянні з Kotlin та іншими мовами програмування, робота є досить повільною.

Android розробка на Kotlin має не менший попит, ніж Java. Мова Kotlin анонсована як друга мова Android. Є підтримка робочих відносин з Google. Розробники Kotlin стверджують, що програми можуть бути набагато цікавішими, розробка швидкою та цікавою. Поки що спільнота мови постійно збільшується.



Мова програмування Kotlin має такі переваги та можливості:

- коди відкриті, впровадження безкоштовні;
- програми застосовують фреймворки, бібліотеки Java;
- доступні компіляції в байт код JVM, JavaScript;
- процес вивчення простий;
- рівень безпеки високий;
- перетворення Java на Kotlin відбувається автоматично.

Завдяки Kotlin написання кодів коротке та виразне. Є вбудована підтримка від IDE. Для порівняння – те, що у Java розміщується на кількох рядках, тут займає лише один рядок.

Kotlin має підвищену продуктивність, інтуїтивний та послідовний синтаксис, що покращує швидкість проведення робіт. Для створення готової програми потрібно менше рядків із кодом і, відповідно, часу.

Сумісність Kotlin та Java 100%, тобто методи Java можна легко викликати з Kotlin. Це плюс для розробників та організацій-замовників із кодовою базою Java. Підтримка мови проста, оскільки вона інтегрована в IDE, SDK. Список інструментів для роботи можна використовувати стандартний.

Останні версії мають зворотну сумісність із попередніми. Це зумовлює надійність кодів, позбавляє розробників необхідності постійної доробки кодів різних версій. У вивченні Kotlin набагато простіше за Java, наявність знань у питанні розробки додатків для смартфонів не обов'язкова.

Android Studio пропонує розширену підтримку для Kotlin, повний перелік інструментів адаптації. Програмісти можуть працювати на Kotlin з Java одночасно. Підтримка спільноти достатня, хоча у Java вона більша. Поки що мовою працює не так багато розробників. Потрібно приділити увагу основним функціям та створенню додатків з їх застосуванням [9].

Отже, з врахуванням усіх недоліків та переваг була обрана мова програмування Java, через велику кількість статей з прикладами додатків на Android розроблених за допомогою мови Java.

## 2.6 Мова розмітки XML

Інтегроване середовище розробки Android Studio використовує мову розмітки XML. XML, eXtensible Markup Language – рекомендована Консорціумом Всесвітньої павутини мова розмітки, що фактично є зведенням загальних синтаксичних правил. XML – текстовий формат, призначений для зберігання структурованих даних (замість чинних файлів баз даних), для обміну інформацією між програмами, а також для створення на його основі більш спеціалізованих мов розмітки (наприклад, XHTML).

Сьогодні XML може використовуватися в будь-яких додатках, яким потрібна структурована інформація – від складних геоінформаційних систем, з гігантськими обсягами інформації, до звичайних "одно комп'ютерних" програм, що використовують цю мову для опису службової інформації. При уважному погляді на навколишній інформаційний світ можна виділити безліч завдань, пов'язаних зі створенням та обробкою структурованої інформації, для вирішення яких може використовуватися XML:

- ця технологія може бути корисною для розробників складних інформаційних систем, з великою кількістю додатків, пов'язаних потоками інформації з різною структурою. У цьому випадку XML-документи виконують роль універсального формату для обміну інформацією між окремими компонентами великої програми, об'єктами різних програмних комплексів, інформаційних систем та мереж;
- XML є базовим стандартом для нової мови опису ресурсів, RDF, що дозволяє спростити багато проблем у Web, пов'язаних з пошуком потрібної інформації, забезпеченням контролю за вмістом мережевих ресурсів, створенням електронних бібліотек і т.д;
- мова XML дозволяє описувати дані довільного типу та використовується для представлення спеціалізованої інформації, наприклад хімічних, математичних, фізичних формул, медичних рецептів і т.д. Це означає, що XML може бути потужним доповненням до HTML для

поширення в Web "нестандартної", тобто об'єктно-орієнтованої інформації. Можливо, у найближчому майбутньому XML повністю замінить собою HTML, принаймні перші спроби інтеграції цих двох мов уже робляться (специфікація XHTML);

- інформація, що міститься в XML-документах, може змінюватися, передаватися на машину клієнта та оновлюватися частинами. Специфікації XLink і Xpointer, що розробляються, дозволять посилатися на окремі елементи документа, з урахуванням їх вкладеності та значень атрибутів. Використання стильових таблиць (XSL) дозволяє забезпечити незалежне від конкретного пристрою виводу відображення документів XML. XML може використовуватися у звичайних програмах для зберігання та обробки структурованих даних в єдиному форматі.

XML-документ є звичайний текстовий файл, у якому за допомогою спеціальних маркерів створюються елементи даних, послідовність і вкладеність яких визначає структуру документа та її зміст. Основною перевагою XML документів є те, що при відносно простому способі створення та обробки (звичайний текст може редагуватися будь-яким тестовим процесором та оброблятися стандартними XML аналізаторами), вони дозволяють створювати структуровану інформацію, яку добре розуміють комп'ютери.

Оскільки XML є досить абстрактною мовою, було розроблено словники XML. Словник дозволяє розробникам домовитися про певний кінцевий набір імен тегів та атрибутів цих тегів. Одним із перших словників є XHTML, який розуміють більшість браузерів. XHTML часто використовують для зберігання та редагування контенту в CMS.

Були створені більш спеціалізовані словники, наприклад, протокол передачі даних SOAP, який не є людино-орієнтованим. Є комерційні словники, такі як CommerceML, xCBL і cXML які використовуються для передачі даних, орієнтованих на торговельну діяльність, ці словники включають опис системи замовлень, постачальників, продуктів та інше [10].

## 2.7 Система автоматичного збирання Gradle

Android Studio підтримує систему автоматичного збирання Gradle.

Gradle – це система автоматичного збирання, що працює на платформі Java. Gradle дозволяє створювати проекти на Java, Scala, Groovy або навіть на C++.

Gradle поєднує в собі сильні сторони Apache Maven і Apache Ant: він поєднує в собі використання угод у стилі Maven з гнучкістю Ant для опису завдань складання з сильною узгодженістю в інтерфейсі.

Інструмент був розроблений для компіляції багатопроєктних файлів, які зазвичай займають багато місця. Його робота полягає в серії завдань компіляції, які виконуються або послідовно, або паралельно. Додаткова компіляція підтримується шляхом визначення, які частини дерева компіляції досі актуальні, жодна задача, яка залежить лише від цих частин, не буде перезапущена. Крім того, Gradle підтримує кешування проміжних компонентів компіляції, які також можуть бути розподілені спільною мережею. Вебсервіс дає змогу візуалізувати етапи компіляції. Нарешті, система плагінів дозволяє розширити можливості програмного забезпечення для підтримки додаткових функцій та інших мов програмування.

Gradle – це безкоштовне програмне забезпечення, випущене під ліцензією Apache License 2.0. Його первісна версія датується 2007 роком.

Gradle дозволяє записувати завдання збирання у файл збирання, використовуючи мову Groovy. Можна імпортувати стандартні завдання, які можуть створювати програми з використанням однієї або декількох мов (Java, Groovy) або дозволяють виконувати дії з розробки програмного забезпечення, такі як виконання модульних тестів, забезпечення якості коду (SonarQube, Checkstyle).

Gradle використовує деякі сильні ідеї Maven:

- життєвий цикл;
- управління залежностями, наприклад Apache Ivy або Maven;

- соглашение вместо конфигурации;
- сховище (або склади).

Gradle має такі переваги:

- можливість скрипту побудови у Groovy у файлі побудови;
- можливість змінити поведінку певних завдань за умовчанням;
- компактний запис для опису залежностей;
- виробничий двигун, призначений для створення багатомовних проєктів.

Gradle дозволяє легко створювати проєкти з використанням мов, які відрізняються від Java. Перехід з Maven на Gradle дуже простий для проєкту, що дотримується угоди Maven [11].

## 2.8 Файл маніфесту

Файл маніфесту `AndroidManifest.xml` надає основну інформацію про програму системи. Кожна програма повинна мати свій файл `AndroidManifest.xml`. Редагувати файл маніфесту можна вручну, змінюючи XML-код або через візуальний редактор `Manifest Editor`, який дозволяє здійснювати візуальне та текстове редагування файлу маніфесту програми.

Файл маніфесту виконує наступні дії:

- оголошує ім'я Java-пакета програми, яка є унікальним ідентифікатором;
- описує компоненти програми – діяльності, служби, приймачі широко-мовних намірів та контент-провайдери, що дозволяє викликати класи, які реалізують кожен із компонентів, та оголошує їх наміри;
- містить список необхідних дозволів для звернення до захищених частин API та взаємодії з іншими програмами;
- оголошує дозволи, які сторонні програми повинні мати для взаємодії з компонентами цієї програми;

- оголошує мінімальний рівень API Android, необхідний роботи програми;
- перераховує пов'язані бібліотеки;

Файл маніфесту інкапсулює всю архітектуру Android-програми, його функціональні можливості та конфігурацію. У процесі розробки програми цей файл постійно редагують, змінюючи його структуру та доповнюючи новими елементами та атрибутами.

Кореневий елемент маніфесту є `<manifest>`. Крім цього елемента обов'язковими елементами є теги `<application>` і `<uses-sdk>`. Елемент `<application>` є основним елементом маніфесту та містить безліч дочірніх елементів, що визначають структуру та роботу програми. Порядок розташування елементів, що є одному рівні, довільний. Усі значення встановлюються через атрибути елементів. Крім обов'язкових елементів, згаданих вище, у маніфесті в міру потреби використовуються інші елементи [12].

## 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

### 3.1 Моделювання процесу розробки

Процес розробки мобільного додатку змодельовано за допомогою методології функціонального моделювання IDEF0.

Методологія IDEF0 складається з контекстної діаграми та діаграм декомпозиції.

Контекстна діаграма – це діаграма самого верхнього рівня, що представляє систему загалом, у вигляді "чорного ящика", і зв'язує її із зовнішнім світом за допомогою інтерфейсних дуг. Контекстна діаграма складається з одного функціонального блоку, будь-якої кількості стрілок, мети моделювання й точки зору.

Діаграми декомпозиції призначені для деталізації функцій і виходять при розбитті контекстної діаграми на великі підсистеми й описують кожну підсистему та їхню взаємодію.

Відповідно до визначених даних була створена контекстна діаграма, яка представлена на рис. 3.1.

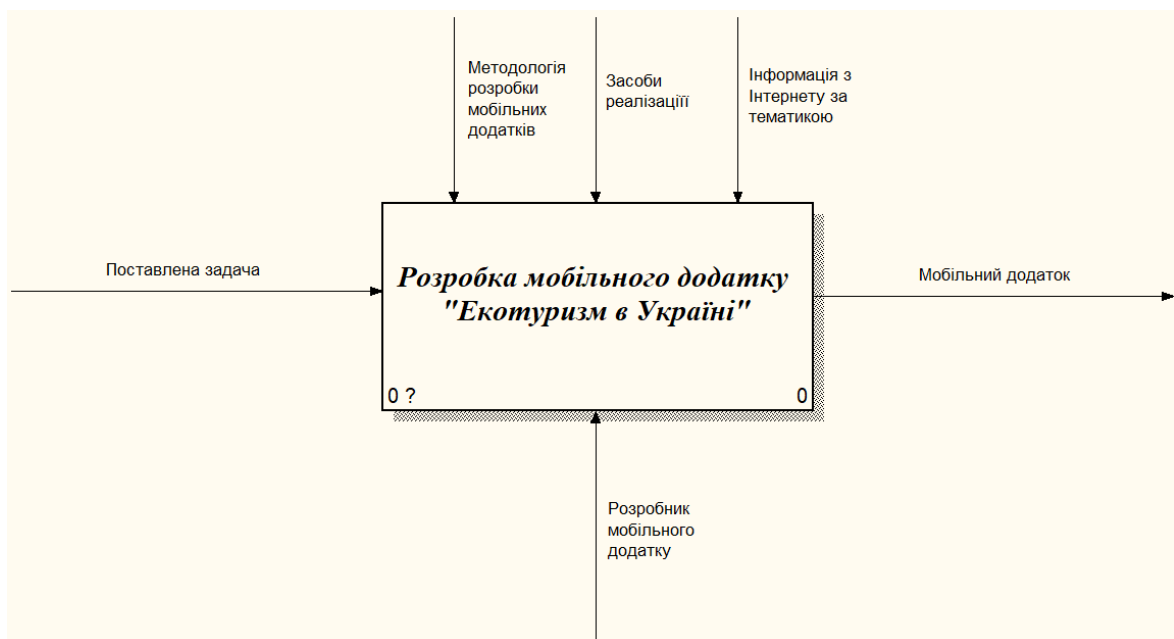


Рисунок 3.1 – Контекстна діаграма

Після опису основної функції виконується функціональна декомпозиція, тобто система розділяється на підсистеми, і кожна з них зображується окремо. Декомпозиція робиться у зв'язку тим, що контекстна діаграма надає лише формальний опис проекту. Підсистеми можна ділити безліч разів для більшої степені подробиць.

Діаграма декомпозиції першого рівня (рис. 3.2) складається з 3 функціональних блоків:

- аналіз предметної області;
- проектування;
- створення мобільного додатку з пошуковими функціями.

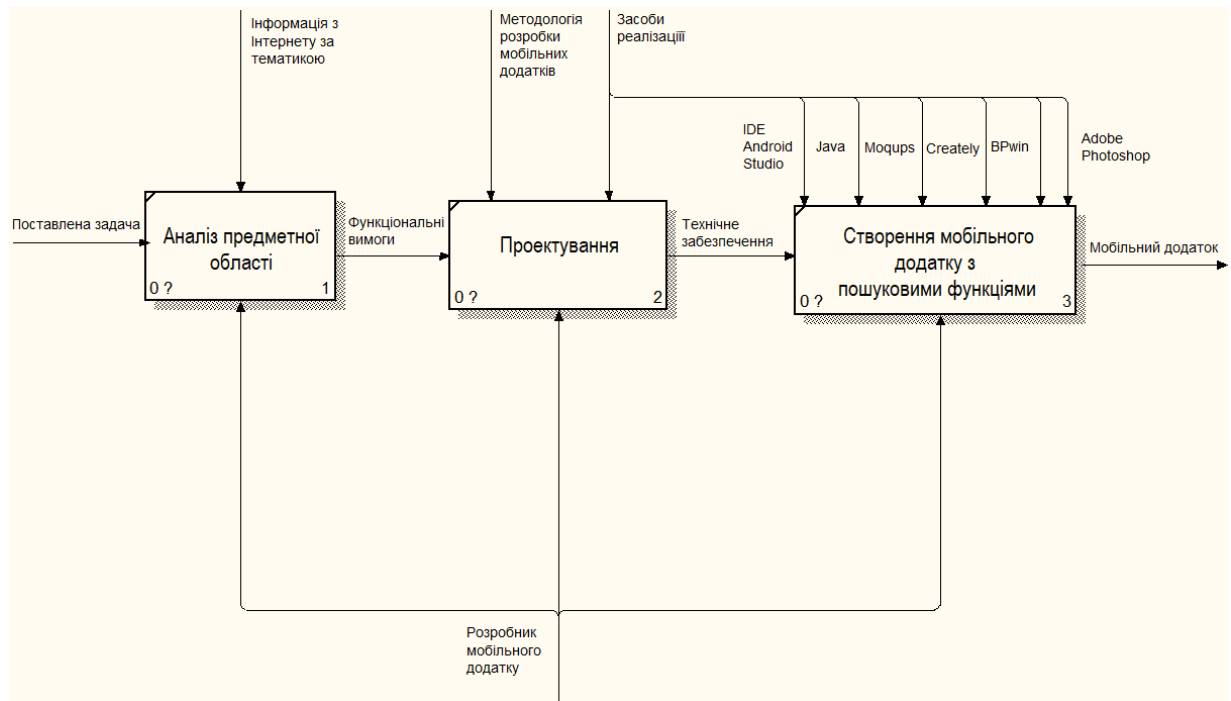


Рисунок 3.2 – Діаграма декомпозиції першого рівня

На рис. 3.3 представлена діаграма декомпозиції другого рівня, яка складається з 9 функціональних блоків:

- аналіз мобільних додатків аналогів;
- вибір ОС;



- вибір IDE;
- вибір мови програмування;
- проектування Android-додатку;
- додавання та зберігання даних;
- побудова діаграм IDEF0 та Use-case;
- розробка окремих програмних модулів;
- створення керівництва користувача.

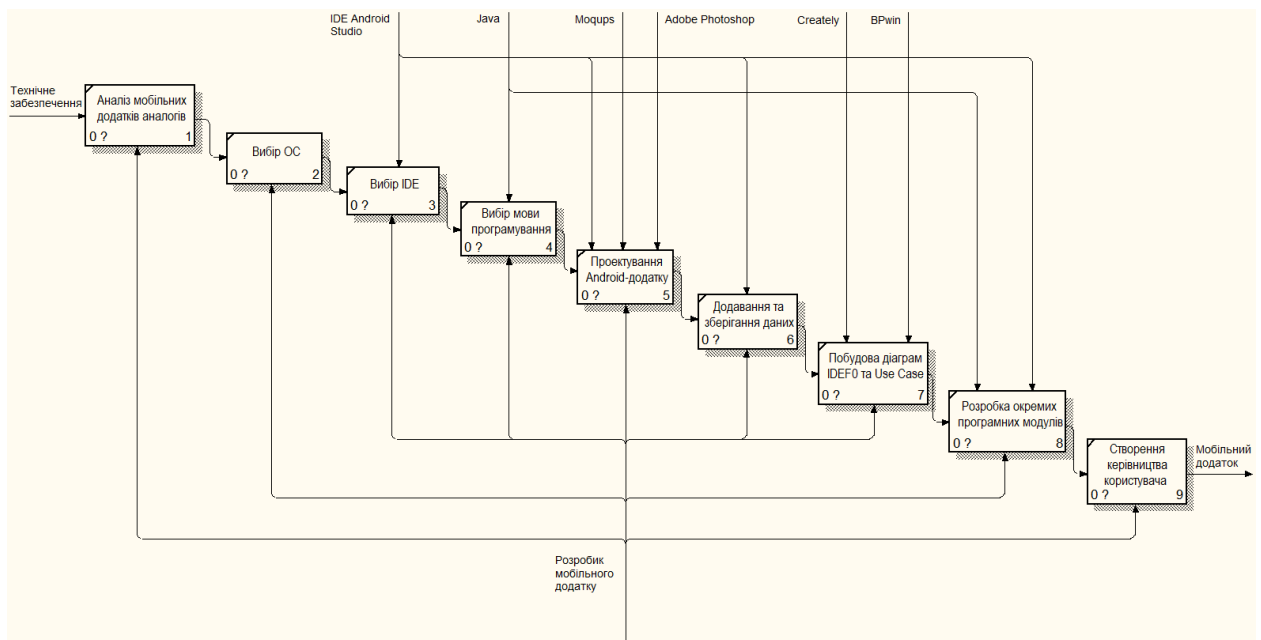


Рисунок 3.3 – Діаграма декомпозиції другого рівня

### 3.2 Проектування діаграми варіантів використання

Діаграма варіантів використання (use-case diagram) в UML – діаграма, що відображає відносини між акторами та прецедентами і є складовою моделі прецедентів, що дозволяє описати систему на концептуальному рівні.

Розробка діаграми має такі цілі:

- визначити загальні межі та контекст модельованої предметної області;

- сформулювати загальні вимоги до функціональної поведінки проектованої системи;
- розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних та фізичних моделей;
- підготувати вихідну документацію для взаємодії розробників системи з її замовниками та користувачами.

Проектована система представляється як безліч сутностей чи акторів, що взаємодіють із системою за допомогою варіантів використання. При цьому актором або дійовою особою називається будь-яка сутність, що взаємодіє із системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може бути джерелом впливу на систему, що моделюється так, як визначить сам розробник. Варіант використання служить для опису сервісів, які система надає актору.

Окремий варіант використання позначається на діаграмі еліпсом, усередині якого міститься його коротка назва або ім'я у формі дієслова з пояснювальними словами.

Ціль варіанта використання полягає в тому, щоб визначити закінчений аспект або фрагмент поведінки деякої сутності без розкриття її внутрішньої структури. В якості такої сутності може виступати система або будь-який елемент моделі, який має власну поведінку.

Кожен варіант використання відповідає окремому сервісу, який надає сутність, що моделюється, за запитом актора, тобто визначає спосіб застосування цієї сутності. Сервіс, який ініціалізується на запит актора, є закінченою неподільною послідовністю дій. Це означає, що після того, як система закінчить обробку запиту, вона має повернутися у вихідний стан, щоб бути готовою до виконання наступних запитів.

Варіанти використання можуть застосовуватися як специфікації зовнішніх вимог до проектованої системи, так і для специфікації функціональної поведінки вже наявної системи. Безліч варіантів використання загалом має визначати всі можливі сторони очікуваної поведінки системи. Крім цього,

варіанти використання неявно встановлюють вимоги, що визначають, як актори повинні взаємодіяти з системою, щоб мати можливість коректно працювати з сервісами. Прикладами варіантів використання можуть бути такі дії: перевірка стану поточного рахунку клієнта, оформлення замовлення на купівлю товару, відбивання графічної форми на екрані монітора та інші дії.

Актор являє собою будь-яку зовнішню сутність, яка взаємодіє з системою і використовує її функціональні можливості для досягнення певних цілей. При цьому актори служать для позначення узгодженої множини ролей, які можуть грати користувачі в процесі взаємодії з системою, що проектується.

Кожен актор може розглядатися як окрема роль щодо конкретного варіанту використання. Стандартним графічним позначенням актора на діаграмах є фігурка чоловічка, під якою записується ім'я актора.

Імена акторів повинні записуватися великими літерами та дотримуватися рекомендацій використання імен для типів та класів моделі.

Прикладами акторів можуть бути: клієнт банку, банківський службовець, менеджер відділу продажів, адміністратор готелю, мобільний телефон, сервер та інші сутності, що стосуються концептуальної моделі відповідної предметної області.

Оскільки у загальному випадку актор завжди знаходиться поза системою, його внутрішня структура не визначається. Для актора має значення лише те, як він сприймається із боку системи.

Актори взаємодіють із системою у вигляді обміну повідомленнями з варіантами використання. Повідомлення становить собою запит актором певного сервісу системи та отримання цього сервісу.

Ця взаємодія може бути виражена за допомогою асоціацій між окремими акторами та варіантами використання. Окрім цього, з акторами можуть бути зв'язані інтерфейси, які визначають, яким чином інші елементи моделі взаємодіють з цими акторами.

Між елементами діаграми варіантів використання можуть існувати різні відносини, які описують взаємодію екземплярів одних акторів та варіантів

використання з екземплярами інших акторів та варіантів. Один актор може взаємодіяти з кількома варіантами використання. У цьому випадку актор звертається до кількох сервісів даної системи. Водночас один варіант використання може взаємодіяти з декількома акторами, надаючи для них свій сервіс.

Два варіанти використання, визначені в рамках однієї моделі також можуть взаємодіяти один з одним, проте характер цієї взаємодії буде відрізнятися від взаємодії з акторами. Однак в обох випадках способи взаємодії елементів моделі передбачають обмін повідомленнями, які ініціюють реалізацію функціональної поведінки системи, що моделюється [13].

Діаграма варіантів використання, згідно з мобільним додатком, що розробляється, зображена на рис. 3.4.

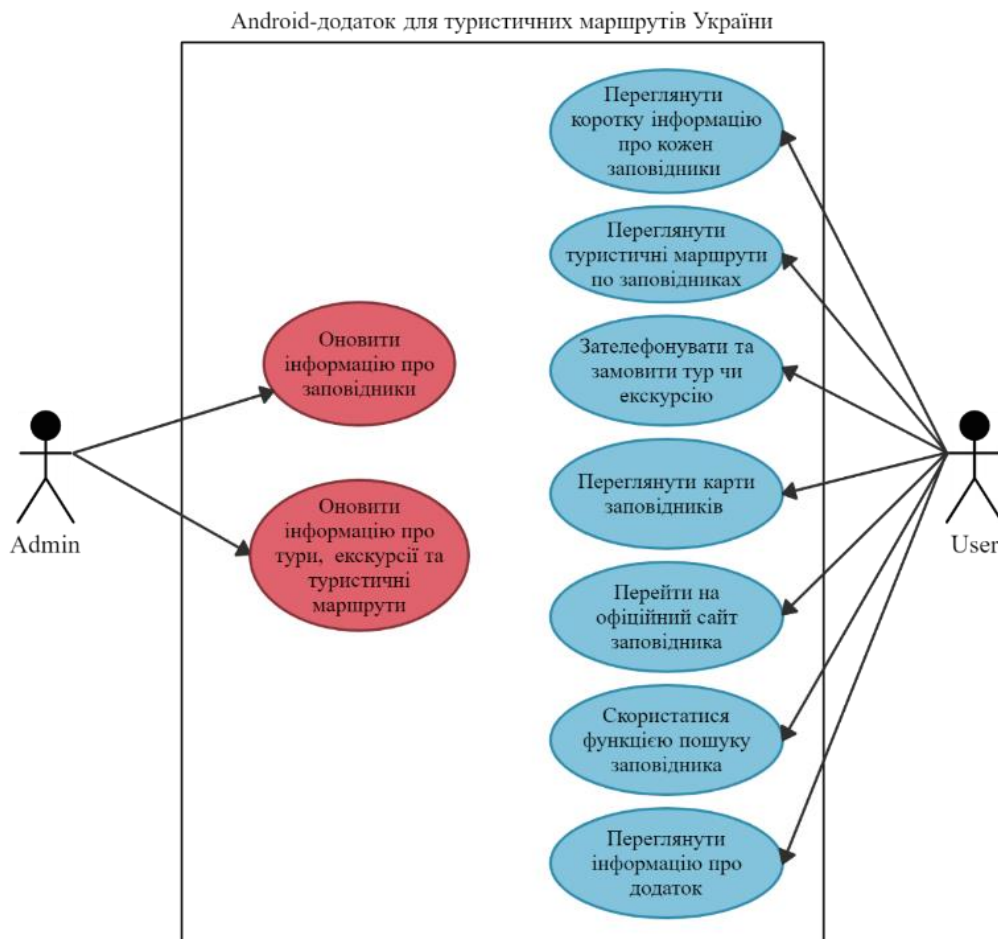


Рисунок 3.4 – Діаграма варіантів використання

### 3.3 Проектування інтерфейсу

Для створення макетів інтерфейсу мобільного додатка було обрано хмарне ПЗ Moqups.

Хмарне ПЗ Moqups призначено для створення прототипів графічних інтерфейсів. У Moqups можна створити інтерактивні прототипи веб-сайтів та додатків як для настільних комп'ютерів і смартфонів, так і для інших пристроїв.

У мобільному додатку буде розроблено 27 сторінок, з яких 24 сторінки будуть мати майже однаковий макет.

Макет головної сторінки зображено на рис. 3.5.

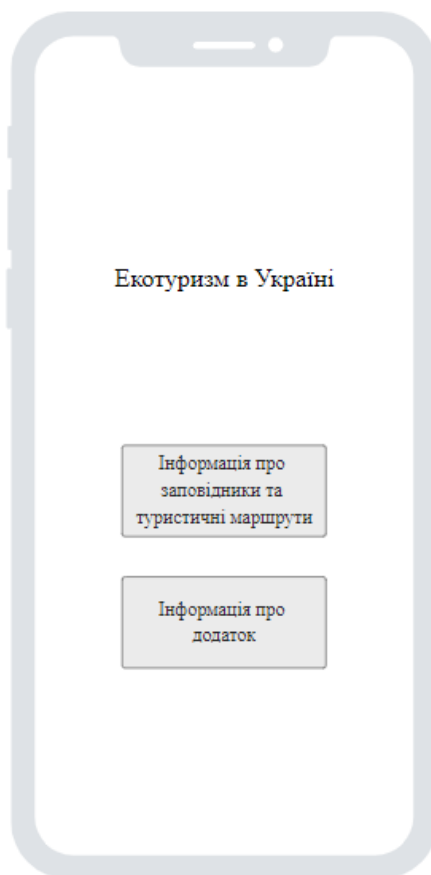


Рисунок 3.5 – Макет головної сторінки додатку

Макет сторінки з інформацією про додаток зображено на рис. 3.6.

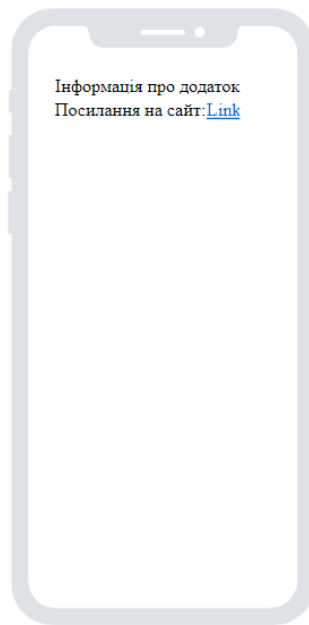


Рисунок 3.6 – Макет сторінки з інформацією про додаток

Макет сторінки вибору заповідника зображено на рис. 3.7.

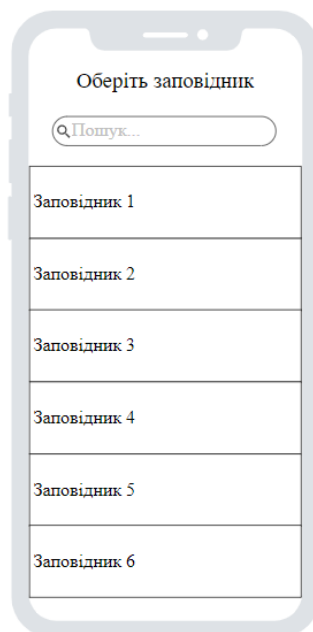


Рисунок 3.7 – Макет сторінки вибору заповідника

Макет для 24 сторінок з інформацією про заповідники, зображено на рис. 3.8.

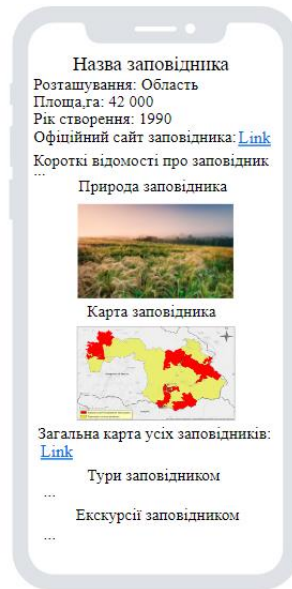


Рисунок 3.8 – Макет для 24 сторінок з інформацією про заповідники

Діаграму взаємозв'язку макетів зображено на рис. 3.9.

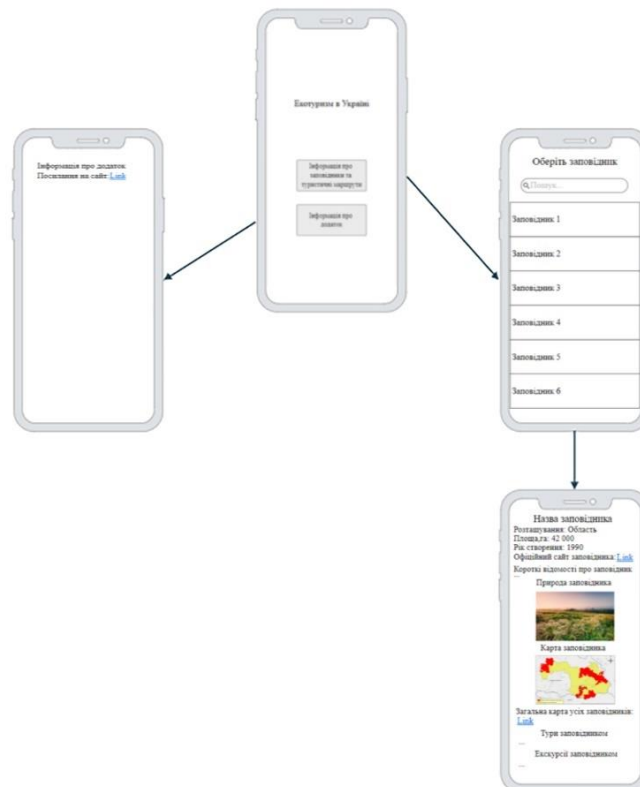


Рисунок 3.9 – Діаграма взаємозв'язку макетів

### 3.4 Установка Android Studio та створення проекту

Для того щоб завантажити Android Studio потрібно перейти на сайт <https://developer.android.com/studio> та натиснути кнопку «Download Android Studio» після чого почнеться завантаження програми.

Після завантаження треба відкрити програму та створити новий проект. Для створення проекту треба вказати його ім'я, ім'я пакета, місце де проект повинен зберігатися, вибрати мову програмування та мінімальну версію SDK, після чого натиснути кнопку «Finish» (рис. 3.10):

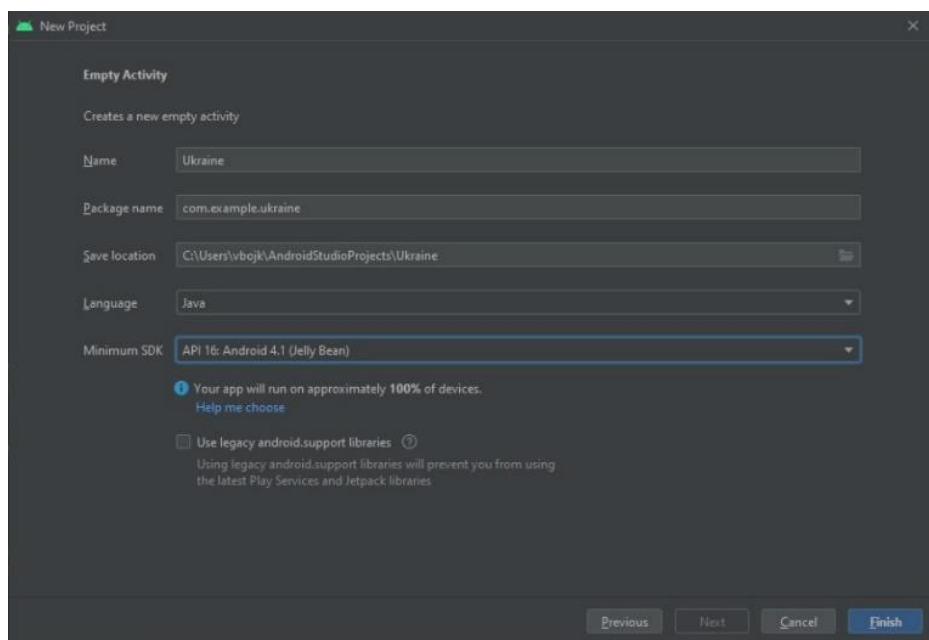


Рисунок 3.10 – Створення проекту

### 3.5 Реалізація функції пошуку

У мобільному додатку була реалізована функція пошуку яка шукає необхідний заповідник по його назві зі списку заповідників.

Програмний код класу з реалізацією цієї функції для списку с заповідниками України наведено у додатку А.



Робота функції пошуку зображена на рис. 3.11.

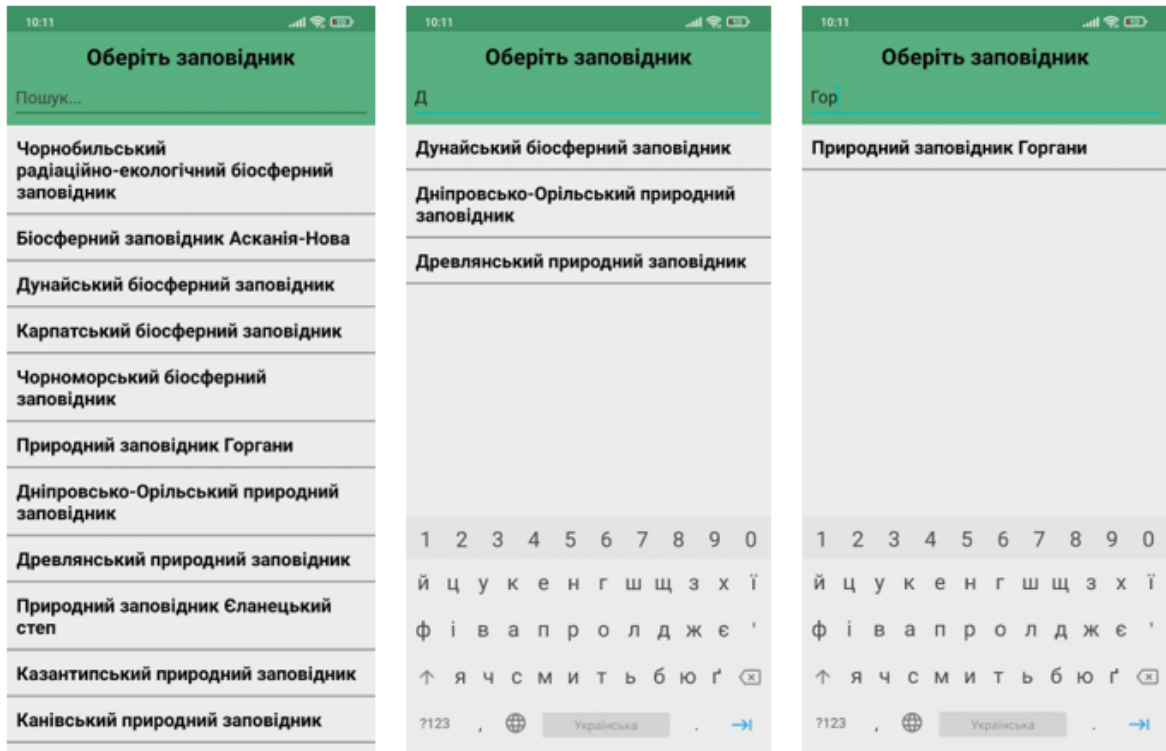


Рисунок 3.11 – Робота функції пошуку

### 3.6 Керівництво користувача

Для запуску мобільного додатка використовується іконка з логотипом (рис. 3.12):

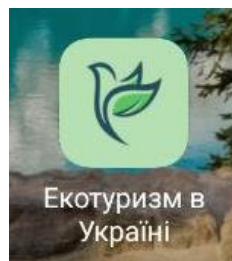


Рисунок 3.12 – Логотип мобільного додатка «Екотуризм в Україні»

Після натиску на логотип відкривається головний екран (рис. 3.13):



Рисунок 3.13 – Головний екран мобільного додатка «Екотуризм в Україні»

На головній сторінці додатка користувач може обрати одну з двох кнопок.

Якщо користувач хоче дізнатися інформацію про додаток він може натиснути кнопку «Інформація про додаток». Натиснувши її відкриється екран з коротким описом додатка і посиланням на сайт веб-системи туристичних маршрутів України (рис. 3.14).

Щоб дізнатися інформацію про заповідники та тури, користувач повинен натиснути кнопку «Інформація про заповідники та туристичні маршрути». Натиснувши її, відкриється екран зі списком усіх заповідників України (рис. 3.15).



Рисунок 3.14 – Екран з інформацією про додаток

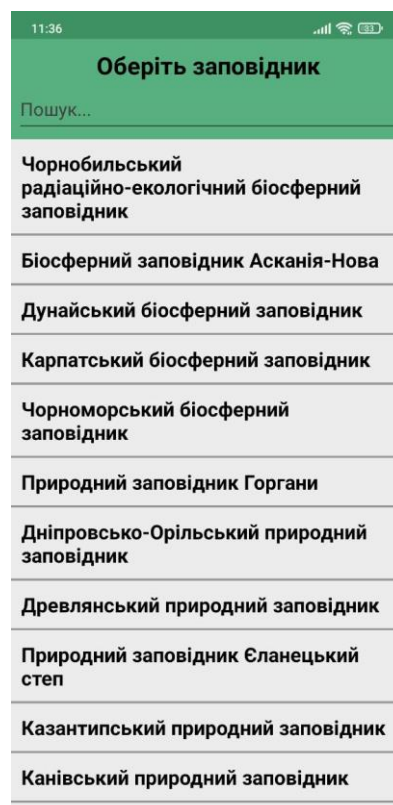


Рисунок 3.15 – Екран вибору заповідника

На екрані з вибором заповідника знаходиться список з назвами 24 заповідників України. Також на цьому екрані присутній рядок пошуку заповідника за його назвою. Робота рядка пошуку представлена на рис. 3.16.

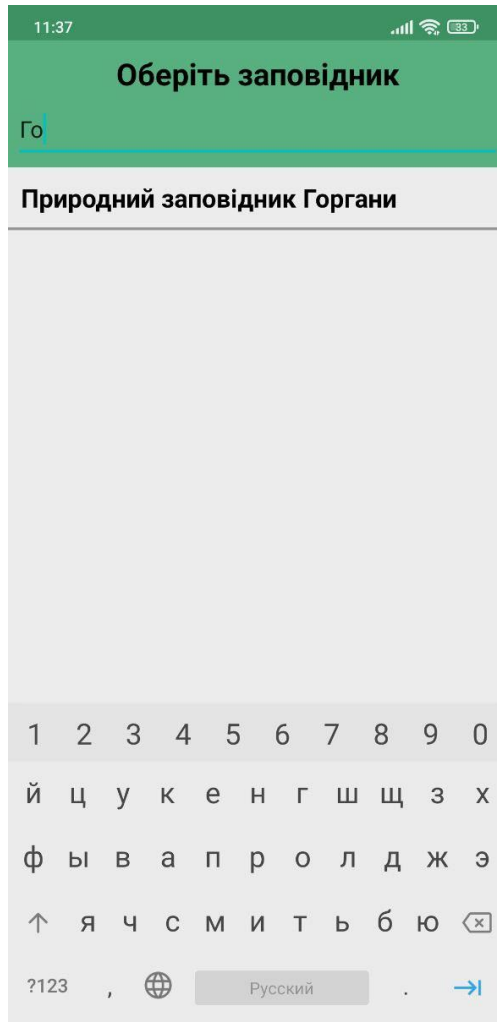


Рисунок 3.16 – Робота рядка пошуку

Щоб дізнатися інформацію про заповідник, що цікавить, треба у списку з заповідниками натиснути на його назву. Після натиску на назву заповідника з'явиться вікно з короткою інформацією про цей заповідник, картинками заповідника, картою заповідника та турами, екскурсіями й туристичними маршрутами по цьому заповіднику. Приклад такого вікна зображено на рис. 3.17.

14:16

### Дніпровсько-Орільський природний заповідник

Розташування: Дніпропетровська область  
Площа, га: 3 766,2  
Рік створення: 1990  
Офіційний сайт заповідника:  
<https://dopzsite.wordpress.com/>


Короткі відомості про заповідник

Природний заповідник «Дніпровсько-Орільський» – природний заповідник в Україні, в межах Петриківського та Дніпровського районів Дніпропетровської області. Розташований у долині Дніпра і плавнів Протовачі. Створений на базі загальнозоологічного та орнітологічного заказників «Таромські плавні» та «Обухівські заплави». У цьому заповіднику охороняються плавневі ліси, гаї, луки, ділянки автохтонної рослинності: ковила, сон-трава, тюльпан, шелога, чорноклен. Ростає 9 видів рослин, занесених у Червону книгу України, і 50 рідкісних для Дніпропетровщини видів. Фауна налічує 31 вид осявців: лось, сарна, олень, свиня дика, заєць сірий, ондатра, сніт уссурійський, бобр, куниця, горностай, борсук. Риб – 50 видів. Фауна птахів заповідника за період досліджень 1991 – 2014 рр. налічує 185 видів птахів, які належать до 6 екологічних груп: водно-болотного, лісового, лучного, синантропного, степового, узлісно-чагарникового.


Природа заповідника



14:17





Карта Дніпровсько-Орільського природного заповідника



Загальна карта усіх заповідників України:  
[https://www.google.com/maps/d/u/0/viewer?mid=12C5mXtX\\_9md\\_\\_\\_O15grWAZnBeN7chsD\\_&ll=48.65170562420363%2C31.981869603114596&z=6](https://www.google.com/maps/d/u/0/viewer?mid=12C5mXtX_9md___O15grWAZnBeN7chsD_&ll=48.65170562420363%2C31.981869603114596&z=6)

14:17



Загальна карта усіх заповідників України:  
[https://www.google.com/maps/d/u/0/viewer?mid=12C5mXtX\\_9md\\_\\_\\_O15grWAZnBeN7chsD\\_&ll=48.65170562420363%2C31.981869603114596&z=6](https://www.google.com/maps/d/u/0/viewer?mid=12C5mXtX_9md___O15grWAZnBeN7chsD_&ll=48.65170562420363%2C31.981869603114596&z=6)

**Експерсії Дніпровсько-Орільським природним заповідником**

Офіційні експерсії мальовничими екостежками від працівників-науковців заповідника.

Загальна протяжність маршрутів – 18 км, у заповіднику 866 видів рослин та 1150 видів тварин. Враховуючи, що самовільне відвідування території заборонено, було створено 2 екостежки, які пристосовані для відвідування натуралістів-аматорів. 1 стежка: «В серце дніпровських плавнів» – 4,8 км, 2 стежка: «По заплавахних луках» – 8 км. Під час міні-походу в учасників буде можливість: Побачити та послухати цікаві факти про типових мешканців лісів, полів Дніпровщини, дізнатися про «іноземних» мешканців (інтродуцентів), тих рослин та тварин, які з'явилися в регіоні штучно, але залишилися та адаптувалися тут, познайомитися з історичним минулим Дніпропетровщини, спостерігати за птахами, тваринами вздовж екостежки.

Замовити експерсії можна за тел. (067)-565-53-15 або (056)-735-12-77

Рисунок 3.17 – Вікно з інформацією про заповідник та експерсії

## ВИСНОВКИ

В ході виконання бакалаврської кваліфікаційної роботи був розроблений мобільний додаток на платформі Android, за допомогою якого можна переглянути інформацію про кожен заповідник України, подивитись карти заповідників України та дізнатися, які є тури, екскурсії та туристичні маршрути по заповідниках.

На етапі аналізу предметної області було виконано:

- характеристика об'єкта розробки;
- опис предметної області;
- аналіз мобільних додатків аналогів.

Серед засобів розробки було обрано:

- ОС Android;
- середовище розробки Android Studio;
- мова програмування Java.

Під час етапу проектування та розробки виконано:

- моделювання процесу розробки;
- проектування діаграми варіантів використання;
- проектування інтерфейсу;
- установка Android Studio та створення проекту;
- керівництво користувача.

При проектуванні мобільного додатку були застосовані всі можливості для того щоб зробити користування додатком максимально зручним та водночас простим та зрозумілим.

Дизайн мобільного додатку створений у мінімалістичному стилі та з інтуїтивно зрозумілим інтерфейсом з урахуванням усіх можливостей та особливостей для швидкого користування.

Додаток має стати у пригоді перш за все людям які люблять подорожі заповідниками, він допоможе швидко і зручно знайти інформацію про заповідник який людина планує відвідати.

В рамках подальшого розвитку у мобільний додаток можна додати карти з конкретними маршрутами та інформацією про дистанцію та приблизний час проходження маршруту. Також крім заповідників у мобільний додаток можна додати інформацію про національні парки, каньйони, озера та інші природні місця України.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Экологический туризм, цели, виды, направления экотуризма. URL: <https://cleanbin.ru/terms/ecological-tourism> (дата звернення 25.05.2022)
2. Зеленый туризм в Украине | ЗРУБ-ЭКОЦЕНТР. URL: <http://www.zrubkarpati.com.ua/zelenyj-turizm-eko-turizm-ekoturizm-v-ukraine/> (дата звернення 25.05.2022)
3. iOS или Android-разработка: что выбрать. URL: <https://blog.skillfactory.ru/ios-vs-android-razrabotka-chto-vybrat/> (дата звернення 26.05.2022)
4. Android или iOS? Плюсы и минусы с точки зрения мобильной разработки. URL: <https://www.it-academy.by/media/stati/android-ili-ios-plyusy-i-minusy/> (дата звернення 27.05.2022)
5. Общая схема работы приложения Android. URL: [https://studbooks.net/1993715/informatika/obschaya\\_shema\\_raboty\\_prilozheniya\\_android](https://studbooks.net/1993715/informatika/obschaya_shema_raboty_prilozheniya_android) (дата звернення 27.05.2022)
6. What is Xamarin? – Xamarin. | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin> (дата звернення 01.06.2022)
7. Установка Eclipse. Подключаем плагин Android Development Tools для разработки Android-приложений. URL: <https://otus.ru/nest/post/1282/> (дата звернення 01.06.2022)
8. Android Studio: что вы можете сделать с помощью этой программы. URL: <https://itznaniya.ru/android-studio-chto-vy-mozhete-sdelat-s-pomoshhyu-etoj-programmy/> (дата звернення 01.06.2022)
9. Kotlin vs Java: что лучше для Android-разработки? URL: <https://lampa-lampa.net/kotlin-vs-java-chto-luchshe-dlya-android-razrabotki/> (дата звернення 01.06.2022)



10. Технология языка XML. Объектно-ориентированные СУБД. Объектно-ориентированное программирование. URL: <http://bourabai.ru/xml/index.htm> (дата звернення 03.06.2022)
11. Gradle. URL: <https://ru.frwiki.wiki/wiki/Gradle> (дата звернення 04.06.2022)
12. Android: Файл манифеста AndroidManifest.xml. URL: <http://developer.alexanderklimov.ru/android/theory/AndroidManifestXML.php> (дата звернення 04.06.2022)
13. Диаграмма вариантов использования (use case diagram). URL: <https://intellect.icu/diagramma-variantov-ispolzovaniya-use-case-diagram-4823> (дата звернення 05.06.2022)

## **ДОДАТКИ**

## ДОДАТОК А

## Лістинг основних класів

```

package com.example.ukraine;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import java.util.ArrayList;
import java.util.HashMap;
public class MainActivity2 extends AppCompatActivity {
    private ListView lv;
    ArrayAdapter<String> adapter;
    EditText inputSearch;
    ArrayList<HashMap<String, String>> words;
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        String words[] = {"Чорнобильський радіаційно-екологічний
біосферний заповідник", "Біосферний заповідник Асканія-Нова",
"Дунайський біосферний заповідник", "Карпатський біосферний
заповідник",
        "Чорноморський біосферний заповідник", "Природ-
ний заповідник Горгани", "Дніпровсько-Орільський природний за-
повідник", "Древлянський природний заповідник", "Природний за-
повідник Єланецький степ",
        "Казантипський природний заповідник", "Канівсь-
кий природний заповідник", "Карадазький природний заповідник",
"Кримський природний заповідник", "Луганський природний за-
повідник",
        "Природний заповідник Медобори", "Природний за-
повідник Михайлівська цілина", "Український степовий природний
заповідник", "Природний заповідник Мис Мартьян",
        "Опукський природний заповідник", "Поліський
природний заповідник", "Природний заповідник Розточчя", "Рівнен-
ський природний заповідник", "Черемський природний заповідник",
"Ялтинський гірсько-лісовий природний за-
повідник"};
        lv = (ListView) findViewById(R.id.list_view);
        adapter = new ArrayAdapter<String>(this, R.lay-
out.list_item, R.id.product_name, words);
        lv.setAdapter(adapter);
        inputSearch = (EditText) findViewById(R.id.inputSearch);
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

```

```

        TextView c = (TextView) view.findViewById(R.id.product_name);
        String reserve = c.getText().toString();
        if( reserve=="Чорнобильський радіаційно-
екологічний біосферний заповідник"){
            position=0;}
        if( reserve=="Біосферний заповідник Асканія-
Нова"){
            position=1;}
        if( reserve=="Дунайський біосферний
заповідник"){
            position=2;}
        if( reserve=="Карпатський біосферний
заповідник"){
            position=3;}
        if( reserve=="Чорноморський біосферний
заповідник"){
            position=4;}
        if( reserve=="Природний заповідник Горгани"){
            position=5;}
        if( reserve=="Дніпровсько-Орільський природний
заповідник"){
            position=6;}
        if( reserve=="Древлянський природний за-
повідник"){
            position=7;}
        if( reserve=="Природний заповідник Єланецький
степ"){
            position=8;}
        if( reserve=="Казантипський природний за-
повідник"){
            position=9;}
        if( reserve=="Канівський природний заповідник"){
            position=10;}
        if( reserve=="Карадазький природний за-
повідник"){
            position=11;}
        if( reserve=="Кримський природний заповідник"){
            position=12;}
        if( reserve=="Луганський природний заповідник"){
            position=13;}
        if( reserve=="Природний заповідник Медобори"){
            position=14;}
        if( reserve=="Природний заповідник Михайлівська
цілина"){
            position=15;}
        if( reserve=="Український степовий природний за-
повідник"){
            position=16;}
        if( reserve=="Природний заповідник Мис
Мартьян"){
            position=17;}
        if( reserve=="Опукський природний заповідник"){
            position=18;}
        if( reserve=="Поліський природний заповідник"){
            position=19;}
        if( reserve=="Природний заповідник Розточчя"){
            position=20;}
        if( reserve=="Рівненський природний за-
повідник"){

```

```

        position=21;}
    if( reserve=="Черемський природний заповідник"){
        position=22;}
    if( reserve=="ЯЛТИНСЬКИЙ гірсько-лісовий природ-
ний заповідник"){
        position=23;}
    switch (position) {
        case 0:
            intent = new Intent(MainActivity2.this,
ReserveActivity.class);
            break;
        case 1:
            intent = new Intent(MainActivity2.this,
Reserve1Activity.class);
            break;
        case 2:
            intent = new Intent(MainActivity2.this,
Reserve2Activity.class);
            break;
        case 3:
            intent = new Intent(MainActivity2.this,
Reserve3Activity.class);
            break;
        case 4:
            intent = new Intent(MainActivity2.this,
Reserve4Activity.class);
            break;
        case 5:
            intent = new Intent(MainActivity2.this,
Reserve5Activity.class);
            break;
        case 6:
            intent = new Intent(MainActivity2.this,
Reserve6Activity.class);
            break;
        case 7:
            intent = new Intent(MainActivity2.this,
Reserve7Activity.class);
            break;
        case 8:
            intent = new Intent(MainActivity2.this,
Reserve8Activity.class);
            break;
        case 9:
            intent = new Intent(MainActivity2.this,
Reserve9Activity.class);
            break;
        case 10:
            intent = new Intent(MainActivity2.this,
Reserve10Activity.class);
            break;
        case 11:
            intent = new Intent(MainActivity2.this,
Reserve11Activity.class);
            break;
        case 12:
            intent = new Intent(MainActivity2.this,
Reserve12Activity.class);
            break;
        case 13:

```



```

        public void afterTextChanged(Editable arg0) {
        }
    });
}}

package com.example.ukraine;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import java.util.ArrayList;
import java.util.HashMap;
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    Button button;
    Button btnActTwo;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnActTwo = (Button) findViewById(R.id.btnActTwo);
        btnActTwo.setOnClickListener(this);
        button = (Button) findViewById(R.id.button);
        button.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btnActTwo:
                Intent intent = new Intent(this, Activity-
Two.class);
                startActivity(intent);
                break;
            case R.id.button:
                Intent intent2 = new Intent(this, MainActiv-
ity2.class);
                startActivity(intent2);
                break;
        }
    }
}

```