Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних систем управління
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка системи автоматизованого вибору туристичних об'єктів»

Виконала: магістр 2 курсу, групи 2АКІТР-23м
спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(шифр і назва напряму підготовки, спеціальності)

Мирослава ТЕСЛЮК
(ім'я ПРІЗВИЩЕ)

Керівник: к.т.н., професор каф. КСУ
Микола БИКОВ
(ім'я ПРІЗВИЩЕ)
«4» 12 2024 р.

Опонент: к.т.н., доцент каф. АІІТ
Юрій ІВАНОВ
(ім'я ПРІЗВИЩЕ)
«4» 12 2024 р.

Допущено до захисту
Завідувач кафедри КСУ
Вячеслав КОВТУН
(прізвище та ініціали)
«4» 12 2024 року

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти другий (магістерський)
Галузь знань – 17 Електроніка, автоматизація та електронні комунікації
Спеціальність - 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітньо-професійна програма - Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
_____ Вячеслав КОВТУН
«14» жовтня 2024 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Теслюк Мирославі Вячеславівні
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка системи автоматизованого вибору туристичних об'єктів»
керівник роботи Биков Микола Максимович, к. т. н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ВНТУ від "17" вересня 2024 року № 310

2. Строк подання студентом роботи "1" грудня 2024 року

3. Вихідні дані до роботи: підтримка Google Maps та Google Api; розроблюваний додаток - кросплатформенний; авторизацію користувачів не використовувати; використання карт та GPS для навігації; мова графічного інтерфейсу – англійська; використання вбудованої системи коментарів; зручне відображення реклами та інформації про об'єкти.

4. Зміст текстової частини: Вступ; Аналіз стану проблеми автоматизованого вибору туристичних об'єктів; Багатоваріантний аналіз способів та технологій для побудови системи автоматизованого вибору туристичних об'єктів на базі мобільного телефону; Розробка програмного забезпечення системи; Тестування програмного забезпечення; Економічний розділ.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) UML-діаграма варіантів використання – А4; Схема алгоритму роботи системи; ER-модель системи – А4; UML-діаграми класів – 3*А4; Основні екранні форми програмного забезпечення – 3*А4.

6. Консультанти розділів роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 4	Кавецький В.В. – доцент кафедри економіки підприємства і виробничого менеджменту	20.11.2024р	22.11.2024р

7. Дата видачі завдання "_14__" __жовтня___2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Багатоваріантний аналіз способів побудови системи автоматизованого вибору туристичних об'єктів, постановка задач.	07. 09. 2024 р.	14.09.2024 р.	
2	Розробка технічного завдання	15.09.2024 р.	16.09.2024 р.	
3	Розробка архітектури системи	17.09.2024 р.	24.09.2024 р.	
4	Вибір технологій розробки системи	25.09.2024 р.	02.11.2024 р.	
5	Розробка програмного забезпечення системи	03.11.2024 р.	15.11.2024 р.	
6	Тестування програмного забезпечення	16.11.2024 р.	19.11.2024 р.	
7	Оформлення пояснювальної записки	20.11.2024 р.	24.11.2024 р.	
8	Апробація і публікації результатів роботи	19.11.2024 р.	30.11.2024 р.	
9	Графічні матеріали Розробка UML-діаграм системи Розробка ER-моделі бази даних системи Розробка демонстраційних плакатів	25.11.2024 р. 28.11.2024 р. 30.11.2024 р.	27.11.2024 р. 29.11.2024 р. 30.11.2024 р.	
10	Попередній захист	01 грудня .2024 р.		
11	Остаточний захист	09 грудня 2024 р.		

Студентка _____ Мирослава ТЕСЛЮК

Керівник роботи _____ Микола БИКОВ

АНОТАЦІЯ

УДК 004.652.4

Теслюк М. В. Розробка системи автоматизованого вибору туристичних об’єктів. Магістерська кваліфікаційна робота зі спеціальності 174 – автоматизація, комп’ютерно-інтегровані технології та робототехніка, освітня програма – інтелектуальні комп’ютерні системи. Вінниця: ВНТУ, 2024. 136 с.

На укр. мові. Бібліогр.: 51 назв; рис.: 29; табл. 13.

Метою магістерської кваліфікаційної роботи є розробка системи автоматизованого вибору туристичних об’єктів, реалізована як програмний додаток до мобільного телефону. Система надає зручний сервіс гостям і жителям міста, забезпечуючи узагальнену інформацію про туристичні об’єкти і відгуки про них інших користувачів, показує маршрути для відвідування наявних об’єктів та пропонує перелік місць, відсортованих за категоріями.

Проведений огляд технологій розробки дозволив обрати технологію Xamarin, яка підтримується становленими на смартфонах OS Android і iOS.

У розділі програмного забезпечення розроблена програма пошуку оптимальних туристичних маршрутів на базі мурашиного алгоритму з використанням Google Maps. Відображення інформації про об’єкти запрограмовано мовою SQL, яка використовує реляційну модель для обробки даних. Машинний експеримент підтвердив адекватність розробленого програмного забезпечення.

Розроблені алгоритми і програмне забезпечення дозволять підвищити відвідуваність туристичних об’єктів та значно спростити їх вибір.

Графічна частина містить 12 плакатів і креслень з ілюстрацією результатів роботи.

Ключові слова: системи автоматизованого вибору, туристичні об’єкти, мурашиний алгоритм, мобільний додаток, програмне забезпечення.

ABSTRACT

UDC 004.652.4

Teslyuk M. V. Development of a system of automated selection of tourist objects. Master's qualification thesis on specialty 174 - automation, computer-integrated technologies and robotics, educational program - intelligent computer systems. Vinnytsia: VNTU, 2024. 136 p.

In Ukrainian language. Bibliography: 51 titles; Fig.: 29; table 13.

The goal of the master's qualification work is to develop a system for automated selection of tourist attractions, implemented as a software application for a mobile phone. The system provides a convenient service to guests and residents of the city, providing generalized information about tourist facilities and reviews of other users, shows routes for visiting existing facilities and offers a list of places sorted by category.

The conducted review of development technologies made it possible to choose Xamarin technology, which is supported by Android and iOS smartphones.

In the software section, a program for finding optimal tourist routes based on the ant algorithm using Google Maps has been developed. The display of information about objects is programmed in the SQL language, which uses a relational model for data processing. The machine experiment confirmed the adequacy of the developed software.

The developed algorithms and software will allow to increase the attendance of tourist objects and significantly simplify their selection.

The graphic part contains 12 posters and drawings illustrating the results of the work.

Keywords: automated selection systems, tourist facilities, ant algorithm, mobile application, software.

ЗМІСТ

ВСТУП……..……...……4

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО ВИБОРУ ТУРИСТИЧНИХ ОБ'ЄКТІВ …….……..7

1.1 Суть проблеми, яка виникла на даному етапі……...7

1.2 Класифікація видів туризму та типів туристичних маршрутів…...12

1.3 Аналіз існуючих систем вибору туристичних маршрутів ……..15

1.4 Стан вирішення проблеми ……....25

1.5 Шляхи вирішення та вибір оптимального варіанту…….26

1.6 Формулювання вимог до розроблюваної системи ……..28

2 ВИБІР ТЕХНОЛОГІЇ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИБОРУ ТУРИСТИЧНИХ ОБ'ЄКТІВ…….……...31

2.1 Огляд технологій, що використовуються для розробки……31

2.2 Аналіз алгоритмів вибору оптимальних маршрутів……..……38

2.3 Розробка алгоритму пошуку оптимального маршруту……49

2.4 Висновки до розділу……...56

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ..……57

3.1 Вибір програмної платформи ..……..57

3.2 Розробка UML-діаграми використання…….…….……59

3.3 Розробка схеми програми роботи системи ……... 61

3.4 Розробка діаграми класів системи …….…….63

3.5 Розробка ER-моделі ……..67

3.6 Тестування програмного забезпечення……...69

3.7 Висновки до розділу ……73

4 ЕКОНОМІЧНА ЧАСТИНА…….. 74

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки ……74

4.2 Визначення рівня конкурентоспроможності розробки ……...78

4.3 Розрахунок витрат на проведення науково-дослідної роботи….81

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором……... 95

4.5 Висновки до розділу…….99

ВИСНОВКИ……100

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……..101

ДОДАТКИ

ДОДАТОК А (обов’язковий). Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень……..107

ДОДАТОК Б (обов’язковий). Технічне завдання ……108

ДОДАТОК В (довідковий). Лістинги програм ……..…...……112

ДОДАТОК Г (обов’язковий) Ілюстративна частина.…….……126

ВСТУП

Актуальність теми. Проблема розробки автоматизованої системи пошуку туристичних маршрутів, яка вирішується в даній магістерській кваліфікаційній роботі, є на сьогодні нагальною з багатьох причин, ключовими з яких є;

- зростання туристичної галузі;
- персоналізація туристичного досвіду;
- велика кількість інформації, яку потрібно враховувати для вибору;
- інтеграція сучасних технологій;
- пандемія та нові вимоги до безпеки;
- можливість розширення для інших застосувань;
- економія часу та зусиль для планування подорожей.

Коротко опишемо сутність названих причин.

Туризм продовжує активно розвиватися, стаючи доступнішим для дедалі більшої кількості людей завдяки покращенню транспортної інфраструктури, зростанню кількості авіарейсів, здешевленню подорожей, а також збільшенню пропозицій у галузі гостинності. Зростає попит на швидкий і зручний пошук цікавих маршрутів та варіантів відпочинку, які відповідали б індивідуальним запитам туристів.

Сучасні туристи все більше цінують індивідуальний підхід. Вони прагнуть знаходити маршрути, які відповідають їхнім інтересам, бюджету, часом перебування та рівнем складності. Автоматизовані системи здатні враховувати ці параметри та надавати рекомендації на основі індивідуальних уподобань та попереднього досвіду туриста, що робить їх незамінним інструментом.

З розвитком Інтернету та інформаційних технологій доступ до туристичних даних збільшився в рази. Однак це породило проблему: туристам стає все важче знаходити релевантну інформацію серед величезної

кількості пропозицій. Автоматизовані системи допомагають обробляти та структурувати ці дані, роблячи пошук маршрутів ефективнішим.

Системи пошуку маршрутів можуть використовувати сучасні технології, такі як штучний інтелект (AI) та машинне навчання (ML), для створення розумних алгоритмів, які навчаються на основі даних про поведінку користувачів. Також вкрай актуальною є інтеграція з картографічними сервісами, реальними відгуками мандрівників, погодними умовами, даними про завантаженість маршрутів та іншими чинниками.

Автоматизовані системи можуть допомагати туристам в реальному часі перевіряти умови подорожі, рекомендації щодо здоров'я, обмеження та інші важливі фактори.

Автоматизовані системи пошуку маршрутів можуть бути адаптовані для різних типів подорожей: міських екскурсій, активного відпочинку (гірські маршрути, піші походи), культурних подорожей чи пляжного відпочинку. Це робить їх універсальним інструментом, здатним залучати широкі аудиторії з різними інтересами.

Завдяки автоматизованим системам туристи можуть скоротити час на планування подорожі, отримуючи готові маршрути, що відповідають їхнім критеріям.

Тому розробка системи, що дозволяє туристам здійснити найкращий вибір туристичного об’єкту і отримати зручний сервіс для планування своїх подорожей, опираючись не тільки на візуальну та рекламну інформацію, а й на зважені вибори інших людей, є на сьогодні **актуальною проблемою.**

Метою магістерської кваліфікаційної роботи є підвищення ефективності процесу вибору та відвідання туристичних об’єктів для туристів та жителів міста.

Для того, щоб досягнути вказаної вище мети, потрібно розв’язати низку таких задач:

- Провести варіантний аналіз існуючих на сьогодні систем даного типу і обрати оптимальний підхід до її побудови;

- Проаналізувати альтернативні технології створення програмного забезпечення для реалізації вибраного підходу;
- Розробити програмне забезпечення системи;
- Розробити інтерфейс для взаємодії користувача з програмним забезпеченням.

Об’єктами дослідження є процес вибору туристичного місця користувачем мобільного додатку.

Предметом дослідження є методи, алгоритми та технології програмування прийняття рішень на основі візуальної та рекламної інформації, а також вибору інших людей.

Методи дослідження базуються на теорії прийняття рішень, інтелектуальних технологіях, теорії ймовірності і статистики.

Наукова новизна одержаних результатів.

Удосконалено метод пошуку туристичних місць і маршрутів шляхом використання мурашиного алгоритму.

Практична цінність. Розроблена система, алгоритм і програмне забезпечення дозволяють підвищити відвідування певних об’єктів та заощадити час та кошти туристів.

Особистий внесок. Здобувачем проведено дослідження сфери туризму, проаналізовано відвідування та віддання переваг туристичним об’єктам і розроблено систему автоматизованого вибору.

Впровадження. Результати магістерської кваліфікаційної роботи використовуються в навчальному процесі на кафедрі комп’ютерних систем управління, а також у розвитку мобільних технологій.

Апробація результатів роботи. Основні положення й результати досліджень доповідалися й обговорювалися на науково-технічній конференції молодих науковців ВНТУ «Молодь в науці» (МН-2025).

Публікації. За результатами конференції опубліковані тези: Теслюк М.В., Биков М.М. “Розробка системи автоматизованого вибору туристичних об’єктів”: https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/22752

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО ВИБОРУ ТУРИСТИЧНИХ ОБ'ЄКТІВ

1.1 Суть проблеми, яка виникла на даному етапі

1.1.1 Засади розвитку туризму в Україні

З кожним роком туризм у світі [1,2,3] стрімко розвивається. В Україні туризм теж має великий потенціал і може стати одним із ключових секторів економіки та важливим фактором міжнародної привабливості країни. Розташована у самому серці Європи, Україна вирізняється багатим культурним і природним різноманіттям: стародавні міста, унікальні природні ландшафти, мальовничі Карпати і Чорне море, пам'ятки історії та архітектури – усе це робить її цікавою для туристів з усього світу. Попри значний потенціал, розвиток туризму в Україні стикається із серйозними викликами, однак існують і численні можливості для його вдосконалення та розширення.

З точки зору привабливості туризму Україна має унікальні переваги порівняно з іншими країнами. Вона пропонує туристам різноманітні види відпочинку та культурного дозвілля.

Культурний туризм, особливо популярний серед іноземців, дозволяє познайомитися з історією та традиціями країни. Київ, Львів, Одеса, Харків та інші великі міста України приваблюють своїми пам'ятками архітектури, музеями, фестивалями, які відображають багатовікову історію країни. Наприклад, Києво-Печерська лавра, Софійський собор у Києві, а також старовинні замки у Кам'янці-Подільському, Луцьку, Мукачеві – це лише деякі з пам'яток, які привертають увагу мільйонів туристів.

Природний туризм також є перспективним напрямком. Гори Карпати – ідеальне місце для любителів гірського туризму, лижного спорту та еко-туризму, а узбережжя Чорного та Азовського морів – популярні курортні зони. Заповідні зони, такі як Асканія-Нова та Чорнобильська зона

відчуження, також привертають увагу екотуристів і наукових туристів, які зацікавлені у унікальних ландшафтах та екосистемах.

Екотуризм та сталий туризм – перспективні напрями, які враховують потребу збереження навколишнього середовища. Україна має можливість стати провідною країною у Східній Європі з екотуризму, розвиваючи такі напрямки, як відвідування природних парків, екскурсії у села, місцеві ремесла та фермерські господарства.

1.1.2 Основні проблеми розвитку туризму в країні

Попри вказані вище численні переваги, розвиток туризму в Україні гальмується через низку проблем. Однією з головних перешкод є недосконала туристична інфраструктура. Брак сучасних готелів, неякісний стан доріг, обмеженість інформації про туристичні об’єкти англійською мовою – усе це ускладнює перебування в Україні іноземних туристів.

Крім того, важливими є питання безпеки та політичної стабільності. Збройний конфлікт на сході України негативно позначається на іміджі країни як безпечного туристичного напряму. Цей фактор змушує багатьох потенційних туристів уникати відвідування України, особливо з огляду на заголовки міжнародних новин, де часто фігурують питання безпеки.

Значною проблемою є також недостатній розвиток маркетингу та брендування країни на міжнародній арені. На відміну від таких популярних напрямків, як Італія, Франція чи Греція, Україна досі залишається маловідомою для багатьох мандрівників. Недостатнє фінансування туристичної галузі та відсутність ефективної державної політики у цій сфері також ускладнюють промоцію країни.

В таблиці 1.1 наведена динаміка зміни розвитку туристичних потоків за 6 років (2017–2022 роки), яка підтверджує існування вказаних проблем. згрупована за видами туризму. Дані наведені у таблиці, узагальнено на основі офіційної статистики Держстату [4] та регіональних управлінь статистики.

Таблиця 1.1 – Динаміка туристичних потоків в Україні за 6 років

Рік	В'їзний туризм (іноземці)	Виїзний туризм (громадяни України)	Внутрішній туризм
2017	16 осіб	43,626 осіб	8,287 осіб
2018	273 осіб	53,633 осіб	8,326 осіб
2019	223 осіб	64,911 осіб	11,077 осіб
2020	4 осіб	36,182 осіб	7,803 осіб
2021	~200 осіб (орієнтовно)	~50,000 осіб (орієнтовно)	~15,000 осіб
2022	Дані недоступні через війну	Значне скорочення	Значне скорочення

В таблиці 1.1 відображені дані по таких видах туризму:

- В'їзний туризм**:** кількість іноземних туристів, обслугованих українськими туроператорами та турагентами.

- Виїзний туризм**:** кількість громадян України, які виїжджали за кордон з туристичною метою через українських туроператорів та турагентів.

- Внутрішній туризм**:** кількість громадян України, які подорожували всередині країни за сприянням українських туроператорів та турагентів.

Ключові тенденції, які можна простежити з даної таблиці, можна коментувати таким чином.

1) В'їзний туризм залишається найменш розвинутим сегментом, із значними втратами у 2020–2022 роках через COVID-19 та війну.
2) По виїзному туризму спостерігалося зростання до 2019 року, після чого почалося скорочення через пандемію.
3) Внутрішній туризм у 2017–2019 роках мав стабільний ріст, але пандемія та війна суттєво вплинули на цей сегмент.

Описані тенденції прослідковуються на графічній діаграмі, представленій на рис. 1.1.

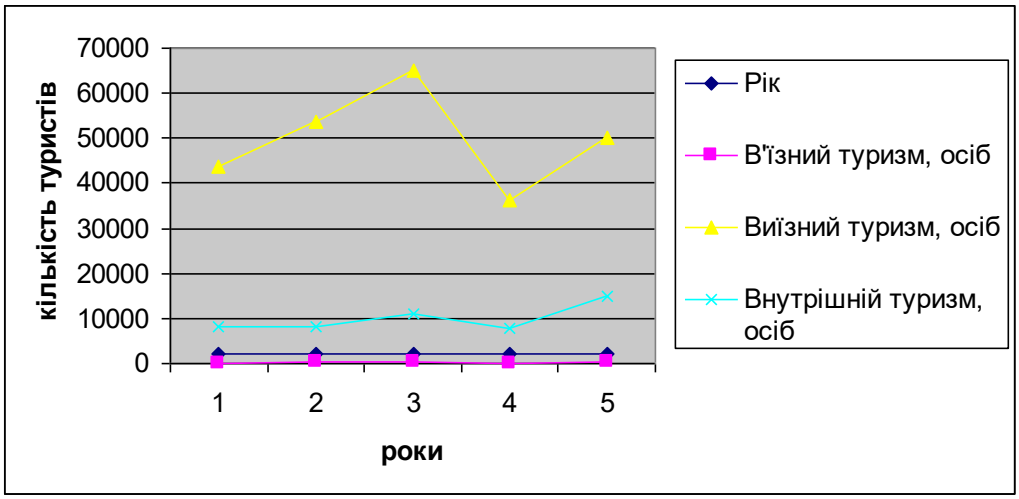


Рисунок 1.1 – Динаміка туристичних потоків в Україні за 2017-2021 рік

1.1.3 Шляхи ров'язання проблем в туристичній галузі

Розвиток туризму в Україні – це складний процес, що потребує цілеспрямованої роботи, інвестицій та співпраці між державою і приватним сектором. Подолання наявних перешкод та використання унікальних можливостей дозволить Україні посісти гідне місце на міжнародній туристичній карті. Туризм може стати не тільки джерелом доходу для держави, а й ефективним засобом зміцнення позитивного іміджу країни у світі.

Впровадження сучасних стратегій, залучення підтримки держави та використання інформаційних технологій дозволить вирішити низку вказаних вище проблем, що гальмують розвиток туризму в Україні і надати цій галузі великі перспективи.

Одним із ключових напрямків є, звичайно, інвестування в інфраструктуру, зокрема в будівництво та ремонт доріг, аеропортів, вокзалів, сучасних готелів та створення інформаційних центрів для туристів. Це дозволить зробити подорож Україною комфортнішою для туристів, що приїжджають із-за кордону.

Маркетинг країни як туристичного напрямку є ще одним важливим напрямком. Поширення позитивного іміджу країни за допомогою

міжнародних виставок, рекламних кампаній, співпраця з популярними туристичними блогерами та впливовими виданнями – усе це може допомогти залучити нових відвідувачів.

Сприяти економічному розвитку галузі внутрішнього туризму дозволяє популяризація історичних і природних місць, організацію тематичних фестивалів та заходів також. Така популяризація можлива за рахунок створення автоматизованої системи інформування туристів і вибору для них оптимальних для даної місцевості туристичних маршрутів, яка є предметом розробки в даній магістерській дисертації.

Розглянемо актуальність цього напрямку.

Кількість відвідувачів стає все більша, але це не означає, що для туристів просто зробити вибір. Проблема вибору туристичних маршрутів для відпочинку є однією з розповсюджених в житті людей. Вона популяризація постає насамперед через те, що туристи, відвідуючи країну, не знають якому місцю віддати перевагу. Щоб зробити потрібний вибір вони втрачають чимало зусиль. Втрачаючи час, гроші, не може йти й мови про хороший відпочинок. Буває часто таке, що приїхавши в країну, туристові потрібно розписати маршрут, який буде вигідним для нього, зважаючи на певні чинники, такі як час, гроші, і т.д. Оскільки в кожної людини свої переваги, об’єкти для відвідування будуть відвідуватись тільки тими туристами яким це дійсно цікаво, задля збагачення своїх знань, духовності чи просто для відпочинку. Не секрет, що засоби масової інформації дійсно сприяють вирішенню цієї проблеми, але цього не достатньо. Не кожна людина подивившись рекламу, захоче відвідати певний об’єкт, слід задуматись, які в неї для цього причини? Не може турист відвідувати туристичні об’єкти тільки для того, щоб задовольнити свою цікавість. До того ж, не у всіх містах, країнах є путівники, які допомагають освоїтись з місцевістю, та прийняти правильний вибір. Тобто проблема дійсно існує [5], і дуже багато людей подорожуючи зіштовхнулись з нею. Наприклад, візьмемо місто, в якому є багато об’єктів для відвідування і людина яка живе там кілька років

не відвідала їх. Хто допоможе їй здійснити хороший вибір туристичного об’єкту, щоб задовольнити її потреби і зробити мінімальним шанс її розчарування, потрачені зусилля та інше. Для того щоб не втрачати зусиль, дивлячись реклами, спілкуючись з іншими людьми в очікуванні поради, потрібно ефективно вирішити дану проблеми шляхом розробки автоматизованої системи пошуку туристичних маршрутів [6,7,8].

1.2 Класифікація видів туризму та типів туристичних маршрутів

Туризм можна класифікувати [1,9,10] за різними параметрами. На рисунку 1.2 зображено схему класифікації туризму:

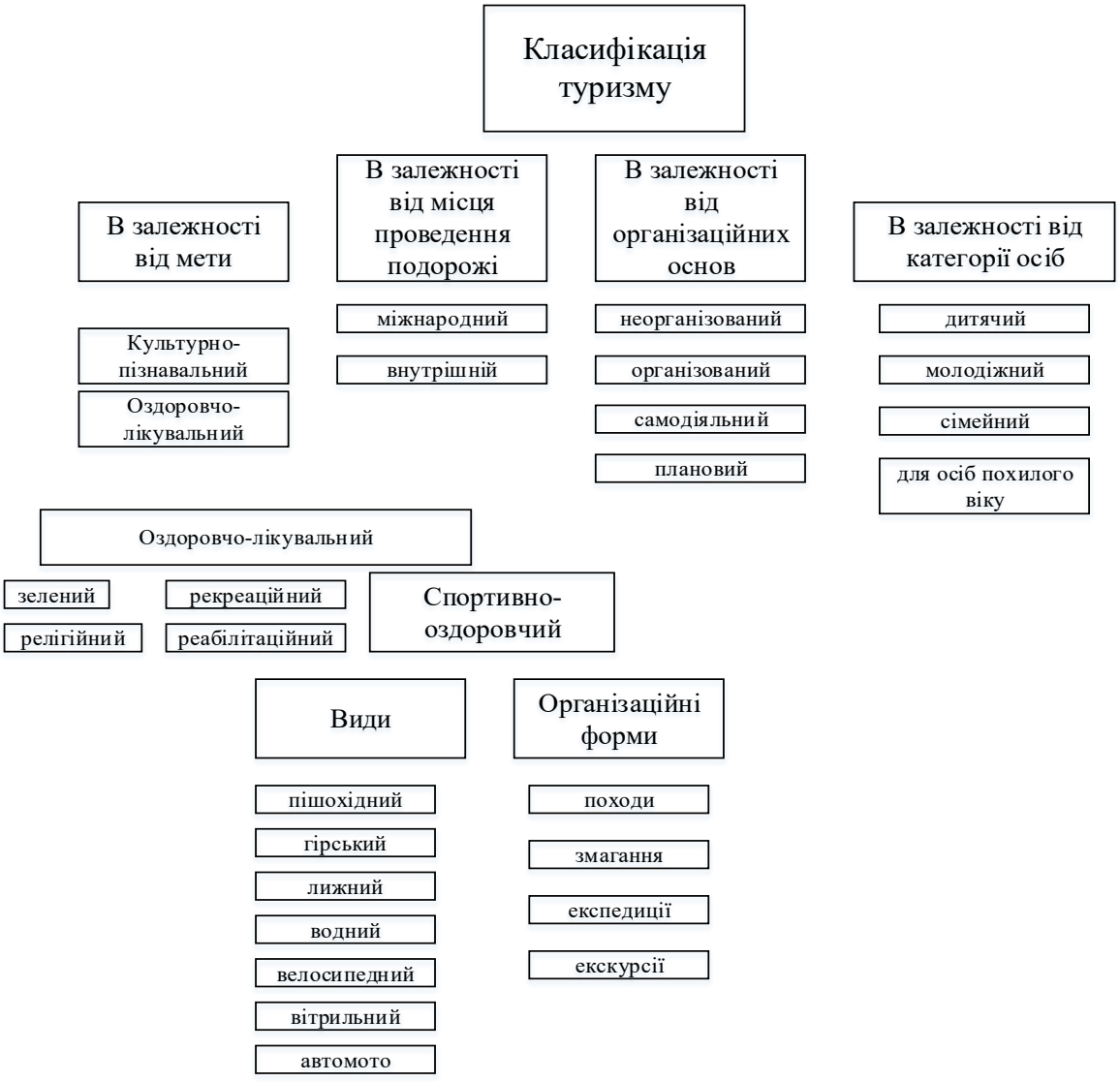


Рисунок 1.2 – Схема класифікація туризму

Отже, як можна побачити на рисунку 1.2, туризм класифікують за певними ознаками, а саме:

- в залежності від мети;
- в залежності від місця проведення подорожі;
- в залежності від організаційних основ;
- в залежності від категорії осіб.

Туристичні об’єкти – це місця, які мають комфортні умови для відпочинку, історичну цінність, пам’ятки культури, мистецтва, які мають приємні параметри чи якості для людини-відвідувача та ін.. Туристичні об’єкти можна класифікувати по різному, в залежності від їх призначення, інформативних якостей, цінностей та впливу на відвідувачів [11].

Туристичні маршрути – це шляхи, раніше продумані туристами чи спеціальними агентствами, для відвідування декількох туристичних об’єктів, які входять до нього. Класифікація туристичних маршрутів зображена на рисунку 1.3:

Рисунок 1.3 – Класифікація туристичних маршрутів

Перед початком довгої подорожі туристи зазвичай складають туристичні маршрути. Вони класифікуються за певними ознаками:

- мета подорожі;
- вид пересування;
- сезонність;
- тривалість;
- контингент;
- форма.

Одним із основних джерел інформації туриста для пересування є туристичні карти [11,12,13,14]. Вони поділяються на деякі види, зображені на рисунку 1.4:

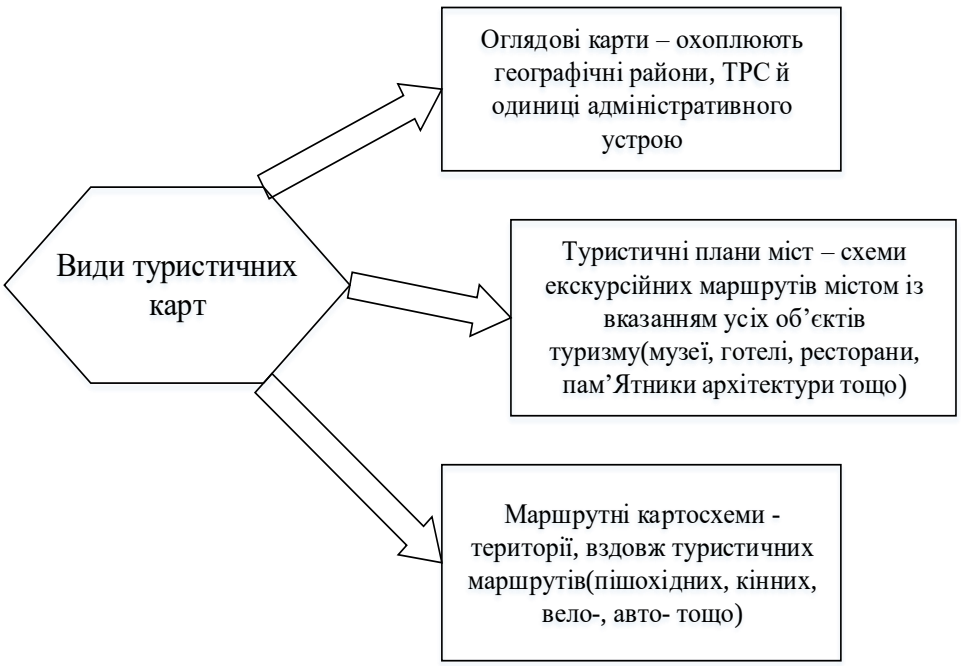


Рисунок 1.4 – Види туристичних карт

Оскільки актуальність туризму зростає з кожним днем все більше і більше, потоки туристів все зростають і зростають. Люди полюбляють подорожувати. Це завжди захоплююче і цікаво. Крім того, що можна помилуватись туристичним об'єктом, люди отримують багато корисної інформації для себе. Це сприяє стрімкому інтелектуальному розвитку. Навіть

заради того щоб дізнатись, чи побачити щось нове, відкрити для себе нові місця, побувати там де ніколи не бував.

1.3 Аналіз існуючих систем вибору туристичних маршрутів

Проведемо аналіз існуючих систем систем автоматизованого вибору туристичних маршрутів [15,16,17,18] на базі мобільних додатків. Існують дуже багато мобільних додатків, які мають функції GPS навігації, карти, об’єкти на мапах, і т.д. Якщо порівнювати їх з створеною системою, то це забере багато часу і зусиль і не принесе ніякого результату, через те, що зазвичай такі програми використовуються або для навігації, або як інформативні. Вони не основані ні на яких алгоритмах, та частіше за все несуть тільки інформативні дані. Прирівняємо їх, подивимось на їх функції, застосування, та популярність використання серед користувачів.

Розглянемо переваги і недоліки основних відомих на сьогодні систем автоматизованого пошуку туристичних маршрутів:

- Maps.me;
- Here WeGo;
- OsmAnd;
- TownPass;
- Cape TownPass;
- Google Trips;
- TripAdvisor;
- Airbnb Experiences.

Maps.me – це мобільний додаток, який дозволяє використовувати автономні карти, які можна використовувати онлайн з покроковою навігацією. Оцінка на Play Market 4.5(досить висока). Приклад такої програми показаний на рисунку 1.5:

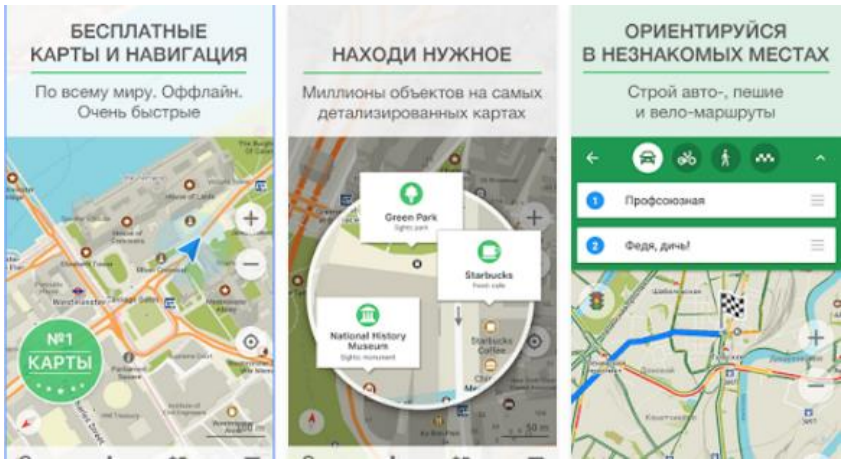


Рисунок 1.5 – Приклад програми Maps.me

Програми **Here WeGo** та **OsmAnd** є аналогами програми Maps.me. Програми, які використовуються в подорожах різного типу, допомагають зорієнтуватись в незнайомих місцях, отримати значну та достатню кількість інформації про ці об’єкти та дістатися до них. Використовують оффлайн карти, можливість записувати свої маршрути, та використання різного типу карт. Оцінки на Play Market – 4.2.

Програма **Cape TownPass** більше зосередженна на туристичних об’єктах, вона надає інформативні дані з про об’єкти, без використання мапи. Пропонує туристам різні місця для відвідання, поділяючи об’єкти на категорії для зручності перегляду. Дозволяє доступ [19] до всіх об’єктів безкоштовно, але основною функцією програми є дисконтні знижки на перебування в цих місця, наприклад ресторани, кафе, та інші об’єкти. Оцінка на Play Market – 4.0. Приклад програми зображений на рисунку 1.6:

Програма **TownPass** – українська програма, є аналогом програми Cape TownPass. Все те ж саме, інформаційні дані про об’єкти, які можна відвідати, поділ об’єктів на категорії для зручності, дисконтні знижки, показ подій, які мають відбутись в даних місцях. Перевагою цієї програми є те, що в ній використовується карта, тобто можна побачити всі об’єкти на мапі. Це досить зручно. Але недоліком є те, що вона використовується тільки в Україні, адже база даних вміщує в собі тільки декілька українських міст. За

кордоном ця програма не має ніякого застосування. Але туристи, які прижають в Україну, можуть сміливо користуватись.

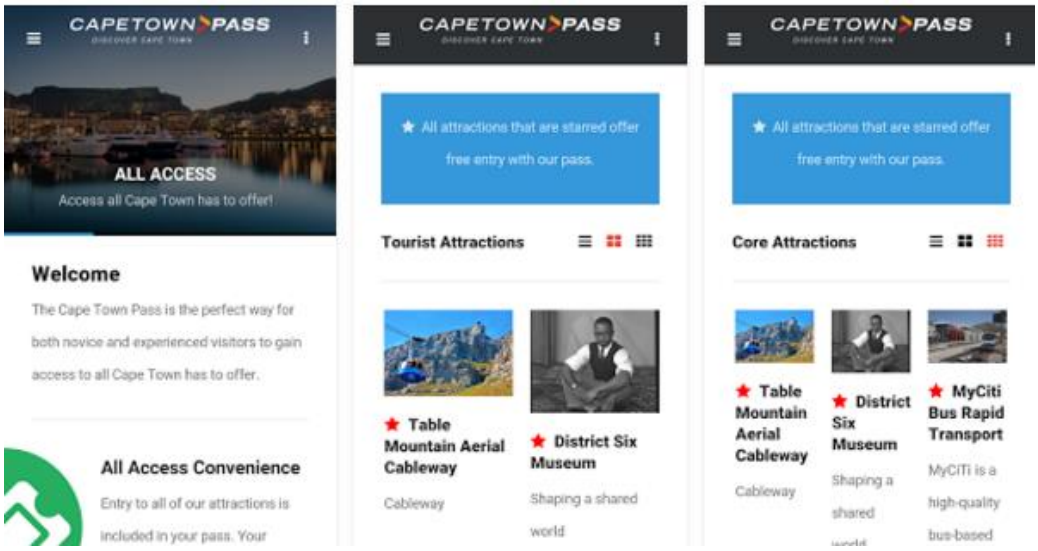


Рисунок 1.6 – Зображення прикладу програми Cape TownPass

Google Trips - це мобільний додаток, розроблений для спрощення планування подорожей і туристичних маршрутів [20,21]. Хоча офіційно додаток було закрито у 2019 році, багато його функцій були інтегровані в інші сервіси Google, такі як Google Maps та Google Travel. Google Trips дозволяв надавати системи автоматизованого пошуку туристичних маршрутів такі основні функціональні можливості:

1) Організація поїздок в одному місці - Google Trips автоматично збирав усю інформацію про поїздки (бронювання готелів, квитки на літаки, автомобілі тощо) з електронної пошти користувача (Gmail) і організовував її у зручному форматі. Користувач отримував доступ до всіх своїх маршрутів, бронювань і важливої інформації в одному додатку.

2) Автономний доступ до інформації - однією з ключових функцій Google Trips був офлайн-доступ. Користувач міг завантажити всю інформацію про свою подорож, включаючи маршрути, туристичні об'єкти, карти та інші дані, щоб користуватися ними без інтернет-з'єднання. Це було особливо корисно для подорожей у місця зі слабким зв'язком або за кордон, де мобільний інтернет може бути дорогим.

3) Індивідуальні рекомендації щодо туристичних об'єктів - Google Trips пропонував рекомендації місць для відвідування, включаючи пам'ятки, ресторани, музеї, парки та інші визначні місця. Він враховував попередні дії користувача, його інтереси та відгуки інших туристів. Також була функція автоматичного створення денних маршрутів на основі популярних місць, які були рекомендовані в певному місті або регіоні.

4) Категоризація та фільтрація місць - користувачі могли сортувати місця за різними категоріями, такими як “Топ-пам’ятки”, “Їжа та напої”, “Шопінг” або “Дозвілля”. Це полегшувало процес вибору місць для відвідування залежно від інтересів користувача.

5) Інтеграція з картографічними сервісами - додаток був інтегрований з Google Maps, що дозволяло користувачам легко знаходити туристичні об'єкти та отримувати інструкції для пересування. Маршрути будувалися в реальному часі, з урахуванням поточного місцезнаходження та умов трафіку.

6)Інформація про час роботи та ціни - користувачі могли переглядати актуальну інформацію про туристичні об'єкти, включаючи години роботи, вартість квитків, а також відгуки та рейтинг від інших користувачів Google.

7) Можливість планування та створення власних маршрутів - Google Trips дозволяв створювати власні маршрути, додаючи улюблені місця або події до списку. Також користувач міг налаштовувати дні поїздки відповідно до своїх планів та цікавих місць.

8) Планування без інтернету - у додатку було передбачено збереження карт і напрямків для навігації без необхідності підключення до Інтернету. Це включало можливість завантажити весь місто для автономної навігації та використання під час подорожі.

До переваг системи Google Trips можна віднести такі його характеристики:

- Зручність організації інформації: додаток автоматично збирав усі бронювання з Gmail і структурував їх, що робило процес планування подорожей менш стресовим.

- Завантаження даних для офлайн-використання було надзвичайно корисним для туристів, які подорожували до країн з поганим інтернет-зв'язком або для економії на роумінгу.

- Пропонування готових денних маршрутів, враховуючи цікаві місця поблизу, що допомагало туристам максимально ефективно планувати свій час.

- Завдяки інтеграції з Google Maps, користувачі могли будувати маршрути, переглядати карти та знаходити оптимальні напрямки для подорожей, що робило додаток комплексним рішенням для подорожей.

- Інтуїтивно зрозумілий інтерфейс спрощував взаємодію з додатком. Користувачі могли легко знаходити інформацію і налаштовувати свої маршрути.

Недоліками системи Google Trips є наступні:

- Залежність від Gmail створювала певні обмеження для тих, хто не використовував Gmail для бронювань. Якщо користувачі використовували інші поштові сервіси, їм потрібно було вручну вводити інформацію про поїздку.

- Відсутність актуальних даних після закриття додатка.

- Обмежений контроль над маршрутом, який полягав у тому, що деяким користувачам бракувало гнучкості та можливості налаштувати маршрути за власними вподобаннями, зокрема змінювати черговість місць або додавати свої зупинки.

- Залежність від Google-сервісів призводила до того, що деякі користувачі, які не мали або не хотіли використовувати обліковий запис Google, були обмежені у використанні додатку. Це також стосувалося тих, хто не користувався Google Maps або Gmail.

- Після закриття додатка у 2019 році підтримка Google Trips припинилася, що змусило користувачів перейти до інших альтернатив або використовувати Google Maps, що не завжди надавало такий самий рівень зручності.

З наведеного аналізу можна зробити висновок, що Google Trips був зручним інструментом для планування подорожей і туристичних маршрутів, особливо завдяки інтеграції з іншими сервісами Google, офлайн-доступу та автоматичній організації поїздок. Однак після закриття додатка багато функцій було втрачено або перенесено до інших додатків.

TripAdvisor – це одна з найпопулярніших платформ для планування подорожей, яка надає користувачам різноманітні інструменти для пошуку туристичних маршрутів, бронювання готелів, ресторанів, екскурсій і розваг [22,23]. Основні функціональні можливості платформи включають:

1) Відгуки та рейтинги туристичних об'єктів - TripAdvisor надає доступ до мільйонів відгуків, оцінок і фотографій, залишених користувачами, які відвідували туристичні об'єкти, готелі, ресторани та інші визначні місця. Це дозволяє туристам отримати об'єктивну інформацію перед тим, як прийняти рішення про відвідування. Користувачі можуть залишати свої власні відгуки та оцінки, допомагаючи іншим подорожуючим.

2) Пошук і фільтрація туристичних об'єктів - TripAdvisor дозволяє шукати туристичні об’єкти за різними критеріями, такими як популярність, рейтинг, категорія (музеї, ресторани, природні парки тощо), відстань від поточного місця розташування або вартість. Можливість застосовувати фільтри, наприклад, відображення лише об'єктів з найвищими рейтингами або лише місць, що підходять для сімейного відпочинку.

3) Планування маршрутів - TripAdvisor пропонує користувачам інструмент “Планувальник подорожей” (Trip Planner), який дозволяє створювати власні туристичні маршрути. Користувач може додавати до маршруту об'єкти, готелі, ресторани й інші місця, які хоче відвідати під час подорожі. Система дозволяє переглядати маршрути інших користувачів і використовувати їх як основу для свого планування.

4) Бронювання туристичних послуг - платформа підтримує функцію бронювання готелів, квитків на авіарейси, ресторанів, екскурсій і різних розваг. Це дозволяє користувачам не тільки планувати, а й бронювати

послуги безпосередньо через додаток або вебсайт. TripAdvisor порівнює ціни на різних сайтах, пропонуючи найкращі варіанти для бронювання.

5) Геолокаційний пошук - TripAdvisor використовує технологію GPS для визначення місця розташування користувача та надає інформацію про туристичні об'єкти, ресторани й готелі поблизу. Це дозволяє швидко знайти цікаві місця неподалік від поточного місцезнаходження або дізнатися про популярні об’єкти в обраній локації.

6) Мобільний додаток - TripAdvisor має мобільний додаток, що дозволяє користувачам планувати подорожі, читати відгуки й переглядати інформацію про місця в будь-який час і в будь-якому місці. Додаток підтримує офлайн-доступ до збережених маршрутів та інформації, що зручно при обмеженому інтернет-з'єднанні.

7) Функція "Питання та відповіді" - користувачі можуть ставити питання про туристичні об'єкти, ресторани або готелі, на які відповідають інші мандрівники або власники бізнесів. Це допомагає отримати конкретну інформацію, яка може бути відсутня в загальних відгуках.

8) Детальна інформація про туристичні об'єкти - користувачі можуть отримувати вичерпну інформацію про туристичні місця: адресу, час роботи, ціни, послуги, контакти, фотографії та інше. Платформа також пропонує списки популярних місць, які можна відвідати в певному місті або регіоні.

9) Порівняння цін – TripAdvisor інтегрований з різними туристичними сервісами (Expedia, Booking.com, Hotels.com та інші), що дозволяє порівнювати ціни на готелі, авіаквитки або екскурсії, забезпечуючи користувачам найкращі пропозиції.

10) Створення персональних списків - користувачі можуть створювати персональні списки об'єктів, які вони планують відвідати, або зберігати вподобані місця для подорожей у майбутньому.

До переваг даної системи можна віднести такі:

- TripAdvisor пропонує одну з найбільших баз даних з мільйонами відгуків, оцінок і фотографій про туристичні об'єкти, готелі, ресторани,

пам'ятки та розваги по всьому світу. Це дає користувачам доступ до величезної кількості реальних відгуків і досвіду інших мандрівників.

- Користувачі можуть переглядати тисячі об’єктивних відгуків і рейтинги від реальних туристів, що допомагає уникнути упереджених або рекламних оглядів. Це дозволяє приймати більш обґрунтовані рішення.

- TripAdvisor інтегрований з багатьма сервісами бронювання, що дозволяє користувачам порівнювати ціни на готелі, авіаквитки, ресторани та інші послуги. Це забезпечує можливість знайти найкращі пропозиції за доступними цінами.

- Додаток і вебсайт дозволяють користувачам створювати персональні туристичні маршрути та зберігати місця для майбутніх поїздок. Це зручний інструмент для організації подорожей і планування відвідувань.

- Мобільний додаток TripAdvisor дозволяє зберігати інформацію про маршрути й туристичні об'єкти для використання в автономному режимі. Це особливо корисно під час подорожей в країни з обмеженим доступом до інтернету або високими тарифами на мобільний зв'язок.

- Завдяки інтеграції з GPS, користувачі можуть швидко знаходити цікаві місця, ресторани й готелі поблизу їхнього поточного місцезнаходження, що значно полегшує подорожі.

- Функція Q&A (питання та відповіді) дозволяє користувачам ставити питання про конкретні об'єкти і швидко отримувати відповіді від інших мандрівників або власників закладів, що забезпечує додаткову інформацію.

- Платформа підтримує бронювання різноманітних туристичних послуг через партнерські сервіси (наприклад, Booking.com, Expedia, Hotels.com), що робить її універсальним рішенням для планування та бронювання подорожей.

- TripAdvisor має активну спільноту користувачів, які діляться досвідом та радять найкращі варіанти для подорожей. Це створює відчуття довіри й підтримки при плануванні подорожі.

До недоліків системи можна віднести необхідність апаратних ресурсів достатнього обсягу і комерційний характер.

Airbnb Experiences – це платформа, що дозволяє користувачам не тільки бронювати житло, але й знаходити унікальні активності, екскурсії та розваги, які організовуються місцевими жителями [24,25,26].

До ключових функціональних можливостей Airbnb Experiences віднрсяться:

1) Пошук туристичних активностей - Airbnb Experiences надає можливість користувачам шукати та бронювати різноманітні заходи та активності в різних містах світу. Це можуть бути кулінарні майстер-класи, прогулянки по місту, екскурсії, заняття спортом, культурні події та багато іншого. Можна застосовувати фільтри для пошуку, наприклад, за тематикою (кулінарія, мистецтво, природа, спорт), тривалістю або рівнем інтенсивності.

2) Організація заходів місцевими жителями - всі події та заходи організовуються місцевими гідами або жителями, що надає можливість туристам отримати аутентичний досвід і зануритися в культуру місця. Це дозволяє познайомитися з місцевими традиціями та звичаями з точки зору самих мешканців.

3) Оцінки та відгуки - користувачі можуть переглядати відгуки та оцінки від інших туристів, які вже брали участь у певних заходах. Це допомагає отримати уявлення про якість і зміст активності, а також про гіда або організатора. Можливість залишати власні відгуки та ділитися досвідом з іншими мандрівниками.

4) Інтеграція з основним сервісом Airbnb - Airbnb Experiences інтегровано з основною платформою для бронювання житла, що дозволяє користувачам не лише забронювати місце для проживання, але й легко знайти активності неподалік. Це забезпечує комплексне планування подорожей. Рекомендації щодо активностей можуть бути персоналізовані на основі місця проживання або попередніх бронювань.

5) Онлайн-експерії - окрім офлайн-заходів, платформа також пропонує онлайн-експерії (онлайн-подорожі), які дозволяють користувачам брати участь у різноманітних подіях з будь-якої точки світу, наприклад, кулінарні майстер-класи, медитації, віртуальні тури тощо.

6) Гнучкі дати та ціни - користувачі можуть вибирати активності з гнучким графіком та різними ціновими категоріями, що дозволяє знайти події, які відповідають їх бюджету та часу. Платформа надає можливість порівняти різні варіанти активностей у обраному регіоні.

7) Унікальні та ексклюзивні заходи - багато активностей є ексклюзивними для Airbnb Experiences, тобто їх не можна знайти на інших туристичних платформах. Це можуть бути унікальні культурні заходи, арт-проекти, приватні екскурсії або майстер-класи, створені спеціально для користувачів Airbnb.

8) Мобільний додаток - Airbnb Experiences доступний через мобільний додаток, що дозволяє користувачам швидко шукати та бронювати активності під час подорожей безпосередньо зі смартфона [28,30]. Додаток має інтуїтивний інтерфейс і підтримує сповіщення про майбутні заходи.

9) Безпека і підтримка - платформа забезпечує захист користувачів та підтримку під час планування і проведення заходів. Airbnb надає гарантії безпеки, відшкодування у випадку проблем, а також оперативну службу підтримки.

Недоліки Airbnb Experiences:

- платформа орієнтована на унікальні та нестандартні заходи, тому користувачі можуть не знайти популярні або традиційні туристичні екскурсії, які надають інші сервіси;

- оскільки активності організовуються різними людьми, якість заходів може відрізнятися. Хоча відгуки і допомагають вибрати якісну послугу, ризик розчарувань залишається;

- в менших містах або віддалених регіонах вибір заходів може бути значно обмеженим або взагалі відсутнім. Платформа краще працює в популярних туристичних напрямках;

- деякі заходи на платформі можуть бути дорожчими порівняно з подібними на інших сервісах. Це зумовлено ексклюзивністю активностей або приватними екскурсіями.

Як висновок, можна відзначити, що Airbnb Experiences – це чудова платформа для тих, хто шукає унікальні враження та хоче зануритися в місцеву культуру під час подорожей, але підходить не всім через обмежений вибір в деяких регіонах і можливі цінові нюанси.

1.4 Стан вирішення проблеми

Недосконаліть існуючих систем автоматизованого пошуку туристичних об’єктів та багато інших проблем дійсно уповільнюють розвиток туризму в наші дні . Вирішення проблеми істотно вплине на покращення відвідування туристичних об’єктів в загальному [27], що спричинить стрімкий розвиток туризму. Отже для того щоб допомогти людям прийняти рішення можна застосувати декілька методів:

1. По-перше, потрібно максимально ефективно покращити рекламні оголошення. Що означає покращити? Зробити рекламу не тільки інформативною, але й супроводжувати її аргументами, які будуть сприяти на покращення прийняття рішень. З іншого боку рекламу потрібно розмістити в місцях, в яких турист зможе найкращим чином побачити її та впевнитись в її ефективності. Як впевнитись в її ефективності? Все просто, люди довіряють відгукам. Розмістити біля місця реклами відгуки інших людей, які вже відвідали це місце або збираються його відвідати. Це дійсно буде сприяти покращенню рекламних оголошень та їх ефективності.

2. По-друге, зробити перегляд інформації про об’єкт доступним та зручним для людини, зробивши цю процедуру комфортною.

3. По-третє, запропонувати вибір. В рекламах зазвичай показують інформацію про щось одне, проте можна дати людині зробити вибір на основі не тільки реклами, а й відгуків інших людей.

1.5 Шляхи вирішення та вибір оптимального варіанту

Якщо підійти до проблеми з точки зору системного підходу, можна дуже багато досягнути. Методів для вирішення проблеми інформування про туристичні маршрути і їх автоматичного вибору може бути достатньо. Якщо ж підійти з точки зору технологій, то це дійсно можна вирішити декількома відомими методами, створивши програмне забезпечення:

- створення мобільного додатку;
- створення веб сайту з кабінетом користувача;
- створення інформативного веб сайту на основі Landing Page, де користувач зможе залишити свої дані та зв’язатись з власниками для подальших переговорів, чи вирішень певних питань;
- створення Desktop-програми для вашого ПК;
- і навіть створити інформативну програму для вашого девайсу(PK/Tablet/Mobile) [29,31] без підключення до мережі інтернет, що забезпечить просто інформацію про певні об’єкти.

Методів багато, але всі мають свої недоліки. Почнемо з інформативної програми для девайсу на вибір, без підключення до мережі інтернет. Це дійсно класно коли можна дізнатись про багато туристичних об’єктів просто сидячи за своїм девайсом. Але без підключення до мережі інтернет також постають певні проблеми, такі як:

- неможливість оновити програму, тобто взяти нову інформацію про неї, це дуже не вигідно розробляти програму без підтримки, та ризиковано;
- з сучасним стрімким розвитком технологій дана програма буде дуже швидко ставати не популярною, та унеможливлювати підтримку на

системах чи пристроях, що у свою чергу викличе дуже малий відсоток користування;

– без доступу до мережі унеможливлюється підтримка бази даних, тобто всі дані потрібно буде зберігати в програмі, що дійсно вплине на її розмір, та швидкість.

Перейдемо до інших методів. Створення інформативного Landing Page є дуже хорошим варіантом, але достатньо дорогим. Тому щоб заощадити кошти, на мою думку краще розробити повноцінний Web site. Щодо Desktop-програми, теж дуже хороший варіант, але дорогий. І розробка Desktop-додатків є досить важкою. Також дуже багато буде залежати від операційної системи. Якщо програма буде для Windows, нею не зможуть користуватись користувачі в яких MacOS. А зробити це все кросплатформенним буде дуже дорого коштувати та займе величезну кількість часу, ще ж до того ж за таке завдання візьметься не кожен розробник, оскільки це дійсно важко. Найкращим варіантом, на мою думку, є створення мобільного додатку. Існує декілька причин переваги цього методу:

– мобільний додаток буде використовуватись майже всіма, оскільки смартфон є одним із найпопулярніших, найдоступніших та найкомфортнішим девайсом на даний момент;

– зв’язок з мережею інтернет – взагалі не проблема;

– не залежить так сильно від вибору операційної системи, що Android [8] дуже популярна, що iOS;

– зробити кросплатформенним не займе дуже багато часу та коштів;

– висока надійність пристрою, нема проблем із акумулятором.

Існує декілька недоліків цього методу. В основному це вузький вибір технологій, який буде забезпечувати кроссплатформенність. Якщо розробляти додаток не кроссплатформенним, тоді це займе більше часу та зусиль. Як для мобільно операційної системи на базі Android, так і iOS є багато технологій для написання програмного забезпечення. Якщо подумати

то єдиною максимально ефективною технологією для розробки мобільних додатків зараз можна вважати технологію Xamarin, на базі .net. Мова програмування для написання C#. Інший недолік, через не дуже високу швидкість інтернету на мобільних девайсах додаток буде не таким швидким, як хотілося б. До того ж такі технології як Google Play дійсно будуть уповільнювати його роботу. Але це залежить не від розробників, Google Play Services величезна структура, яка має багато мікросервісів та послуг, тому очікувано, що програма буде програвати по швидкості Web сайтам чи навіть тому самому Desktop додаткові. Насправді, для мене як для користувача важливість цієї програми не зменшується через швидкість. Наявність цієї програми, головне, що надає змогу користування, адекватного, правильного вибору туристичних об’єктів на основі візуальної реклами, перегляду інформації, відгуків інших людей.

Розробка мобільного додатку не тільки не займе багато часу, вона ще й багато років буде зберігати свою популярність. Уявити себе на хвилину без свого мобільного телефону/смартфону сьогодні дуже важко [30]. Є дуже багато сервісів, які підтримують карти, що також покращить роботу даного додатку. Це не тільки сучасно, а й дуже гарно візуально, вибирати об’єкти для туристичного візиту на карті. Для різних операційних систем є доступними різні карти. Найпопулярнішими є Google Maps для Android [26], які підтримуються Google Play Service, та iOS Maps для iOS. Є дуже цікавим той факт, що користувач зможе залишати відгуки, та передивлятись всі об’єкти та ще й інформацію про них. На мою думку це дійсно ефективний метод вирішення проблеми, яка постає. Це значно вплине на туризм, та на відношення людей до туристичних об’єктів в цілому.

1.6 Формулювання вимог до розроблюваної системи

Ми живемо у час швидкого розвитку технологій. Важко уявити, що майже кожен день людство розробляє та забезпечує себе новими

технологіями та ресурсами. На даний момент провідні компанії та підприємства використовують нові технології, щоб забезпечити безпеку, швидкість прийняття рішень, та багато іншого. Кожна з них самостійно вчиняє вибір систем, якими слід користуватись. За звичай майже всі системи працюють на архітектурі клієнт/сервер. Таким чином досягається високий рівень оптимізації, відбувається значне заощадження коштів, зусиль та часу.

Розроблена система також має клієнтську сторону та серверну. Клієнтською стороною вважається мобільний інтерфейс, який буде встановлюватись на мобільному девайсі. Там буде відображатись вся інформація, яка буде синхронізуватись із серверною частиною. Серверна частина в свою чергу відповідає за цілісність та безпеку передавання даних. Всі дані логічно структуровані та забезпечуються логічною архітектурою програмного коду. Взаємодія між клієнтською і серверною стороною і отримання/відправлення запитів зображено на рисунку 1.6:

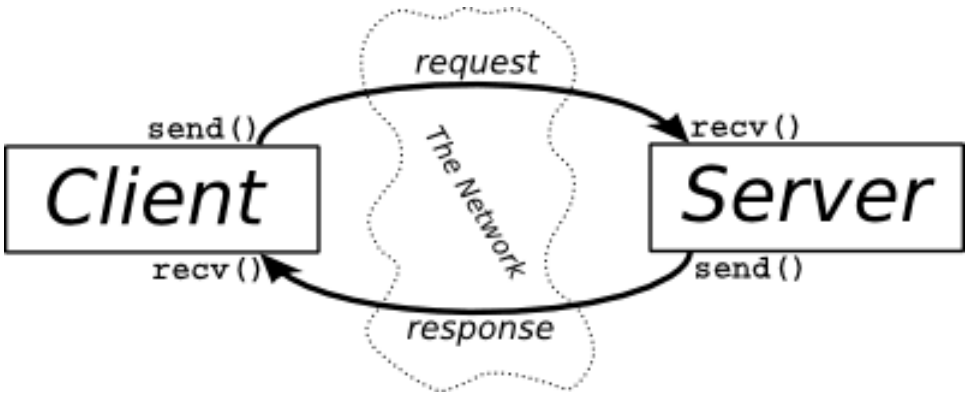


Рисунок 1.6 – Взаємодія між сторонами клієнт/сервер

Аналіз прийнятої інформації з клієнта може логічно оброблятись як на клієнті так і на сервері. Але прийнято робити це на сервері, це зменшує кількість запитів, та впливає на швидкість системи.

Взаємодія між клієнтом і сервером доволі проста, клієнт вибираючи, що йому потрібно на UI(User interface) посилає потрібні запити на сервер. Запити обробляються. Якщо це наприклад запит на певні дані, запит

формується, і генерується автоматично в запити на мові SQL, що дозволяє тісно взаємодіяти з базою даних, і відправляється на сервер. Сервер в свою чергу приймає та опрацьовує запити, робить всі потрібні дії, і віддає дані клієнту, або ж клієнтам. Логічно, що один сервер має зв'язок з клієнтом як 1 to N. Зв'язок, відомий ще як один до багатьох. Це розповсюджений зв'язок в структурі баз даних, який означає, що один сервер може взаємодіяти з декількома клієнтами. А як вже буде це реалізовано, це вже технологія написання програмного коду, тобто як клієнти будуть опрацьовуватись на сервері, асинхронно чи в черзі, тощо.

2 ВИБІР ТЕХНОЛОГІЇ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИБОРУ ТУРИСТИЧНИХ ОБ'ЄКТІВ

2.1 Огляд технологій, що використовуються для розробки

В розробці системи використовувалось багато технологій. Почнемо з базової. Базовою технологією для розробки клієнтської та серверної сторони є Xamarin [31], оснований на платформі .net, мова програмування C#. Докладніше розглянемо цю та інші технології, що використовувались при розробці системи автоматизованого вибору туристичних об'єктів.

2.1.1 Технологія Xamarin

Xamarin – одна із провідних технологій, розроблена на платформі .net. В першу чергу звісно Xamarin – це американська компанія, яка займається розробкою програмного забезпечення. Провідною рисою технології є кроссплатформенна розробка мобільного програмного забезпечення. Це справді актуальна технологія, оскільки для того щоб розробити додаток на різні платформи, такі як Android чи iOS, потрібно вивчати такі мови програмування як Java і Objective-C.

Сутність технології проста і доступна, все що потрібно це просто писати код на мові C#, зі застосуванням цього фреймворка ви отримаєте доступ до всіх нативних технологій розробки різних мобільних платформ, таких як Android [33], iOS, Windows Phone. Фреймворк складається з деяких компонентів, котрі в свою чергу забезпечують розробку кожної платформи, її тестування та підтримку плагінів/сервісів:

- плагін для IDE, мається на увазі Visual Studio [33] або Xamarin Studio, тому що тільки ці IDE підтримують Xamarin;
- IDE Xamarin Studio або Visual Studio;
- компілятори для Android та iOS;
- бібліотеки класів для C#, які надають доступ до Android та iOS SDK;

– емулятори для Windows Phone та Android.

Xamarin оснований на реалізації open-source через платформу .net. На платформі в свою чергу розвертається Mono. Ця реалізація використовує середу виконання всіх .net бібліотек, компіляцію і середу виконання. Фреймворк значно спрощує запуск програм на різних платформах, які в свою чергу всі написані на мові програмування C#. Для цього передбачено різну компіляцію на стороні Android і iOS. Все просто, логічно та прозоро. Але є один, на мою думку, своєрідний недолік. Компіляція та виконання додатку на iOS неможливе без Mac OS. Як відомо не всі віддають перевагу, чи не можуть собі дозволити Mac OS, тому частіше всього розробники піднімають віртуальну машину з цією ОС на своєму комп'ютері, незалежно від платформи. Ядра для Mono і Dalvik написані на мові C, що означає повну незалежність від ОС. Оскільки на стороні з розробкою iOS є недолік, то потрібно його урівняти якоюсь позитивною функцією. Розробка на Objective-C, рідній мові програмування для iOS пристроїв відрізняється від розробки на C#. В чому різниця? Різниця в тому що код на C# не займає так багато місця як код на Objective-C, що є дуже ефективною рисою для розробинків. Адже чим код лаконічніший, тим його легше читати та і взагалі працювати з ним. Різниця між кодом написаним на Objective-C [34] та C# зображено на рисунку 2.1:

Рисунок 2.1 – Різниця між кодом Objective-C та C# на фреймворкові Xamarin

Часто досвідченні розробники вважають за потрібне робити додатки з великою архітектурою асинхронними, що забезпечує високу швидкість та меншу трату ресурсів. В Xamarin це не є проблемою, адже можна використовувати простор імен System.Threading та інші бібліотеки C#. Це дуже високо підносить даний фреймворк.

Детальніше про UI. З UI розробники Xamarin зробили все дуже зручно. Це означає, що якщо ви вже працювали з розробкою мобільних додатків то вам буде набагато легше. Вони надають право використовувати нативні способи розробки UI та можливість використання всіх нативних елементів користувацького інтерфейсу. Наприклад для Android [35] можна створювати UI або в коді, або візуальним редактором, або взагалі використовуючи XML розмітку. Для iOS все так само, або ж в коді, або візуально, або використовуючи спеціальні xib-файли.

Хотілося б не залишити без уваги ще таку річ, як архітектуру та структуру коду. Xamarin не потребує написаних повністю програм для всіх ОС, оскільки він є кроссплатформенним [36]. Структура виглядає таким чином, що для кожної платформи потрібно писати свій UI, скажемо що це невеличка плата за користування нативними елементами всіх систем. Коли ми відійдемо від UI там все простіше. Архітектура просто ділиться на певні шари, приклад яких ви можете побачити на рисунку 2.2:

Рисунок 2.2 – Архітектура Xamarin додатку

На рисунку 2.2 зображена умовна архітектура звичайної програми написаної на 3 платформах за допомогою фреймворку Xamarin. Зазвичай в великих додатках архітектура має схожу структуру, але не має 3 різних UI елементи. Розглянемо їх детальніше:

- Data Layer (DL) – шар бази даних, всі дані які зберігаються або передаються;
- Data Access Layer (DAL) – шар доступу до даних, відомі в світі розробки CRUD-операції;
- Business Layer (BL) – шар бізнес-логіки додатку, тут зберігатиметься вся логіка проекту, що і як себе буде поводити в певних ситуаціях;
- Service Access Layer (SAL) – шар взаємодії з даними (Rest, Json, WCF);
- Application Layer (AL) – шар з кодом залежно від платформи monotouch.dll або monodroid.dll;
- User Interface Layer (UI) – шар користувацького інтерфейсу, пам’ятаєм, що умовно їх 3, до кожної з платформ, так як зображено на рисунку вище.

2.1.2 Використання карт

Оскільки для зручності вирішено показувати об’єкти на картах, розглянемо технології які передбачають використання технологій на базі фреймворку Xamarin. Технологія карт [37] є обов’язковим доповненням до мобільних пристроїв. Настільні пристрої зазвичай не мають інформацію про постійне місцезнаходження або ж користувач може регулювати це. З іншого боку, дуже часто мобільні пристрої використовують такі додатки для виявлення пристроїв або для відображення змін про інформацію місцезнаходження. Android, так само як і iOS має досить потужні вбудовані технології, які виявляють дані про місцезнаходження на картах з використанням апаратних засобів визначення місцеположення, які можуть бути доступні на пристрої. Зазвичай виявлення місцезнаходження широко

використувується через WIFI або ж через GPS навігацію. Карти під Xamarin для Android та iOS повинні запропонувати користувачеві такі функції:

– використання вбудованих карт-додатків для швидкого додавання функціональних можливостей відображення;
– робота з API карт для управління відображенням карти на мобільних пристроях;
– використання різних методів для додавання графічних функцій накладення на карту певних фігур чи зображень.

На рисунках 2.3 та 2.4 зображено часте використання Xamarin [36] карт як для Android(Google Maps) так і для iOS(iOS Maps):

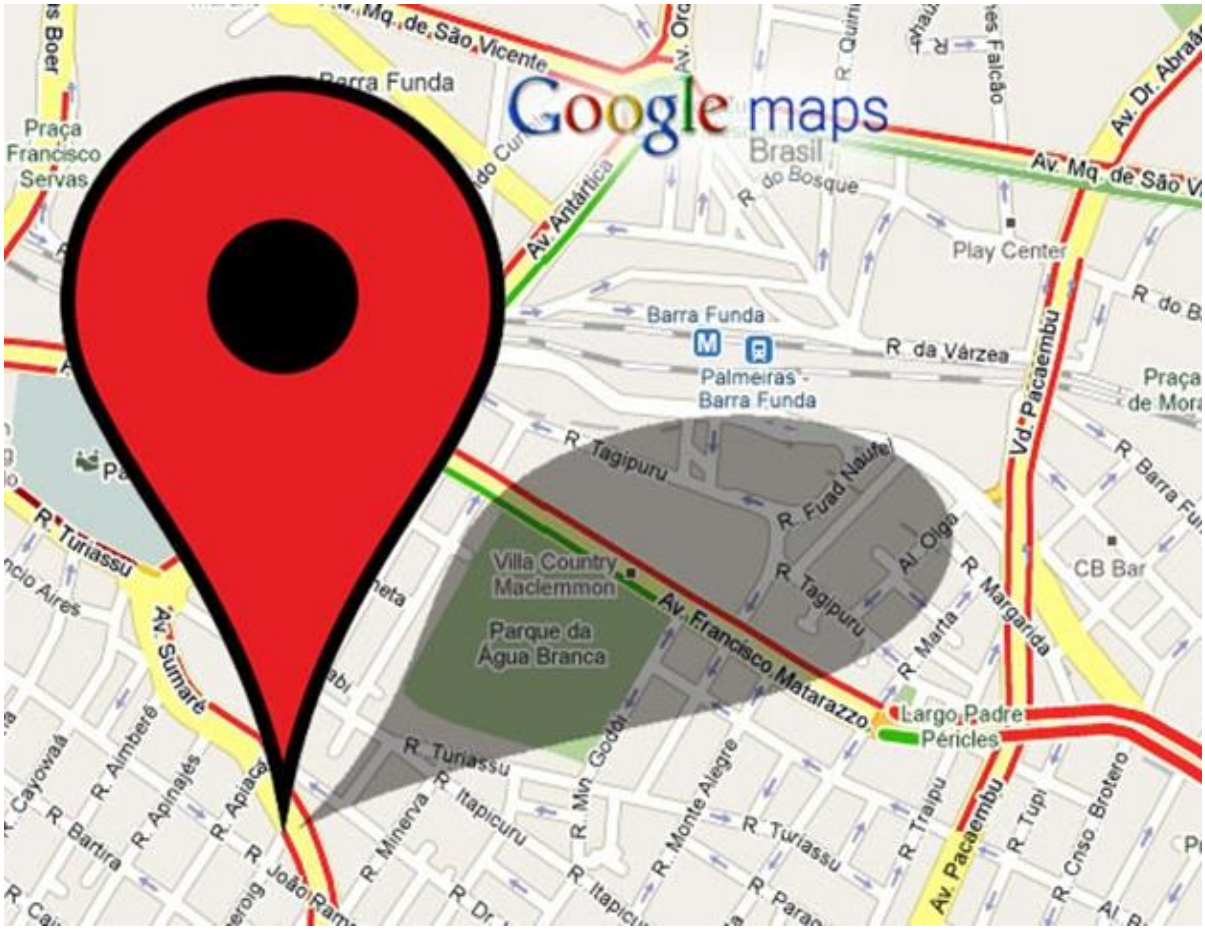


Рисунок 2.3 – Google maps для Android

Рисунок 2.4 – iOS maps для iOS

Переваги та недоліки використання різних видів карт. Чому саме карти Google та iOS maps? Все дуже просто, тільки ці карти добре співпрацюють з фреймворком Xamarin. Але все ж можна розглянути докладніше чим ці карти краще. Працюючи над картами, їх не так просто відображати. Для того щоб карти коректно відображались на девайсі [38], потрібно зробити Api. Існує багато статей про знайомство чи використання з Google Api, Yandex Api, Bing Api, та багато інших. Суть в тому, що Google Api є найпопулярнішим серед всіх. Причиною цього є його простота та прозорість для того щоб реалізувати. Є багато документації, що і як потрібно правильно зробити, щоб отримати Google Api та працювати з ним, використовуючи карти. До того ж Google Api має не одне покоління. Існує 3 чи 4 покоління карт Google Api [38], і реалізація кожного відрізняється. Тому було вибрано ці карти, адже вони є найкращими. На рисунку 2.5 зображено Google Developers Console, яка призначається для роботи з Google Api:

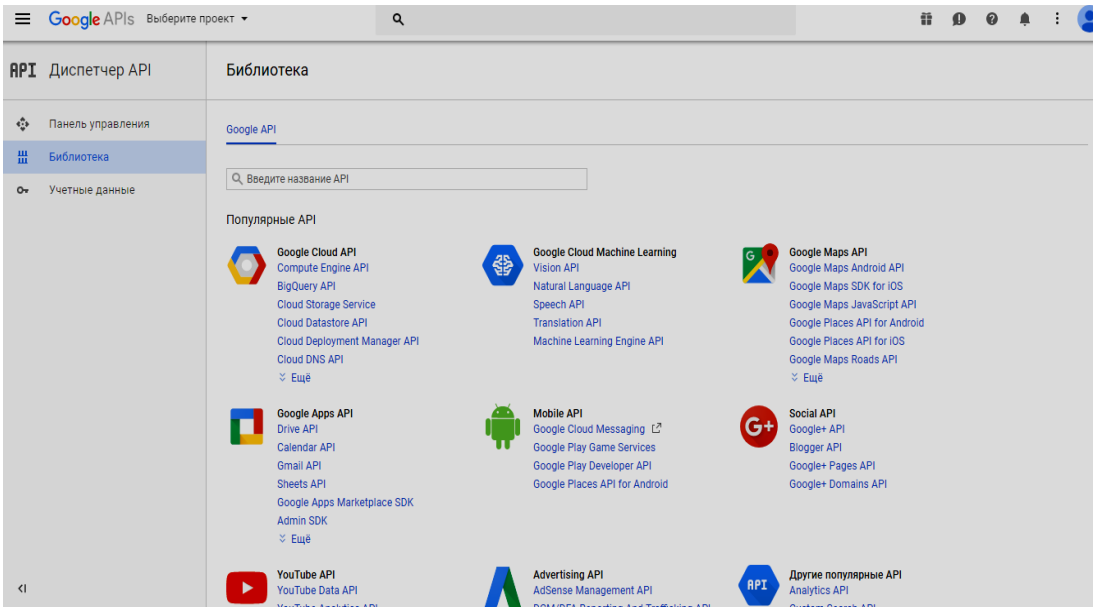


Рисунок 2.5 – Google Developers Console

2.1.3 База даних та мова SQL запитів

База даних – це сукупність даних, структурно організованих та сортованих, яка описує характеристики цих даних і зв’язки між ними. Дані у базі зазвичай організовують відносно моделі організації даних. Однією з найпопулярніших моделей організації даних є реляційна модель. Суть реляційної моделі полягає в тому, що дані зберігаються в таблицях. Це дуже структурно організовано та зручно для маніпуляції ними. Можна виконувати багато функцій за допомогою мови SQL [39] запитів. Запит оброблюється сервером і виконується, що впливає на поточний результат, тобто це легкий спосіб маніпулювання даними. Мова запитів проста та зручна для користування. Оскільки використовується Microsoft Visual Studio як IDE, то для виконання максимально ефективно використовувати Microsoft SQL Server Management Studio [40], яка являється IDE в сфері SQL запитів. Це дуже зручна програма для користування, і вона тісно пов’язана з Visual Studio, оскільки їх розробником є одна й та ж сама компанія. Microsoft SQL [40] Server у свою чергу використовується для керування базами даних, це система керування, яка була створена компанією Microsoft для виконання

структурованих SQL запитів. На рисунку 2.6 показана IDE Microsoft SQL Server Management Studio:

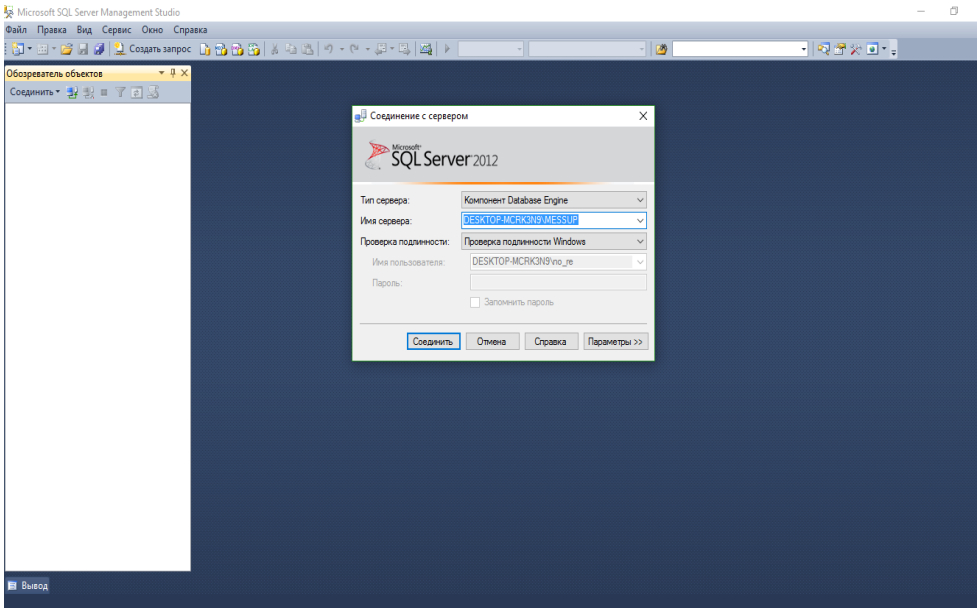


Рисунок 2.6 - Microsoft SQL Server Management Studio

2.2 Аналіз алгоритмів вибору оптимальних маршрутів

Задачу вибору оптимального туристичного маршруту можна представити моделлю $M_{СТМ}$ повного зваженого графа [41,42]:

$$M_{СТМ} \to G(V,E) \qquad (2.1),$$

У виразі (2.1) V – множина вершин графа, що відповідають компонентам розв’язку,

$$V = \{v_1, v_2, ..., v_i, ..., v_n\}, \qquad (2.2)$$

E- множина ребер графа,

$$E = \{e_{ij}\}, \qquad i = (\overline{1,n}),\ j = (\overline{1,n}), \qquad (2.3)$$

ребра вказують на можливі переходи між відповідними вершинами графа.

Модель передбачає визначення функції вартості для всіх ребер, яка може залежати від часового параметра *t*. Також в такій моделі задається деяка множина обмежень $L = L(V, E, t)$ для вершин і ребер, які встановлюють допустимість зв'язків між вершинами *V* та ребрами *E* (відповідно між об'єктами задачі та переходами між ними). Дані обмеження визначають результати підсумкового розв'язку задачі. Всі розв'язки задачі оптимізації задаються у вигляді допустимих шляхів на заданому графі *G*.

Оптимальний розв'язок задачі вибору туристичного маршруту в даній інтерпретації може бути знайдений шляхом використання комбінаторних алгоритмів оптимізації. Дані алгоритми знаходять допустимі шляхи мінімальної вартості, які задовольняють встановленим в моделі обмеженням.задачі. При цьому значення цільової функції *f*(*x*),у вигляді вартості відповідає тому розв'язку *x*, який мінімізує функції вартостей всіх з'єднань, що належать знайденому розв'язку. Розглянемо характеристики основних алгоритмів комбінаторної оптимізації.

2.2.1 Класифікація алгоритмів комбінаторної оптимізації

Алгоритми комбінаторної оптимізації маршрутів є важливим розділом в обчислювальній науці та математиці, що має широкий спектр застосувань – від логістики і транспортних систем до планування і телекомунікацій. Вони використовуються для вирішення задач оптимального планування маршрутів, розміщення ресурсів та координації дій, що мінімізує витрати, час або інші ресурси. Наведемо огляд основних підходів до використання алгоритмів комбінаторної оптимізації маршрутів.

Комбінаторна оптимізація включає методи, що шукають оптимальний спосіб поєднання дискретних елементів за визначеними критеріями. Задачі маршрутизації зазвичай належать до класу NP-складних задач, що означає, що час їх вирішення збільшується експоненційно із зростанням розмірності

задачі. До найпоширеніших задач комбінаторної оптимізації маршрутів відносяться такі [43] :

- задача комівояжера (Traveling Salesman Problem, TSP), що полягає в знаходженні найкоротшого шляху, що дозволяє відвідати всі задані пункти один раз і повернутися до початкового пункту;

- задача про розвезення (Vehicle Routing Problem, VRP), в якій знаходять оптимальний маршрут для транспортного засобу з метою мінімізації загальної відстані або часу обслуговування клієнтів;

- задача пошуку найкоротшого шляху, в якій знаходять найкоротший маршрут між двома заданими точками в графі.

Основні підходи до вирішення вказаних задач, реалізуюься за допомогою методів, що поділяються на три класи: точні, евристичні та метаевристичні.

На рис. 2.7 показано їх різновиди.

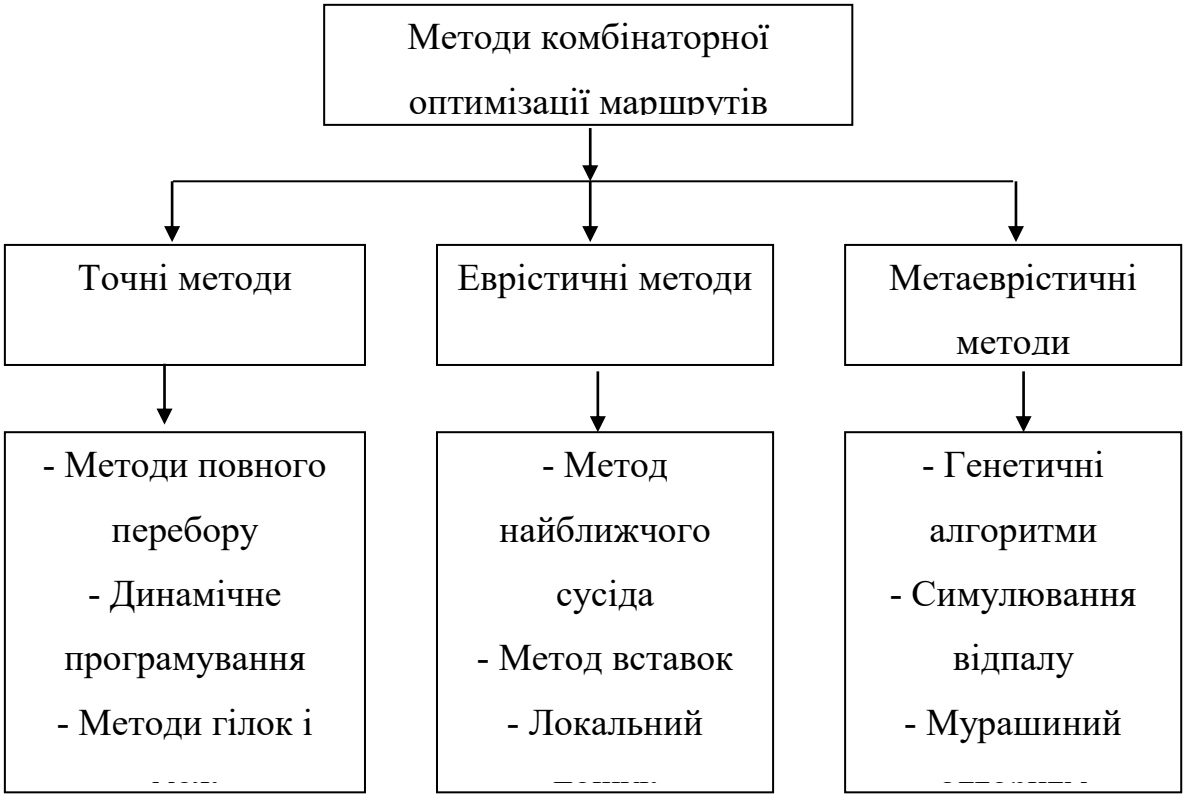


Рисунок 2.7 – Класифікація алгоритмів комбінаторної оптимізації

Кожен із цих підходів має свої переваги, недоліки і сфери застосування, які випливають з їх особливостей, описаних нижче.

Коротко розглянемо суть наведених методів і алгоритмів.

Точні методи комбінаторної оптимізації

Точні методи забезпечують пошук оптимального рішення, гарантуючи, що знайдене рішення є найкращим можливим. Найпоширенішими серед них є методи перебору, динамічного програмування, розгалуження та границь, а також методи лінійного програмування.

Методи повного перебору. Цей підхід передбачає перевірку всіх можливих варіантів рішень. Наприклад, у задачі комівояжера для графу з nxn міст повний перебір перевіряє всі $(n-1)!(n-1)!$ можливих маршрутів. Однак обчислювальна складність таких методів є експоненційною, що робить їх неприйнятними для великих задач.

Динамічне програмування. Метод динамічного програмування, наприклад алгоритм Беллмана-Хелда-Карпа для TSP, використовує принцип оптимальності Беллмана, розбиваючи задачу на підзадачі і комбінуючи їхні результати. Попри поліпшену ефективність порівняно з перебором, складність алгоритму залишається високою.

Метод розгалуження та границь. Цей метод ґрунтується на побудові дерева рішень і відсіченні гілок, які не можуть містити оптимального рішення. Він значно зменшує обсяг обчислень, але все одно залишається обчислювально затратним для великих задач.

Точні методи ідеально підходять для невеликих задач або в ситуаціях, коли оптимальність рішення є критично важливою. Проте через їхню складність вони поступаються місцем евристичним і метаевристичним підходам у великих і реальних задачах.

Евристичні методи.

Евристики — це алгоритми, які шукають «достатньо хороші» рішення за відносно короткий час, але не гарантують оптимальності. Вони засновані на певних правилах або емпіричних спостереженнях.

Конструктивні евристики. Ці алгоритми створюють рішення з нуля, додаючи компоненти до поточного часткового рішення. Наприклад:

- Метод найближчого сусіда (nearest neighbor) у задачі комівояжера: починається з випадкової вершини і додає до маршруту найближчу невідвідану вершину.
- Метод вставки (insertion heuristic), де точки додаються до маршруту так, щоб мінімізувати приріст вартості.

Конструктивні методи швидкі, але часто дають рішення, що значно відрізняються від оптимального.

Методи поліпшення. До них належать локальні пошуки, які починаються з певного початкового рішення і поступово покращують його, перевіряючи сусідні варіанти. Прикладом є алгоритм 2-opt, який замінює дві дуги маршруту, щоб зменшити загальну довжину.

Хоча евристики є швидкими й ефективними для великих задач, вони схильні «застрягати» у локальних мінімумах і не здатні масштабно досліджувати простір рішень.

Метаевристичні методи.

Метаевристики — це вищий рівень евристичних алгоритмів, розроблений для ефективного пошуку оптимальних або близько оптимальних рішень у великих просторах рішень. Вони об'єднують локальний пошук із глобальними механізмами, які дозволяють уникати локальних мінімумів.

Генетичні алгоритми (GA). Генетичні алгоритми натхненні еволюційними процесами. Вони працюють із популяцією рішень, використовуючи операції кросоверу, мутації і відбору. GA ефективно досліджують великий простір рішень, але потребують налаштування параметрів, таких як розмір популяції і частота мутацій.

Імітація відпалу (Simulated Annealing, SA). SA моделює процес охолодження металу. Алгоритм поступово зменшує ймовірність прийняття гірших рішень, дозволяючи уникнути локальних мінімумів. Він простий у

реалізації і добре підходить для задач, де потрібно знаходити рішення в обмеженому просторі.

Ройові методи. До них належать алгоритми на основі поведінки живих організмів, як-от алгоритм мурашиної колонії (ACO) і оптимізація рою частинок (PSO). Наприклад, ACO імітує пошук їжі мурахами, формуючи ефективні шляхи за допомогою феромонів.

Табу-пошук розширює локальний пошук, забороняючи повернення до щойно відвіданих рішень. Це дозволяє уникати зациклення і покращує якість рішень.

Метаевристики є гнучкими та ефективними, але вони не гарантують оптимального рішення. Їхньою перевагою є можливість вирішення великих задач за розумний час, що робить їх ідеальними для реальних проблем.

Виконаємо порівняння даних методів та розглянемо практичне застосування.

- Точні методи підходять для малих задач із високими вимогами до точності.

- Евристики забезпечують швидкі рішення для задач середнього масштабу.

- Метаевристики є незамінними для великих задач із численними обмеженнями.

У реальних сценаріях часто використовують гібридні підходи, які поєднують точні алгоритми з евристиками або метаевристиками. Наприклад, спочатку може використовуватися евристика для отримання початкового рішення, яке потім доопрацьовується метаевристичним алгоритмом.

З проведеного порівняльного аналізу методів комбінаторної оптимізації можна зробити такий висновок.

Точні, евристичні та метаевристичні методи комбінаторної оптимізації маршрутів мають різні сильні та слабкі сторони. Точні методи забезпечують оптимальність, але є обчислювально затратними. Евристики швидкі, але не завжди дають якісні результати. Метаевристики пропонують ефективний

компроміс між точністю та швидкістю, що робить їх найбільш придатними для вирішення великих і складних задач. Розвиток цих підходів є важливим кроком до автоматизації та оптимізації транспортних і логістичних систем.

2.2.2 Обгрунтування використання мурашиного алгоритму

Вчені на основі проведених експериментів встановили, що поведінка мурах є теоретично оптимальною [44]. Це проявляється в тому, як вони будують мурашники, транспортують їжу, обходять перешкоди, тощо. Характерною рисою їхньої взаємодії є використання тільки локальної інформації без звернення до глобального образу чи централізованого управління.

Мурашині алгоритми є одним із найпопулярніших методів, які застосовуються для розв’язання задач комбінаторної оптимізації, таких як задача комівояжера, оптимізація транспортних мереж, розклад завдань і багато інших. Вони надихаються поведінкою реальних мурах, які колективно шукають оптимальні маршрути між колонією та джерелами їжі.

Мурахи передають інформацію двома способами: прямим, який включає обмін їжею, мандибулярні, візуальні та хімічні контакти, та непрямим – через стигмережі [31]. Стигмерережі являють собою опосередкований спосіб взаємодії, коли окремі елементи системи змінюють середовище, залишаючи сигнали, які інші суб’єкти використовують у майбутньому. Біологічно це реалізується через феромони – спеціальні хімічні речовини, що відкладаються мурахами під час руху. Чим більша концентрація феромону на певному маршруті, тим більше мурах його обирає. Феромони мають здатність випаровуватися з часом, що дозволяє адаптувати поведінку мурах до змін у навколишньому середовищі. Просторовий розподіл феромонів виконує функцію динамічної "глобальної пам’яті", яка змінюється в реальному часі. Приклад використання феромонів для пошуку оптимального шляху ілюструє рисунок 2.8.

Феромонні сліди полегшують пошук найкоротших маршрутів іншими мурахами, а принципи цього процесу допомогли розробити ефективні обчислювальні алгоритми для вирішення задач оптимізації маршрутів. У туризмі мурашині алгоритми можуть допомогти в автоматизованому плануванні маршрутів, що дозволяє враховувати різні потреби туристів та обмеження.

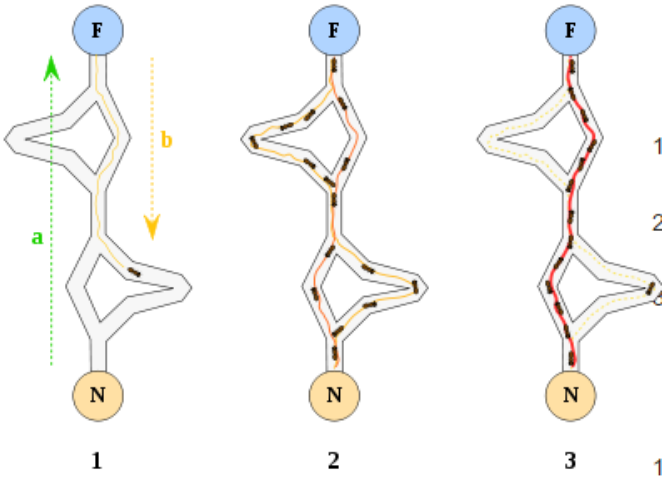


Рисунок 2.8 – Зображення мурашиних шляхів

Мурашині алгоритми є одним із найпопулярніших методів, які застосовуються для розв’язання задач комбінаторної оптимізації, таких як задача комівояжера, оптимізація транспортних мереж, розклад завдань і багато інших. Вони надихаються поведінкою реальних мурах, які колективно шукають оптимальні маршрути між колонією та джерелами їжі.

Ключовою ідеєю цих алгоритмів є використання феромонного сліду, який мурахи залишають під час пересування. Феромон надає інформацію про вигідність маршруту: чим більше мурах проходить через певний шлях, тим сильнішим стає його слід. У свою чергу, це підвищує ймовірність того, що наступні мурахи виберуть цей маршрут. У штучних мурашиних алгоритмах цей принцип реалізується як ітеративний пошук оптимальних рішень з оновленням "феромонного графа". Розглянемо характеристики найпоширеніших мурашиних алгоритмів.

Класичний мурашиний алгоритм (ACO). Є одним із найвідоміших є алгоритм Ant Colony Optimization (ACO)**,** був запропонований Марко Доріго у 1990-х роках. У цьому алгоритмі агенти (мурахи) досліджують простір рішень, використовуючи баланс між жадібністю (локально оптимальні рішення) і дослідженням нових шляхів. Застосування до:

- Задачі комівояжера (TSP);

- Оптимізації розкладу;

- Прокладання маршруту в мережах (наприклад, комп'ютерних мережах або логістиці).

ACO працює ефективно, проте має обмеження в обчислювальній складності на великих графах. Існує ряд модифікацій ACO, які розширюють його можливості. Для покращення ефективності ACO було запропоновано кілька варіантів алгоритму:

- *MAX-MIN Ant System (MMAS):* У цьому алгоритмі вводяться обмеження на максимальний і мінімальний рівень феромону, що дозволяє уникати надмірної концентрації мурах на одному шляху та сприяє дослідженню інших рішень.

- *Ant Colony System (ACS):* Використовує більш інтенсивну стратегію локального оновлення феромону, що пришвидшує збіжність алгоритму.

- *Rank-Based Ant System (RAS):* Пріоритет надається найкращим рішенням, які роблять більший внесок у оновлення феромону.

Відомі також **гібридні мурашині алгоритми,** які отримують шляхом інтегрування з іншими методами оптимізації:

- *Генетичні алгоритми:* Комбінування мурашиних стратегій із генетичними операторами, як-от кросовер і мутація, для розширення пошукового простору.

- *Методи рою часток (PSO)***:** Поєднання глобального пошуку PSO із локальною оптимізацією мурашиних алгоритмів.

- *Машинне навчання:* Використання моделей для передбачення вигідності маршрутів або адаптації параметрів алгоритму.

Спеціалізовані алгоритми є варіантами мурашиних алгоритмів, що орієнтовані на специфічні задачі:

- *Мурашиний алгоритм для динамічних графів:* Застосовується для задач, де граф змінюється в реальному часі, наприклад, у маршрутизації в дорожньо-транспортних мережах.

- *Мурашиний алгоритм для багатокритеріальної оптимізації:* Використовується для задач, де потрібно враховувати кілька факторів, як-от час і вартість.

До переваг мурашиних алгоритмів відносяться:

- Природна здатність до пошуку глобального оптимуму;

- Висока адаптивність до різних типів задач;

- Можливість паралельної реалізації.

Водночас мурашиним алгоритмам притаманні недоліки, до яких можна віднести:

- Велика обчислювальна складність для великих задач;

- Наявність параметрів, які складно налаштувати;

- Ризик ранньої збіжності до локального оптимуму.

Мурашині алгоритми знайшли широке практичне застосування в багатьох сферах, до основних з яких відносяться:

- Логістика: Оптимізація перевезень і складування.

- Телекомунікації: Побудова маршрутів у мережах.

- Робототехніка: Навігація автономних роботів.

- Медицина: Планування променевої терапії.

Наведений огляд мурашиних алгоритимів різних типів дає можливість зробит наступні висновки.

Мурашині алгоритми є потужним інструментом для розв’язання задач комбінаторної оптимізації завдяки їхній природній здатності моделювати колективний пошук. Водночас, їх ефективність значною мірою залежить від правильного налаштування параметрів і вибору модифікацій. Завдяки постійному розвитку та інтеграції з іншими методами, мурашині алгоритми

залишаються актуальними в сучасних дослідженнях і практичних застосуваннях.

Розглянемо принципи роботи мурашиного алгоритму та його можливості для покращення досвіду планування туристичних маршрутів.

Основні принципи роботи мурашиного алгоритму такі.

Мурашиний алгоритм імітує пошук маршруту у мурашиній колонії, використовуючи наступні основні елементи:

1. Феромонний слід: кожна мураха залишає слід на своєму маршруті, інтенсивність якого залежить від якості маршруту. Чим коротший маршрут, тим сильніший феромонний слід, що збільшує ймовірність його обрання іншими мурахами.

2. Випаровування феромонів: з часом феромонний слід випаровується, що запобігає «застою» на неефективних маршрутах.

3. Стохастичний вибір маршруту: мурахи вибирають маршрут на основі ймовірності, яка залежить від сили феромонного сліду та відстані. Це дозволяє знаходити нові шляхи й запобігає зацикленню на одному варіанті.

Алгоритм розвивається циклічно: мурахи створюють нові маршрути, а інформація про них постійно оновлюється за допомогою феромонних слідів. У задачах пошуку оптимального туристичного маршруту ці принципи дозволяють ефективно аналізувати безліч варіантів, аби знайти найкращий маршрут, який задовольняє потреби туриста.

У сучасних інформаційних системах для планування подорожей і турів мурашині алгоритми можуть бути інтегровані в сервіси з використанням геолокації, що дозволить розробляти маршрути в режимі реального часу. Це корисно як для індивідуальних туристів, так і для туроператорів.

Для туристів завдяки їх використанню можливе індивідуальне планування: турист, який планує самостійний тур, може вказати свої інтереси, доступний час та бажані точки відвідування. Мурашиний алгоритм обробить ці параметри, згенерує маршрут і запропонує найоптимальніший варіант, враховуючи відстань між точками, трафік та інші дані.

Для туристичних компаній та агентства мурашині алгоритми для групових турів можуть розробляти маршрути, які забезпечують максимальну зручність для групи та зменшують витрати на транспорт. Оператори можуть також пропонувати різні пакети турів із оптимізованими маршрутами, що враховують найкращий спосіб відвідання кількох туристичних об'єктів у короткий проміжок часу.

Також одним із напрямків застосування можуть бути онлайн-сервіси для туристів: платформи та додатки можуть використовувати мурашині алгоритми для створення адаптивних і персоналізованих рекомендацій, які змінюються в режимі реального часу. Це дозволить туристам оперативно змінювати плани залежно від умов на місці (наприклад, закриття музею, затори на дорогах тощо).

На основі вищеописаних особливостей застосування мурашиних алгоритмів в якості алгоритму комбінаторної оптимізації туристичних маршрутів можна зробити такий узагальнюючий висновок. Мурашині алгоритми мають великий потенціал у сфері туристичних послуг, оскільки дозволяють автоматизувати процес планування маршрутів і робити його ефективнішим. Враховуючи динамічні фактори, які важко передбачити, мурашиний алгоритм забезпечує гнучкість і здатність адаптуватися до змін. Його застосування в туризмі допоможе створювати зручніші, швидші та економічно вигідніші маршрути, покращуючи загальний туристичний досвід. Впровадження таких алгоритмів у платформи для туризму і подорожей може стати важливим кроком у розвитку галузі, дозволяючи Україні та іншим країнам пропонувати туристам високий рівень зручності та обслуговування.

2.3 Розробка алгоритму пошуку оптимального маршруту

2.3.1 Вербальний опис мурашиногоалгоритму

Під час розробки мурашиного алгоритму вибору оптимального туристичного маршруту потрібно реалізувати кілька ключових

обчислювальних кроків, що імітують поведінку мурах при пошуку найкоротшого шляху. Нижче вербально описано дані основні етапи [45]:

1. Ініціалізація

На цьому кроці задаються початкові параметри:

- Кількість мурах: обирається кількість віртуальних мурах, які одночасно починають пошук маршрутів.
- Феромонна матриця: створюється матриця, яка містить початкові рівні феромонів на кожному шляху між туристичними об’єктами. Зазвичай спочатку феромонний слід задається на одному рівні для всіх маршрутів.
- Параметри феромонів та випаровування: встановлюються коефіцієнти інтенсивності феромонів, рівень випаровування та інші параметри, які впливають на оновлення феромонів під час роботи алгоритму.

2. Побудова маршрутів

Кожна мураха починає створювати маршрут, переходячи від одного туристичного об'єкта до іншого. При виборі наступного об’єкта мураха враховує:

- Інтенсивність феромону: ймовірність обрати певний шлях збільшується з ростом концентрації феромону на ньому.
- Евристичний фактор (наприклад, відстань) : перевага може надаватися маршрутам з коротшою відстанню між об’єктами.

Цей крок є стохастичним – вибір кожного шляху залежить від ймовірності, розрахованої на основі значень феромонів та відстаней. Мураха будує маршрут до тих пір, поки не відвідає всі заплановані пункти.

3. Оцінка маршруту

Після завершення маршруту мураха обчислює його вартість , що зазвичай визначається загальною довжиною маршруту, часом у дорозі або іншими параметрами, важливими для конкретної туристичної задачі.

4. Оновлення феромонів

Після того як всі мурахи завершили свої маршрути, оновлюється феромонна матриця:

- Випаровування феромонів : на всіх маршрутах зменшується кількість феромонів, що дозволяє алгоритму уникнути застою на менш оптимальних шляхах.

- Нанесення нового феромону : мурахи, які пройшли більш оптимальні маршрути, залишають сильніші феромонні сліди. Чим кращий маршрут, тим більше феромону залишається на ньому, збільшуючи ймовірність його вибору в наступних ітераціях.

5. Повторення та зупинка алгоритму

Кроки 2–4 повторюються протягом кількох ітерацій, поки не виконається одна з умов зупинки:

- досягнення визначеної кількості ітерацій;
- стабілізація значення найкращого маршруту;
- обмеження на час обчислення.

Після завершення алгоритм надає оптимальний або близький до оптимального маршрут.

Існують варіанти мурашиного алгоритму, які додають покращення, наприклад, регулювання феромонів залежно від кращих знайдених маршрутів або комбінування з іншими методами (локальний пошук, генетичні алгоритми) для прискорення збіжності. Ці кроки дозволяють мурашиному алгоритму адаптивно шукати найкращий маршрут, який задовольняє вимоги туристичної задачі, навіть за наявності складних обмежень та великої кількості варіантів. До цих варіантів мурашиного алгоритму можна вднести [44] базовий мурашиний алгоритм AS; модифікація цього мурашиний алгоритму з введенням елітних мурах ASE; максимінний мурашиний алгоритм MMAS; ранговий мурашиний ASR; ACS – алгоритм колонії мурах. Ці алгоритми мають велику перевагу над детермінованими комбінаторними алгоритмами оптимізації, оскільки їх обчислювальна складність залежить від розмірності задач поліноміально порівняно з поліноміальною залежністю детермінованих алгоритмів [42].

2.3.2 Розробка математичної моделі мурашиного алгоритму

Математичну модель мурашиного алгоритму представимо з урахуванням результатів, отриманих в роботах [44, 45, 46].

Опишемо вхідні дані для моделі. До основних параметрів, від яких залежить перехід агента (мурахи) з об'єкта i в об'єкт j по ребру (шляху) (i,j) на t-ій ітерації залежить від трьох параметрів, які відповідають володінню мурахами власною "пам'яттю", "нюхом" і "зором" відповідно: списка заборонених місць (табу списка), видимості та віртуального сліду феромону. Введемо позначення цих параметрів:

- $J_{i,k}$ – список об'єктів, які потрібно відвідати k-ій мурасі, що знаходться в i-тому місті (це міста, що не знаходяться в запам'ятованому табу-списку відвіданих міст);

- $\eta_{ij} = 1/D_{ij}$ – видимість, величина, обернено пропорціональна відстані між містами D_{ij}, визначає евристичне бажання k-ої мурахи відвідати місто j;

- $\tau_{ij}(t)$ - кількість феромону на ребрі (i,j) на t-й ітерації, показує евристичне бажання k-ої мурахи відвідати місто j з міста i на підставі досвіду інших мурах.

Тоді ймовірність переходу k -го мурашки в місто j з міста i можна описати ймовірнісно-пропорційним рівнянням:

$$\begin{cases} P_{ij,k}(t) = \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{l \in J_{,j,k}} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}, \; j \in J_{i,k}, \\ P_{ij,k}(t) = 0, \; j \in J_{i,k}, \end{cases} \tag{2.4}$$

У виразі (2.4) параметри α і β є регульованими. Вони задають ваги інтенсивності сліду феромону τ_{ij} та видимості η_{ij}. При $\alpha = 0$ найбільш вірогідним буде перехід у найближче місто. В теорії оптимізації це відповідає "жадібному" алгоритму. При $\beta = 0$ перевага надається інтенсивності сліду феромону підсилення, при цьому алгоритму швидко

закінчує роботу з причини збіжності маршрутів усіх мурах, що даєв даному випадку субоптимальний розв'язок.

Кількість феромону, яку відкладає на ребро (i,j) кожна k-мураха після завершення маршруту, визначається виразом:

$$\Delta\tau_{ij,k}(t)=\begin{cases}\dfrac{Q}{L_k(t)}, (i,j)\in T_k(t);\\ 0.(i,j)\notin T_k(t).\end{cases} \qquad (2.5)$$

У виразі (2.5) через $T_k(t)$ позначено маршрут, пройдений k-ою мурахою на ітерації t, $L_k(t)$ визначає його довжину, а Q є регульованим параметром, величину якого вибирають такого ж порядку, що і довжина оптимального маршруту.

Для забезпечення можливості експлуатації простору рішень потрібно забезпечити випаровування феромону — зменшення кількості відкладеного на попередніх ітераціях феромону. Інтенсивність випаровування феромону задається за допомогою коефіцієнта випаровування

Правила зовнішнього середовища визначають, в першу чергу, випаровування феромону. Нехай $p\in[0..1]$ є коефіцієнтом випаровування, тоді правило випаровування має вигляд

$$\begin{aligned}&\tau_{ij}(t+1)=(1-\rho)\cdot\tau_{ij}(t)+\Delta\tau_{ij}(t),\\ &\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij,k}(t),\end{aligned} \qquad (2.5)$$

Кількість феромону на переходах $i\to j$ на початку процедури оптимізації приймається рівною невеликому додатному числу τ_0. Кількість мурах в мурашиній колонії залишається постійною і рівною кількості міст на весь час розв'язання задачі. Маршрут кожної мурахи починається зі свого міста.

В модифікованому алгоритмі вводять деяких “елітних’ комах, здатних посилювати ребра найкращого маршруту T^* довжини L^*, що був знайдений від початку пошуку. При наявності в мурашиній колонії e елітних мурах ребра даного маршруту отримають підсиленн на величину

$$\Delta\tau_e = e \cdot \frac{Q}{L^*} \ . \qquad (2.6)$$

Наведемо псевдокод розробленого мурашиниго алгоритму, який вирішує задачу оптимізації туристичних маршрутів.

Initialization of parameters n, α, β, ρ, Q, τ_0

N ← n ▷ number of town

pheromone (i,j) ← τ_0 ▷ matrix of pheromone levels between towns i , j

η(i, j) ← 1/D(i, j) ▷ visibility(i,j) between towns i , j

alpha ← α ▷ influence of pheromone

beta ← β ▷ influence of visibility

rho ← ρ ▷ evaporation coefficient of pheromone

Q ← Q_{in} ▷ amount of pheromone deposited by an ant after completing a route

num_ants ← m ▷ number of ants

iterations ← t_{max} ▷ number of iteration

Basic algorithm

for *t* = 1 to iterations :

  for *k*=1 to num_ants:

    initialize current city (random or given);

    choose a route until all cities have been visited:

      calculate the transition probabilities to each available city:

        prob[i][j] = (pheromone[i][j]^alpha) * η(i, j) ^beta) /

```
                    the sum of all possible transitions;
            choose the next city using the roulette wheel method;
            add a city to the route;
        calculate the length of the route;
        update the best route found if the current one is better;
          End
End
   # Pheromone update:
      for  i=1 to N:
         for j=1 to N:
            pheromone(i,j) = pheromone(i,j) * (1 - rho)     ▷ evaporate pheromones
          End
      End
      for  i=1 to N:
         for j=1 to N:
            pheromone(i,j)  = pheromone(i,j) + Q / T_k(t)    ▷ add new pheromones
          End
      End
   # Get the best route
```

Пояснення ключових кроків:

1. Ініціалізація: Параметри (α, β, ρ) контролюють баланс між впливом феромону та видимості (інверсії відстані).

2. Пошук маршруту: Мураха послідовно вибирає міста з урахуванням рівня феромону та відстані до наступного міста.

3. Оновлення феромону: Рівень феромону зменшується на всіх шляхах (випаровування) та збільшується на маршрутах, які використовували мурахи, з урахуванням їхньої ефективності.

4. Оптимізація: Після завершення всіх ітерацій зберігається найкоротший знайдений маршрут.

В розроблюваній в магістерській роботі системі мурашиний алгоритм застосовується в системі в якості коментарів, які туристи будуть залишати(по бажанню) після відвідування об’єктів. Для того щоб залишити коментар потрібно буде здійснити декілька дій, а саме заповнити всі текстові поля для цього, тобто написати своє ім’я, свій коментар, та що сподобалось найбільше(можна порівнювати об’єкт з іншими, та на основі цього робити висновки, чому було б краще та ефективніше в плані часу чи коштів відвідати саме цей об’єкт). Також по бажанню турист буде залишати свої контактні дані, під коментарем, в разі чого з ним можна буде зв’язатись та отримати додаткову інформацію. Для того щоб коментарі відповідали цим стандартам, було введено в програмному коді поле IsValidated(чи є валідним), щоб можна було контролювати коментарі. Смаки туристів, жителів міста звісно ж відрізняються, але це дозволить приймати рішення на основі вибору інших людей. Програмний код міститься в Додатку В.

2.4 Висновки до розділу

В даному розділі був проведений аналіз способів та технологій для побудови системи автоматизованого вибору туристичних об’єктів на базі мобільного телефону. Проаналізовано переваги та недоліки технологій та обґрунтовано вибір технології проектування системи автоматизованого вибору оптимальних туристичних маршрутів. Також проведено детальний огляд технологій алгоритмів комбінаторної оптимізації маршрутів, обґрунтовано вибір мурашиного алгоритму для пошуку оптимального туристичного маршруту та проведена його розробка.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір програмної платформи

Програмування мобільних додатків для Android та iOS вимагає вибору платформи чи мови програмування, які найкраще відповідають завданням проекту. Розглянемо переваги та недоліки основних платформ і мов програмування. До таких можна віднести нативні Android (мови Java/Kotlin), iOS (Swift/Objective-C), кросплатформенні Flutter (Dart), React Native (JavaScript), Xamarin (.NET/C#), Ionic (HTML/CSS/JavaScript) та гібридну Unity (C#). Серед відомих розглянемо тільки кросплатформенні, оскільки розроблювану систему передбачається встановлювати на телефони з двома різними операційними системами Android та iOS.

Платформа **Flutter** використовує мову програмування Dart і має такі переваги:

- Єдиний код для обох платформ: Значно зменшує витрати часу на розробку.

- Висока продуктивність: Завдяки використанню власного графічного рушія (Skia).

- Гнучкість дизайну: Забезпечує кастомізацію та адаптивний інтерфейс.

- Швидка розробка: Функція Hot Reload дозволяє миттєво бачити зміни.

До недоліків можна віднести необхідність значних ресурсів на пристрої та менш популярну мову програмування Durt, що обмежує її доступність для досвідчених розробників.

Серед переваг платформи React **Native** можна відзначити такі:

- Єдиний код для Android і iOS, що знижує витрати на розробку.

- Популярна мова JavaScript, яка має велику кількість бібліотек і ресурсів.

- Активна підтримка та велика кількість готових модулів.

- Підтримка функції Hot Reloading, яка дозволяє прискорити процес розробки.

До недоліків відносяться менша продуктивність, ніж у нативних додатків., залежність деяких критичних функцій від сторонніх рішень, складність інтеграції рідного коду, яка проявляєься в тому, що для доступу до низькорівневих функцій пристрою потрібно писати рідний код.

Характеристики платформи Ionic, яка використовує мови програмування HTML,CSS, JavaScript, наступні.

Переваги:

- Простота використання: Використання стандартних веб-технологій.

- Швидкий прототип: Ідеально підходить для MVP і простих додатків.

- Єдиний код: Економія часу на розробку.

Недоліки:

- Продуктивність: Залежність від веб-переглядача робить додатки повільнішими.

- Обмежений доступ до рідних функцій: Використання плагінів часто є обов'язковим.

Для розробки системи була обрана технологія Xamarin і мова C #, що є найпопулярнішою мовою для написання бізнес логіки на платформі .NET, основні переваги технології Xamarin:

- Використання кроссплатформенної розробки.
- Можливість розширення, так як платформа Android, iOs, Windows Phone.
- Використання однієї бізнес логіки для написання додатку.
- Тестування. Відбувається розділення різних завдань програми на різні, незалежні один від одного частин програмного забезпечення, дана структура дозволяє будувати зрозумілі програми, що легко тестуються. Для кожного фрагмента компонентно-орієнтованого дизайну платформи, за даної структури, забезпечена

структурованість, необхідна для виконання вимог модульного тестування та засобів макетування.

Як IDE була обрана Visual Studio 2019 – це інтегроване середовище розробки Microsoft (IDE) [47], що є самою функціональною IDE для розробки на платформі .NET.

Для модульного тестування була обрана вбудована підтримка в Visual Studio 2019 [48] оскільки зараз функціональність вбудованої підтримки нітрохи не гірше ніж у конкурентів, а тісна інтеграція з IDE ставить її на перше місце.

Для зберігання даних була обрана СУБД SQL Server 2017 і ORM Entitty Framework від Microsoft. Так як обидва продукти від Microsoft у них є найбільш тісна інтеграція з іншими продуктами.

Фреймворк Xamarin має чимало переваг над іншими платформами розробки, являє собою середовище розробки для Mobile Development [49], яка включає в себе різні сервіси і API, що спрощують розробку програм. Xamarin є незалежною технологією, оскільки хоч цей популярний фреймворк реалізований на мові C# є частиною .NET Framework, але він не є продуктом компанії Microsoft. Отже це дозволяє використовувати переваги керованих додатків і звертатися до бібліотеки класів .NET Framework.

Отже, технологія Xamarin та мова програмування C# забезпечують простоту в освоєнні та стабільність роботи додатка, а також надзвичайну швидкість розробки програм завдяки об’єктно-орієнтованому підходу [50].

3.2 Розробка UML-діаграми використання

UML діаграма використання – це діаграма [51], на якій зображується використання системи, а саме дії між акторами та прецедентами. Користуватись системою будуть туристи та жителі міста. Тобто вони і є наші актори. Для того щоб їх об’єднати, назвемо актора customer, що з англійського користувач. Все просто та дуже доступно. Прецедентами

діаграми будуть дії які зможе користувач виконувати. В системі автоматизованого вибору користувач зможе виконувати такі дії:

- переглянути мапу та побачити на ній різні об’єкти;
- перевірити інформацію про об’єкти, дізнатись більше про них;
- якщо потрібно залишити коментар, який в майбутньому допоможе комусь здійснити вибір;
- перевірити коментарі інших людей, що може спонукати до вибору певного об’єкту;
- зробити вибір.

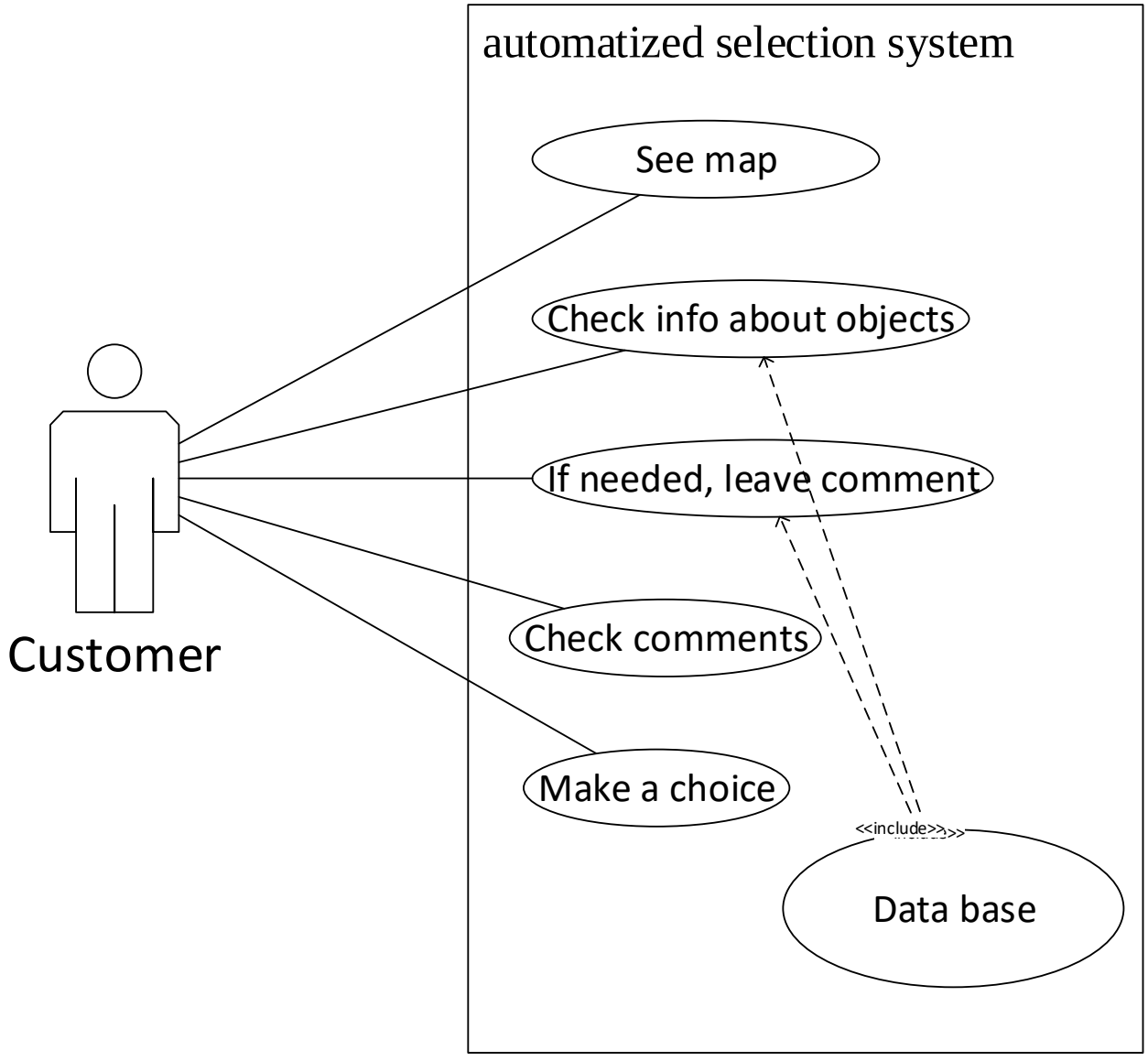


Рисунок 3.1 – UML-діаграма варіантів використання

Потрібно зауважити, що коментар, який залишить користувач, буде занесений в базу даних. Всі коментарі будуть занесені в базу даних та там будуть зберігатись протягом певного періоду, поки будуть актуальними. На коментарі будуть значні обмеження, задля параметрів безпеки, оскільки користувачеві, щоб залишити коментар не потрібно реєструватись. Отже для того щоб люди не залишали коментарі, які можуть пошкодити програму або ж вплинути значно на її роботу, буде зроблена validation of comments. В ній зможемо поставити максимальний та мінімальний об’єм коментаря, який буде залишати користувач. Це убереже нас від значних втрат пам’яті, та від певних проблем в роботі програми. Інформація про об’єкти буде братись з бази даних для зручності. Щоб не зберігати інформацію в програмі, або в проміжних файлах, які можуть втрачатись під час обробки інформації та запуску додатку. Діаграма варіантів використання зображена на рисунку 3.1:

3.3 Розробка схеми програми роботи системи

При вивченні програмування необхідно мати чітку уяву, що таке алгоритм, як він працює і як його створити/запрограмувати. Від добре спроектованого алгоритму залежить скільки часу буде займати відлагодження програми і пошук помилок. Такий алгоритм повинен мати такі властивості; універсальність, скінченність, результативність. Також він повинен показувати структуру виконання програми або частини коду, що є дуже важливим фактором для модифікації програми в разі необхідності.

Блок схема алгоритму системи зображена на рисунку 3.2.

Алгоритм описує порядок взаємодії користувача з розробленим додатком вибору туристичних маршрутів, встановленому на телефоні. На початку (Start) користувач знаходить іконку додатку на екрані телефону і тапає по ній, запускаючи його в роботу (Launching the app). Якщо додаток запускається (умова if Success - працює Інтернет), то користувач має

можливість з ним працювати (Continue app) і обирає одну із запрограмованих опцій (Choosing action):

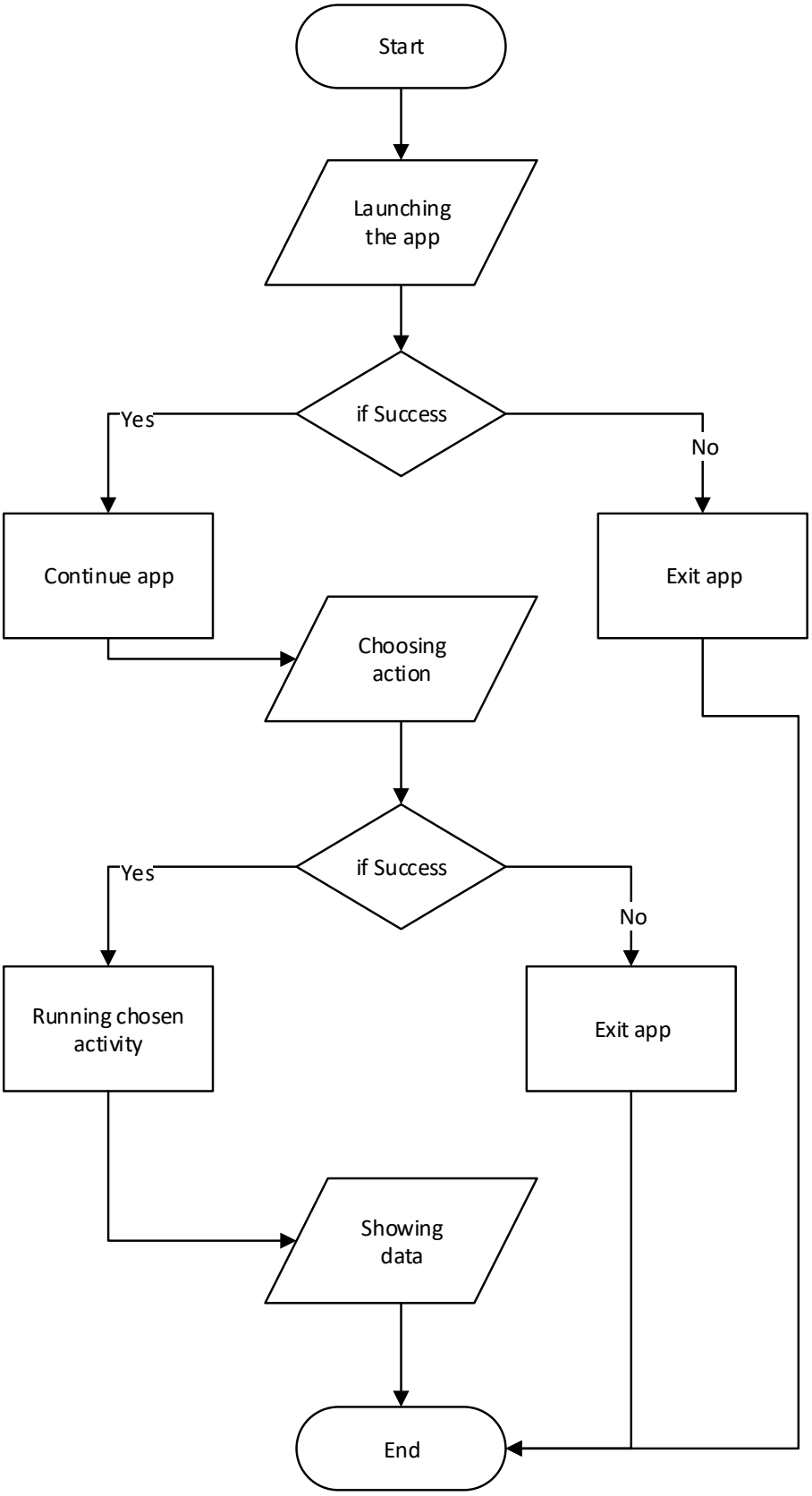


Рисунок 3.2 – Схема алгоритму роботи системи

- перегляд карти;

- перегляд інформації про туристичні об’єкти;

- запис коментара в разі необхідності;

- вибір коментарів інших відвідувачів;

- прийняття рішення про вибір об'єкта.

В разі спрацювання вибраної опції (умова if Success) виконує обрану в ній діяльність (Running chosen activity).

Якщо якась з наведених умов не виконується, робота з додатком закінчується.

Розроблений алгоритм описує схему взаємодії з додатком в укрупненому вигляді. В деталізованому вигляді алгоритм роботи реалізований у вигляді програми мовою C#, лістинг якої наведений в довідниковому Додатку В.

3.4 Розробка діаграми класів системи

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класів може зображати, зокрема, різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи відношень. На даній діаграмі не вказується інформація про часові аспекти функціонування системи. З цієї точки зору діаграма класів є подальшим розвитком концептуальної моделі проектованої системи.

Діаграма класів представляє собою деякий граф, вершинами якого є елементи типу «класифікатор», які зв'язані різними типами структурних відношень. Варто зауважити, що діаграма класів може також містити інтерфейси, пакети, відношення і навіть окремі екземпляри, такі як об'єкти і зв'язки. Коли говорять про дану діаграму, мають на увазі статичну структурну модель проектованої системи. Тому діаграму класів прийнято

вважати графічним представленням таких структурних взаємозв'язків логічної моделі системи, які не залежать від часу.

Діаграма класів складається з множини елементів, які в сукупності зображають декларативні знання про предметну область. Ці знання інтерпретуються в базових поняттях мови UML, таких як класи, інтерфейси і відношення між ними і їх складовими компонентами. При цьому окремі компоненти цієї діаграми можуть утворювати пакети для представлення більш загальної моделі системи. Якщо діаграма класів є частиною деякого пакету, то її компоненти повинні відповідати елементам цього пакету.

У загальному випадку пакет статичної структурної моделі може бути представлено у виді однієї або декількох діаграм класів. Декомпозиція деякого представлення на окремі діаграми виконується для зручності і графічної візуалізації структурних взаємозв'язків предметної області. При цьому компоненти діаграми відповідають елементам статичної семантичної моделі. Модель системи, у свою чергу, повинна бути узгоджена з внутрішньою структурою класів, яка описується мовою UML.

Клас у мові UML використовується для позначення множини об'єктів, які мають однакову структуру, поведінку і відносини з об'єктами з інших класів. Графічно клас зображається у виді прямокутника, який додатково може бути поділений горизонтальними лініями на секції. У цих секціях можуть вказуватися ім'я класу, атрибути (змінні) і операції (методи).

Обов'язковим елементом позначення класу є його ім'я. На початкових етапах розробки діаграми окремі класи можуть позначатися простим прямокутником з зазначенням тільки імені відповідного класу. У процесі розробки окремих компонентів діаграми описи класів доповнюються атрибутами і операціями. Передбачається, що кінцевий варіант діаграми містить найбільш повний опис класів, який складається з трьох секцій. Іноді в позначеннях класів використовується додаткова четверта секція, в якій приводиться семантична інформація довідкового характеру або явно вказуються виняткові ситуації. Навіть якщо секція атрибутів і операцій пусті,

в позначенні класу вони виділяються горизонтальною лінією, щоб відразу відрізнити клас від інших елементів мови UML.

Ім'я класу повинне бути унікальним в межах пакету, який описується деякою сукупністю діаграм класів. Воно вказується в першій верхній секції прямокутника. Ім'я класу записується по центру секції імені жирним шрифтом і повинно починатися з великої букви. Рекомендується як імена класів використовувати іменники, які записуються без прогалин. Саме імена класів утворюють словник предметної області при ООАП. Крім імені при позначенні класу, можуть бути посилання на стандартні шаблони або абстрактні класи, від яких утворений даний клас і, відповідно, від яких він успадковує властивості і методи. Клас може не мати екземплярів або об'єктів. У цьому випадку він називається абстрактним класом, а для позначення його імені використовується курсив. На рисунку 3.3 зображено діаграму класів для головного проекту(TouristsObjects.Core):

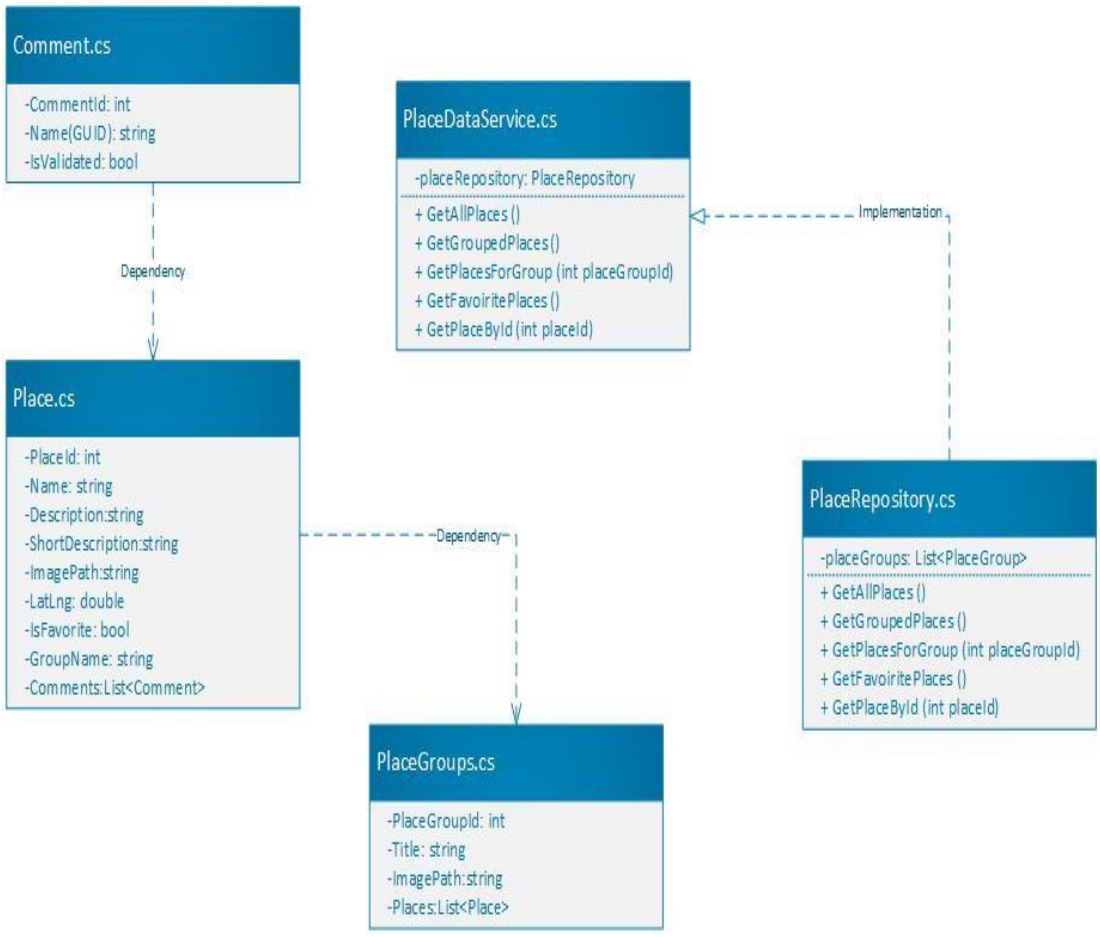


Рисунок 3.3 – Діаграма класів для головного проекту

На рисунках 3.4-3.5 зображено діаграму класів для проекту TouristsObjects.Android, діаграма розділена на дві частини, перша частина зображена на рисунку 3.4, а друга на рисунку 3.5.

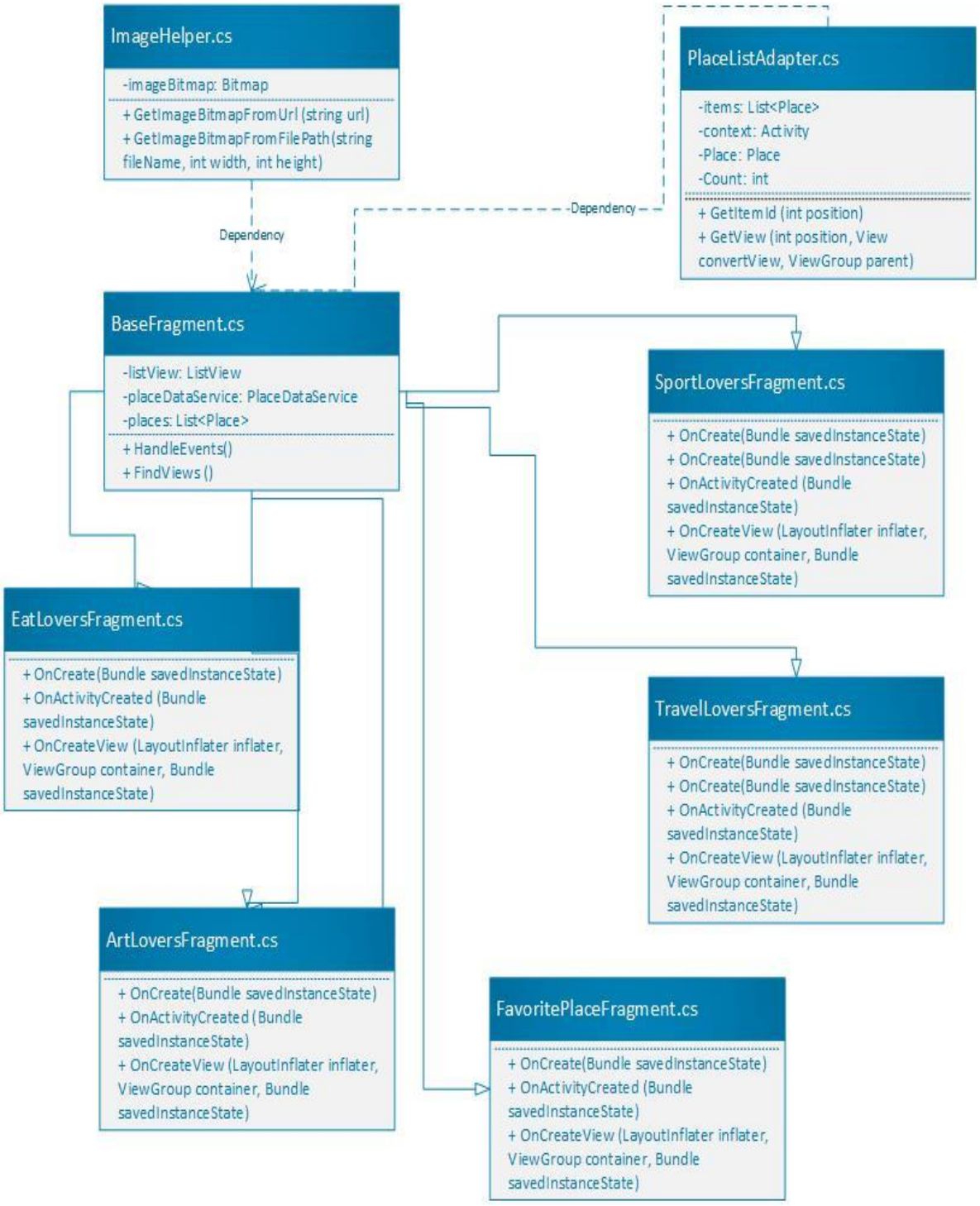


Рисунок 3.4 – Діаграма класів для проекту Touristsobjects.Android(частина 1)

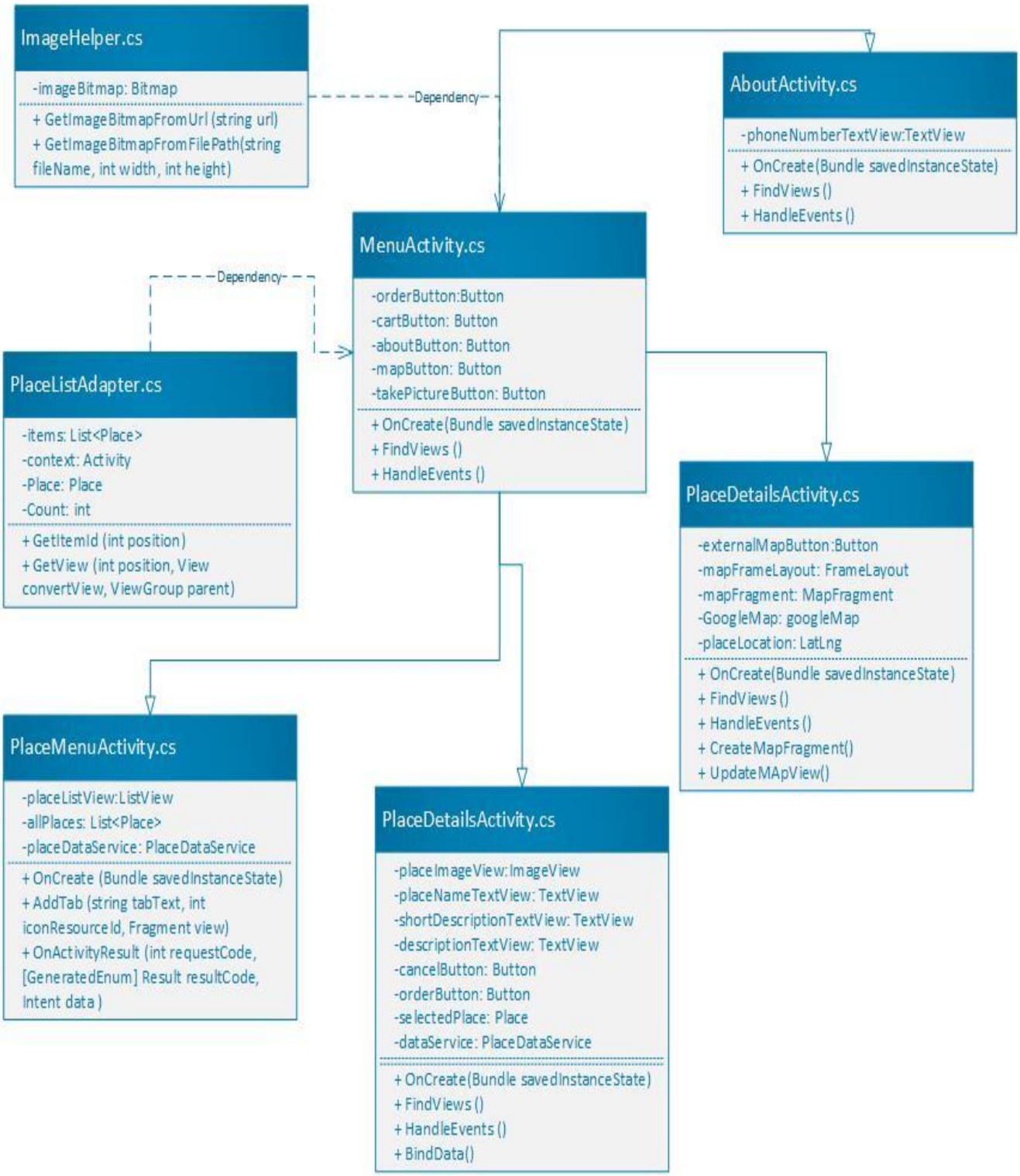


Рисунок 3.5 – Діаграма класів для проекту TouristsObjects.Android (частина 2)

3.5 Розробка ER-моделі

В даній системі використовується реляційна модель даних (РМД). Відповідно до реляційної моделі база даних подається у вигляді сукупності

таблиць, над якими можуть виконуватися операції. В реляційній моделі операції над об’єктами бази даних мають теоретико-множинний характер. Відношенням називається деяка сукупність об’єктів, яка характеризується однаковим набором атрибутів. Зручно зображувати відношення як таблицю, де кожний рядок є кортеж і кожний стовпець є атрибут. Стовпці таблиці – це елементи даних, а рядки – записи.ER – модель характеризує нашу БД. В базі даних зберігаються об’єкти та коментарі. Оскільки користувачі можуть залишати коментарі не зареєстрованими, то нам не потрібно ще однієї таблиці для цього. ER – модель даної системи зображена на рисунку 3.6:

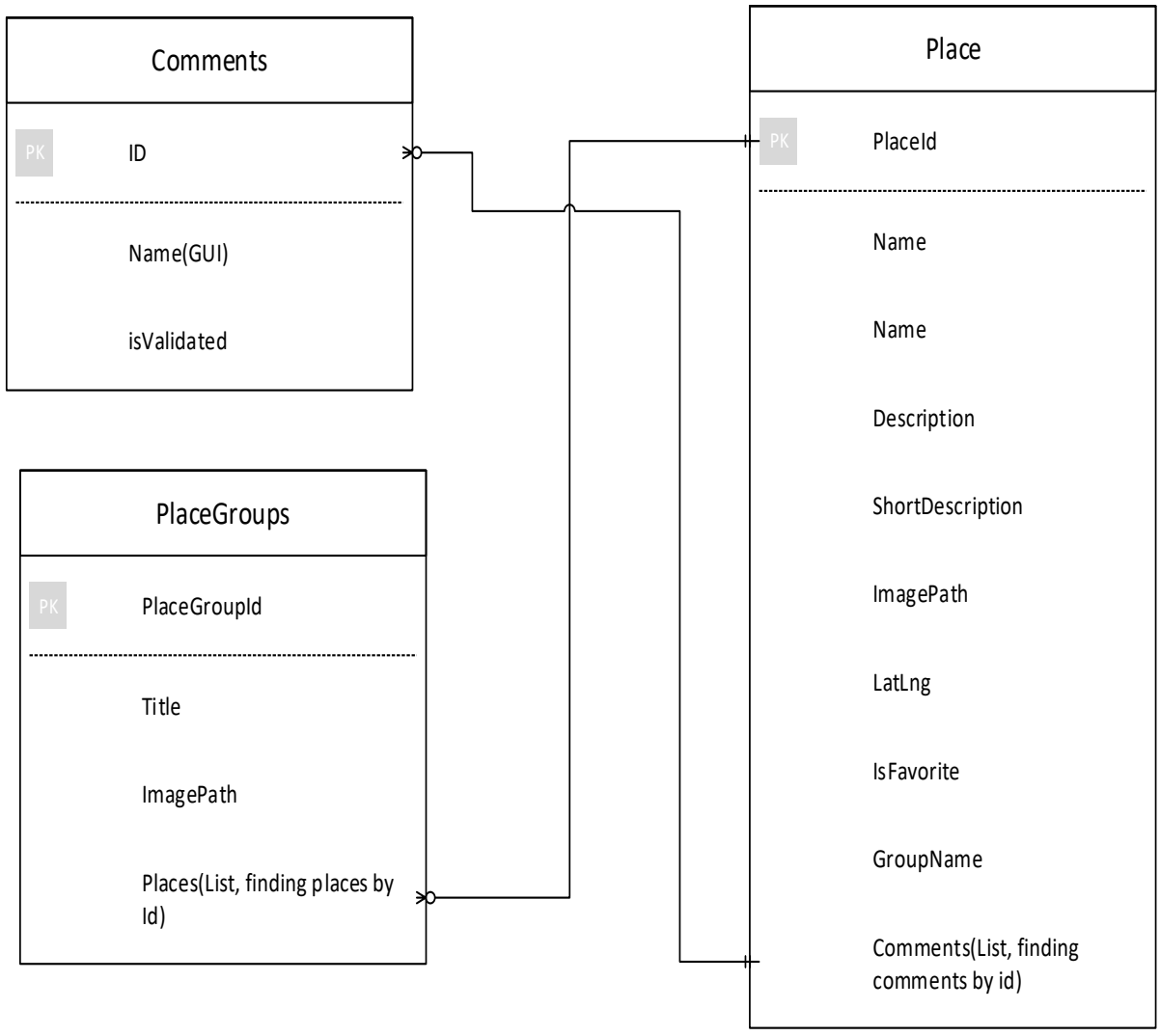


Рисунок 3.6 – ER – модель розробленої системи

3.6 Тестування програмного забезпечення

Складові програмного забезпечення, розроблені у вигляді мобільного додатку, були протестовані на мобільному телефоні з ОС Андроїд. Результати тестування наведені нижче на рис. 3.7 – 3.15.

Тестування навігація системи показано на рис. 3.7.

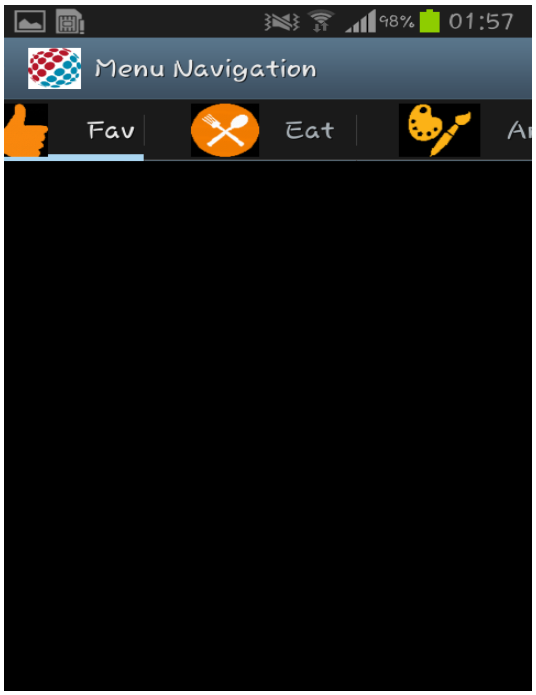


Рисунок 3.7 –Меню навігації системи

Навігацію системи з внутрішніми об’єктами зображено на рис. 3.8-3.9:

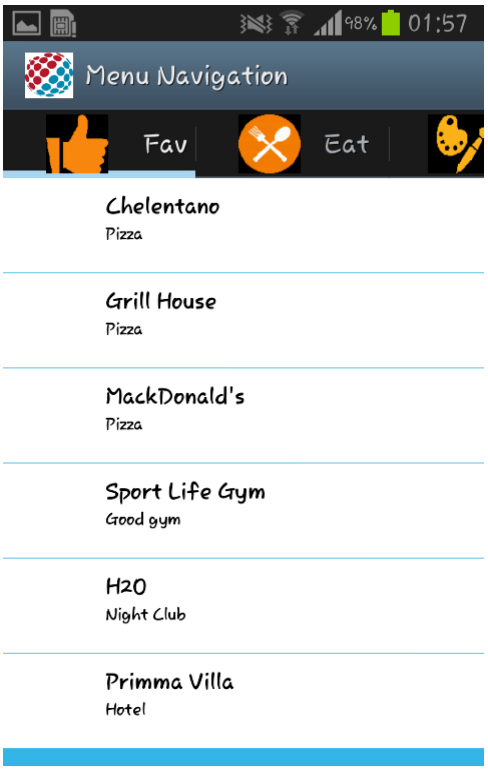


Рисунок 3.8 – Навігація системи з внутрішніми об’єктами

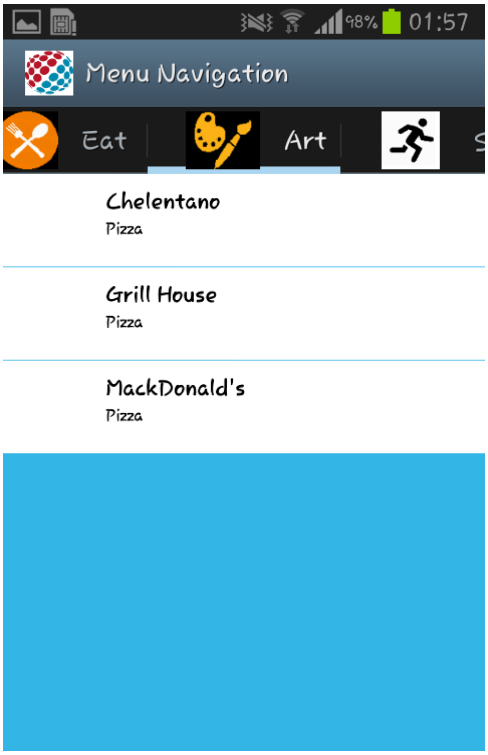


Рисунок 3.9 – Навігація системи з внутрішніми об’єктами

Структура проекту зображена на рисунку 3.10:

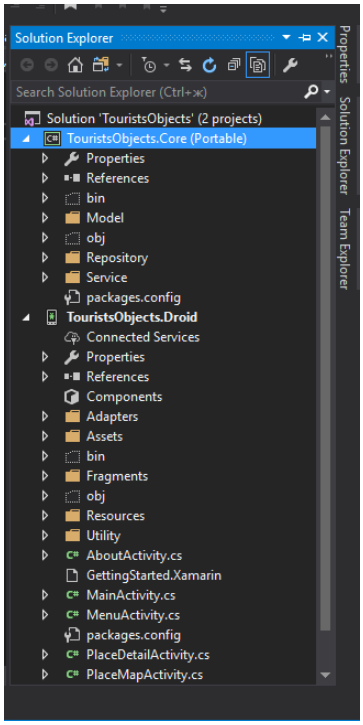


Рисунок 3.10 – Структура проекту

Тестування Google Maps, Google Maps Api зображена на рисунку 3.11:

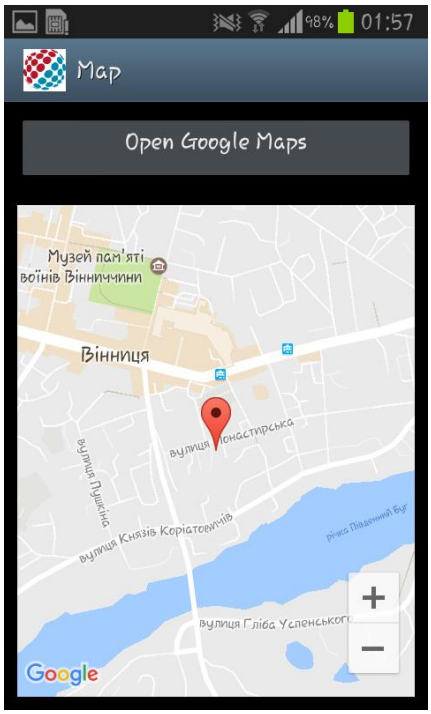


Рисунок 3.11 – Розробка карт

Виклик нативних Google Maps при натисненні на клавішу “Open Google Maps” зображено на рисунку 3.12:

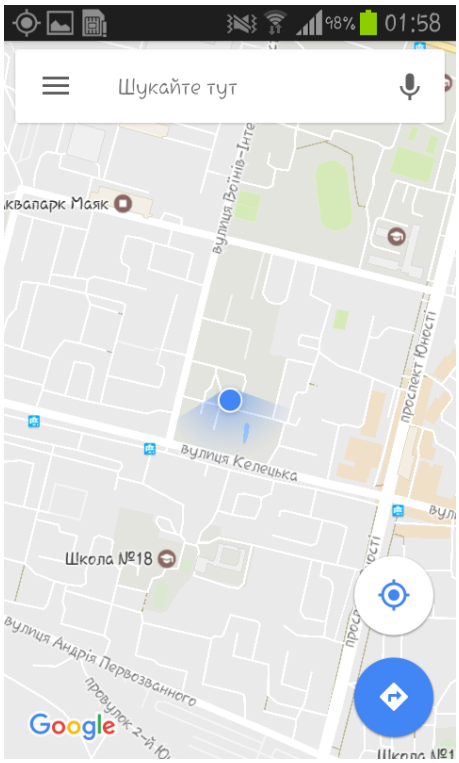


Рисунок 3.12 – Виклик нативних Google Maps

Головне меню програми зображено на рисунку 3.13:

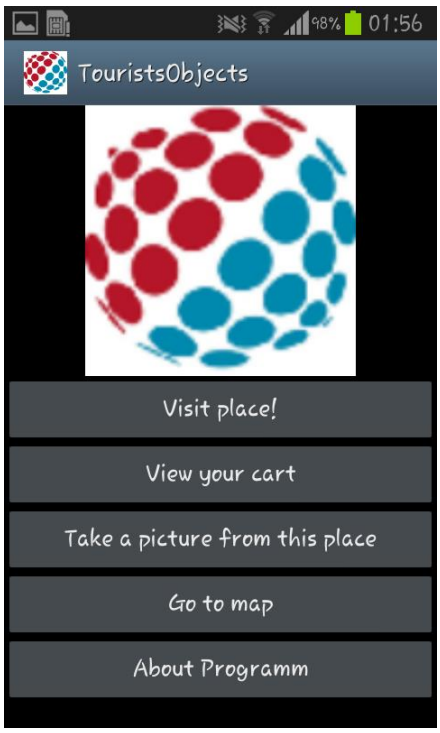


Рисунок 3.13 – Головне меню програми

На рисунку 3.14 зображене вікно коментарів:

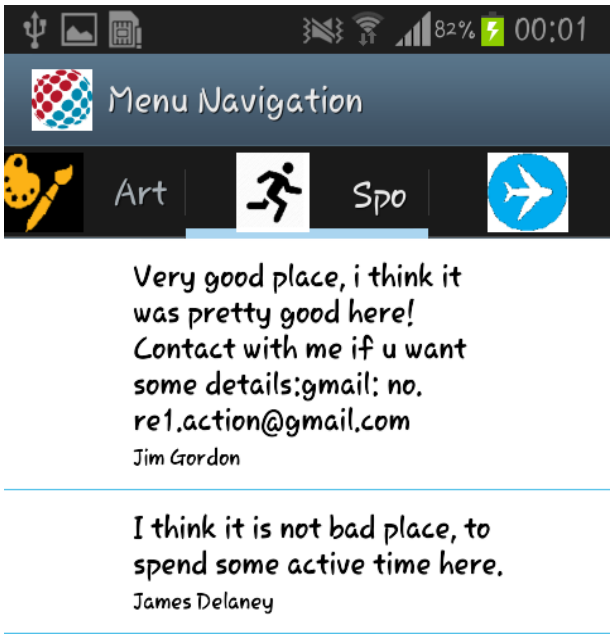


Рисунок 3.14 – Коментар

Лістинг програми наведений в додатку В.

3.7 Висновки до розділу

В даному розділі в якості програмної платформи обрано Visual Studio 2019, а для зберігання даних була обрана СУБД SQL Server 2017 і ORM Entitty Framework від Microsoft, які мають тісну інтеграцію з іншими продуктами цієї фірми. Для розробки системи обрані технологія Xamarin і мова C#, що є найпопулярнішою мовою для написання бізнес логіки на платформі .NET. Розроблена схема алгоритму програми, діаграми класів для проекту, ER-модель бази даних. Написана програма роботи системи автоматизованого пошуку туристичних маршрутів мовою C#.

Результати проведеного тестування програмного забезпечення системи на мобільному телефоні з ОС Android підтвердили практичну працездатність використаних технологій і розробленого програмного забезпечення.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Розробка системи автоматизованого вибору туристичних об’єктів» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка системи автоматизованого вибору туристичних об’єктів» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [52].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно	Технічні та споживчі влас-тивості продукту трохи гірші, ніж	Технічні та споживчі властивості продукту на рівні	Технічні та споживчі властивості продукту трохи	Технічні та споживчі властивості продукту значно
5	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в	Експлуатаційні витрати на рівні експлуатаційних	Експлуатаційні витрати трохи нижчі, ніж в	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
6	Ринок малий і не має позитивної	Ринок малий, але має позитивну	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною
7	Активна конкуренція великих	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Таблиця 4.1 (Продовження)

9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуютьс
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	4	4
2. Ринкові переваги (наявність аналогів)	2	3	3

Таблиця 4.2 (Продовження)

3. Ринкові переваги (ціна продукту)	2	2	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	2	2	2
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	34	37	37
Середньоарифметична сума балів $СБ_c$	36,0		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [52].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ ,розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка системи автоматизованого вибору туристичних об’єктів»

становить 36,0 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою [52]:

$$q_i = \frac{P_i}{P_{базі}}. \quad (4.1)$$

де q_i – одиничний параметричний індекс, розрахований за i-м параметром;

P_i – значення i-го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Об'єм памяті дистрибутивів для запуску серверу (на сервері)	МБ	640	400	1,6	0,2

Таблиця 4.4 (Продовження)

Об'єм оперативної пам'яті	МБ	310	150	2,1	0,25
Кількість таблиць бази даних	шт	75	33	2,27	0,2
Об'єм бази даних	ГБ	32	64	2	0,1
Кількість мікросервісів	шт	15	4	3,75	0,25
Експлуатаційні витрати	грн.	500	320	0,64	0,5
Запланована вартість підписки (доступу до ресурсу)	грн.	70	78	1,11	0,5

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [51]:

$$I_{НП} = \prod_{i=1}^{n} q_i \ , \tag{4.2}$$

де $I_{нп}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i-м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{нп} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [51]:

$$I_{ТП} = \sum_{i=1}^{n} q_i \cdot \alpha_i \,, \tag{4.3}$$

де $I_{тп}$ – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i-го параметра;

α_i – вагомість i-го параметричного показника, $\sum_{i=1}^{n} \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{mn} = 1{,}6 \cdot 0{,}2 + 2{,}1 \cdot 0{,}25 + 2{,}27 \cdot 0{,}2 + 2 \cdot 0{,}1 + 3{,}75 \cdot 0{,}25 = 2{,}44.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [51]:

$$I_{ЕП} = \sum_{i=1}^{m} q_i \cdot \beta_i \,, \tag{4.4}$$

де $I_{ЕП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i-го виду;

β_i – частка i-го економічного параметра, $\sum_{i=1}^{m} \beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання. Проведемо аналіз параметрів згідно даних таблиці .

$$I_{ЕП} = 0{,}64 \cdot 0{,}5 + 1{,}11 \cdot 0{,}5 = 0{,}88.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [51]**:**

$$K_{ІНТ} = I_{НП} \cdot \frac{I_{ТП}}{I_{ЕП}}, \tag{4.5}$$

$$K_{ІНТ} = 1 \cdot 2{,}44 / 0{,}88 = 2{,}78.$$

Інтегральний показник конкурентоспроможності $К_{ІНТ} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Розробка системи автоматизованого вибору туристичних об’єктів», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [52]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=22 дні.

$$З_о = 16500{,}00 \cdot 18 / 22 = 13500{,}00 \; грн.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	16500,00	750,00	18	13500,00
Інженер-розробник автоматизованих інформаційних систем	15300,00	695,45	18	12518,18
Інженер-розробник програмного забезпечення	15350,00	697,73	12	8372,73
Консультант (менеджер-аналітик туристичного бізнесу)	20000,00	909,09	4	3636,36
Технік	8200,00	372,73	12	4472,73
Всього				42500,00

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Розробка системи автоматизованого вибору туристичних об’єктів» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \quad (4.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.8)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо *Мм*=8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [52];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 22 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 8000{,}00 \cdot 1{,}10 \cdot 1{,}15 / (22 \cdot 8) = 57{,}50$ *грн.*

$З_{р1} = 57{,}50 \cdot 8{,}20 = 471{,}50$ *грн.*

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка обладнання для моделювання та проектування інформаційної системи	8,20	2	1,10	57,50	471,50

Таблиця 4.6 (Продовження)

Підготовка робочого місця розробника програмного забезпечення	4,70	3	1,35	70,57	331,67
Підготовка серверного обладнання	3,20	4	1,50	78,41	250,91
Інсталяція програмного забезпечення для моделювання та розробки інформаційної системи	6,22	4	1,50	78,41	487,70
Компіляція програмних блоків	8,00	5	1,70	88,86	710,91
Налагодження програмних блоків	4,60	6	2,00	104,55	480,91
Тестування системи	10,00	2	1,10	57,50	575,00
Всього					3308,60

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.9)$$

де $Н_{дод}$ =11% – норма нарахування додаткової заробітної плати.

$З_{дод} = (42500,00 + 3308,60) \cdot 11 / 100\% = 5038,95$ *грн.*

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.10)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (42500,00 + 3308,60 + 5038,95) · 22 / 100% = 11186,46 грн.

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка системи автоматизованого вибору туристичних об’єктів».

Витрати на матеріали ($М$), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (4.11)$$

де $Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

$М_1$ = 3 · 223,00 · 1,02 - 0 · 0 = 682,38 грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір Cristal A4 500	223,00	3	0	0	682,38
Папір для записів Cristal LightPapers 65 A5	100,00	4	0	0	408,00
Органайзер офісний Cristal	195,00	4	0	0	795,60
Набір офісний Cristal Base	204,00	3	0	0	624,24
Картридж для принтера Canon LBP5000	1420,00	1	0	0	1448,40
Диск оптичний CD-R	23,00	2	0	0	46,92
Flesh-пам'ять Kingston 128 GB 10	410,00	1	0	0	418,20
Тека для паперів	72,00	6	0	0	440,64
Інші матеріали	140,00	1	0	0	142,80
Всього					5007,18

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Розробка системи автоматизованого вибору туристичних об'єктів», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (4.12)$$

де $Н_j$ – кількість комплектуючих *j*-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих *j*-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ *= 1,1 ... 1,15*).

$К_в = 1 \cdot 4250{,}00 \cdot 1{,}02 = 4335{,}00$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
RouterBOARD TP-Link230	1	4250,00	4335,00
База даних (тестувальна) БД64-А10	1	3100,00	3162,00
Всього			7497,00

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$В_{спец} = \sum_{i=1}^{k} Ц_i \cdot С_{пр.i} \cdot К_i \ , \qquad (4.13)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.і}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$К_і$ – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($К_і$ = *1,10…1,12);*

k – кількість найменувань устаткування.

$В_{спец}$ = *43599,00 · 1 · 1,02 = 44470,98 грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання на основі EOM DELUX F43-A71BC	1	43599,00	44470,98
Всього			44470,98

4.3.6 Програмне забезпечення для наукових робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.і} \cdot K_i \ , \qquad (4.14)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_і$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_і$ = *1,10…1,12);*

k – кількість найменувань програмних засобів.

$B_{прг} = 650{,}00 \cdot 1 \cdot 1{,}02 = 663{,}00$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Доступ (підписка) для вивчення аналогу системи для підбору туристичних об'єктів	1	650,00	663,00
Емулятор серверної інтернет-платформи для моделювання поведінки автоматизованого інформаційного ресурсу	1	6520,00	6650,40
Visual Studio 2019(Framework Xamarin)	1	6590,00	6721,80
SQL Server 2017	1	45,00	45,90
ORM Entitty від Microsoft	0	4210,00	0,00
Всього			14081,10

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.15)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (32599,00 \cdot 1) / (3 \cdot 12) = 905,53$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер	32599,00	3	1	905,53
Обчислювально-аналітична система програмної розробки	42300,00	3	1	1175,00
Ноутбук HP Laptop 15s-eq2037ua Natural Silver (422G7EA)	34800,00	3	1	966,67
Робоче місце розробника програмного забезпечення	7200,00	5	1	120,00
Пристрій виводу інформації HP-2100	6650,00	4	1	138,54
Оргтехніка	9350,00	5	1	155,83
Приміщення лабораторії	425000,00	30	1	1180,56

Таблиця 4.11 (Продовження)

ОС Windows 11	6520,00	2	1	271,67
Прикладний пакет Microsoft Office 2019	6600,00	2	1	275,00
Прикладний пакет розробки інформаційних систем (мови Java, C#, C++, Perl, PHP)	8350,00	2	1	347,92
Всього				5536,71

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (4.16)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 10,98 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,02 \cdot 90,0 \cdot 10,98 \cdot 0,95 / 0,97 = 19,76$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
RouterBOARD TP-Link230	0,02	90,0	19,76
Серверне обладнання на основі EOM DELUX F43-A71BC	0,56	90,0	553,39
Персональний комп'ютер	0,25	90,0	247,05
Обчислювально-аналітична система програмної розробки	0,32	90,0	316,22
Ноутбук HP Laptop 15s-eq2037ua Natural Silver (422G7EA)	0,04	90,0	39,53
Робоче місце розробника програмного забезпечення	0,10	70,0	76,86
Пристрій виводу інформації HP-2100	0,06	4,5	2,96
Оргтехніка	0,60	1,1	7,25
Всього			1263,03

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка системи автоматизованого вибору туристичних об’єктів» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов’язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з’їзди, конференції, наради, пов’язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» відсутні.

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (4.17)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$B_{сп}$ = (42500,00 + 3308,60) · 30 / 100% = 13742,58 грн.

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$І_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.18)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$І_в$ = (42500,00 + 3308,60) · 50 / 100% = 22904,30 грн.

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків;

витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.19)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$B_{нзв}$ = (42500,00 + 3308,60) · 100 / 100% = 45808,60 грн.

Витрати на проведення науково-дослідної роботи на тему «Розробка системи автоматизованого вибору туристичних об'єктів» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + І_в + B_{нзв}. \quad (4.20)$$

$B_{заг}$= 42500,00 +3308,60 +5038,95 +11186,46 +5007,18 +7497,00 +44470,98 + 14081,10 + 5536,71 +1263,03 +0,00 +13742,58 +22904,30 +45808,60 = 222345,49 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.21)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 222345,49 / 0,9 = 247050,55 грн.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Розробка системи автоматизованого вибору туристичних об’єктів» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	10000	10000	10000	8000

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 100000 осіб;

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 42,00 грн;

$\pm\Delta Ц_о$ – зміна вартості послуги від впровадження результатів, приймемо 10,33 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [52]**:**

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.22)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 38\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (10{,}33 \cdot 100000{,}00 + 52{,}33 \cdot 10000) \cdot 0{,}83 \cdot 0{,}38 \cdot (1\text{-}0{,}18/100\%) = 402588{,}10$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (10{,}33 \cdot 100000{,}00 + 52{,}33 \cdot 20000) \cdot 0{,}83 \cdot 0{,}38 \cdot (1\text{-}0{,}18/100\%) = 537935{,}89$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (10{,}33 \cdot 100000{,}00 + 52{,}33 \cdot 30000) \cdot 0{,}83 \cdot 0{,}38 \cdot (1\text{-}0{,}18/100\%) = 673283{,}69$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (10{,}33 \cdot 100000{,}00 + 52{,}33 \cdot 38000) \cdot 0{,}83 \cdot 0{,}38 \cdot (1\text{-}0{,}18/100\%) = 781561{,}92$ грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.23)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу в роках, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП$ =402588,10/(1+0,15)[1]+537935,89/(1+0,15)[2]+673283,69/(1+0,15)[3]+
+781561,92/(1+0,15)[4]=350076,61+406756,82+442694,95+446860,56=1646388,9
5 грн.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \quad (4.24)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 247050,55 грн.

$PV = k_{інв} \cdot ЗВ$ = 2 · 247050,55 = 494101,09 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.25)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1646388,95 грн;

PV – теперішня вартість початкових інвестицій, 494101,09 грн.

$E_{абс} = ПП - PV$ = 1646388,95 - 494101,09 = 1152287,86 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.26)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 1152287,86 грн;

PV – теперішня вартість початкових інвестицій, 494101,09 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+1152287,86/494101,09)^{1/4} = 0,35.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.27)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні d =0,12;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,15.

$\tau_{мін}$ = 0,12+0,15 = 0,27 < 0,35 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка системи автоматизованого вибору туристичних об’єктів» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \quad (4.28)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,35 = 2,85 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка системи автоматизованого вибору туристичних об’єктів» становить 36,0 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,44 рази.

Також термін окупності становить 2,85 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка системи автоматизованого вибору туристичних об’єктів».

ВИСНОВКИ

В магістерській кваліфікаційній роботі розроблена система автоматизованого вибору туристичних об’єктів, реалізована як програмний додаток до мобільного телефону. Система надає зручний сервіс гостям і жителям міста, забезпечуючи узагальнену інформацію про туристичні об’єкти і відгуки про них інших користувачів, показує маршрути для відвідування наявних об’єктів

Проведено аналіз варіантів існуючих систем пошуку туристичних маршрутів та визначено ключові переваги та недоліки наявних рішень. Було проаналізовано вимоги до сучасних систем автоматизованого вибору туристичних маршрутів, їх переваги та недоліки і запропоновано свій оптимальний варіант.

Розроблено алгоритм на основі мурашиного методу, який дозволяє ефективно обробляти велику кількість даних та враховує різні фактори, такі як популярність об'єктів, відгуки користувачів, географічні обмеження та погодні умови. Реалізовано програмне забезпечення, яке інтегрується з картографічними сервісами та надає користувачам оптимальні маршрути.

Для розробки системи обрані програмна платформа Visual Studio 19, технологія Xamarin і мова C#, що є найпопулярнішою мовою для написання бізнес-логіки на платформі .NET. Розроблена схема алгоритму програми, діаграми класів для проекту, ER-модель бази даних. Написана програма роботи системи автоматизованого пошуку туристичних маршрутів мовою C#. Проведено тестування програмного забезпечення системи на мобільному телефоні з ОС Android, результати якого підтвердили практичну працездатність використаних технологій і розробленого программного забезпечення.

Проведено дослідження і розрахунки, які показали що рівень комерційного потенціалу розробки становить 36,0 бала, Це свідчить про комерційну важливість проведених в роботі досліджень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ковальчук Ю. М. Аналіз систем автоматизованого пошуку туристичних маршрутів // Вісник Одеського національного університету. Серія: Комп’ютерні науки. – 2019. – Вип. 6. – С. 34–42.

2. Паламар І. М. Сучасні підходи до розробки туристичних додатків // Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. – 2017. – Вип. 879. – С. 112–120.

3. Бондаренко М. В., Савченко Т. В. Інформаційні системи для підтримки туристичної діяльності // Наукові праці ЧНУ ім. Ю. Федьковича. – 2018. – Вип. 295. – С. 34–42.

4. Державна служба статистики України. Офіційний вебсайт [Електронний ресурс]. – Режим доступу: http://www.ukrstat.gov.ua.

5. Кравченко І. В. Розробка автоматизованих систем туристичного консультування // Наукові праці ХНУРЕ. – 2018. – Вип. 11. – С. 58–65.

6. Мельник Н. О. Персоналізовані рекомендаційні системи для туристів // Наукові праці ХНУРЕ. – 2020. – Вип. 13. – С. 72–80.

7. Nasir H. M., Hussain S. Development of tourism management systems using artificial intelligence // Journal of Tourism Research & Hospitality. – 2020. – Vol. 9, no. 1. – Pp. 24–35.

8. Luo R. C., Chiang H. K., Chan T. T. Intelligent travel route recommendation based on personal preferences // IEEE Access. – 2018. – Vol. 6. – Pp. 69680–69690.

9. Іванов В. В. Використання AI для розробки мобільних туристичних додатків // Вісник ЧДТУ. Серія: Інформаційні технології. – 2019. – Вип. 13. – С. 44–51.

10. Олійник С. В. Системи підтримки прийняття рішень у туризмі на базі мобільних додатків // Вісник ХНУРЕ. – 2019. – Вип. 29. – С. 33–40.

11. Letchmunan S. K., Lu Y., Wong Y. W. A location-based tour recommendation system using mobile devices // Journal of Ambient Intelligence and Humanized Computing. – 2019. – Vol. 10, no. 6. – Pp. 2315–2330.

12. Jain A. K., Gadre M. M. Personalized mobile tourism recommender systems: A survey // International Journal of Computer Applications. – 2017. – Vol. 174, no. 2. – Pp. 1–7.

13. Cho J., Choi H. G., Park S. B. Developing a tourism route recommendation system by machine learning // IEEE International Conference on Big Data and Smart Computing (BigComp). – Kyoto, Japan. – 2020. – Pp. 452–456.

14. de Almeida S. A. L., Ferreira J., Rodrigues A., dos Santos P. M. A personalized tourist route recommendation system // Journal of Information and Data Management. – 2019. – Vol. 10, no. 2. – Pp. 101–114.

15. Kim K. M., Cho H., Lee J. H. Mobile tour guide recommendation system using clustering techniques // IEEE Transactions on Consumer Electronics. – 2019. – Vol. 65, no. 3. – Pp. 233–242.

16. Чумак О. І. Розробка мобільного додатку для персоналізації туристичних маршрутів // Науковий вісник НТУУ «КПІ». Серія: Інформаційні технології. – 2020. – Вип. 25. – С. 99–106.

17. Глушков С. Г. Мобільні додатки в сучасній туристичній діяльності // Туризм і гостинність: науковий журнал. – 2016. – Вип. 12. – С. 22–29.

18. Рудько А. О. Мобільні додатки для туристичної індустрії: аналіз можливостей // Журнал «Цифрові технології в освіті та науці». – 2020. – Вип. 45. – С. 98–105.

19. Лісова О. М. Системи рекомендацій для туристичних маршрутів на базі мобільних платформ // Науковий вісник НТУУ «КПІ». Серія: Інформаційні системи. – 2017. – Вип. 22. – С. 109–117.

20. Krueger D. W., Dhillon K. K. Smart tourism and machine learning-based route planners // IEEE International Conference on Machine Learning and Applications (ICMLA). – Orlando, USA. – 2020. – Pp. 652–658.

21. Tsai C. C., Chen L. H. A mobile tourist guide recommendation system for smart tourism // International Journal of Electronic Business. – 2019. – Vol. 14, no. 3. – Pp. 101–122.

22. Lin B. Y., Chiu C. Y., Wang H. Y., Wong S. K. A mobile recommendation system for planning smart travel routes // Journal of Systems and Software. – 2019. – Vol. 157. – Pp. 1–12.

23. Turk M. O. Mobile computing and tourism route personalization using AI // Journal of Applied Computing and Artificial Intelligence. – 2021. – Vol. 17, no. 3. – Pp. 121–134.

24. Скибінський А. М. Застосування нейронних мереж у туризмі для побудови маршрутів // Вісник Львівського університету. Серія: Прикладна математика. – 2018. – Вип. 34. – С. 55–61.

25. Грищенко Л. І. Інновації у мобільних додатках для туризму // Технології та інновації у туризмі. – 2021. – Вип. 7. – С. 42–49.

26. Ковтун Т. С. Туристичні мобільні сервіси: інноваційні підходи до розробки // Вісник Національного університету «Львівська політехніка». – 2021. – Вип. 920. – С. 110–118.

27. Лук’яненко В. О. Мобільні рішення для персоналізованих туристичних маршрутів // Науковий вісник ОНПУ. – 2020. – Вип. 32. – С. 79–87.

28. Weiler M., Floyd R. W., Keith G. W. Mobile travel apps: An opportunity for tourism innovation // Tourism Management. – 2018. – Vol. 70. – Pp. 112–120.

29. Pham C., Zhou S. Mobile recommender system for tourists // ACM Transactions on Interactive Intelligent Systems (TiiS). – 2020. – Vol. 10, no. 1. – Pp. 1–28.

30. McCarthy M. C. Augmented reality mobile applications for tourism route planning // International Journal of Mobile Human Computer Interaction. – 2019. – Vol. 11, no. 2. – Pp. 45–65.

31. Xu L. L., Li X. Z., Zhang P. Y. Big data-driven route optimization for smart tourism // IEEE Access. – 2020. – Vol. 8. – Pp. 37224–37235.

32. Xamarin Android [Електронний ресурс]. – Режим доступу: https://developer.xamarin.com/guides/android/getting_started/. – Назва з екрану.

33. Smithwick M., Verma M. Pro OpenGL ES for Android. – Apress, 2012. – 308 p.

34. Hillegass A., Ward M. Objective-C Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides). – 2013. – 312 p.

35. Android Api Levels [Електронний ресурс]. – Режим доступу: https://developer.xamarin.com/guides/android/application_fundamentals/understanding_android_api_levels/. – Назва з екрану.

36. Using the Xamarin Android SDK Manager [Електронний ресурс]. – Режим доступу: https://developer.xamarin.com/guides/android/application_fundamentals/using-the-sdk-manager/. – Назва з екрану.

37. 9 things to know about Google's maps data: Beyond the Map [Електронний ресурс]. – 2019. – Режим доступу: https://mapsplatform.google.com/resources/blog/9-things-know-about-googles-maps-data-beyond-map/. – Назва з екрану.

38. Android Api Levels - [електронний ресурс]– Режим доступу: https://developer.xamarin.com/guides/android/application_fundamentals/understanding_android_api_levels/ .

39. Graber M. Mastering SQL. – Lori, 2007. – 672 p.

40. Weinberg P. N., Groff J. R. SQL: The Complete Reference. – McGraw-Hill, Berkeley, 2016. – 994 p.

41. Мулеса О. Ю. Основи мови запитів SQL. – Ужгород, 2015. – 48 с.

42. Papadimitriou C. H., Steiglitz K. Combinatorial Optimization: Algorithms and Complexity. – Englewood Cliffs, New Jersey, Prentice-Hall Inc., 2011. – 510 p.

43. Гуляницький Л. Ф., Мулеса О. Ю. Прикладні методи комбінаторної оптимізації. – К.: ВПЦ "Київський університет", 2016. – 142 с.

44. Штовба С. Д., Рудий О. М. Мурашині алгоритми оптимізації // Вісник ВПІ. – 2004. – № 4. – С. 62–69.

45. Литвин В. В., Угрин Д. І. Методика вирішення завдань пошуку оптимальних туристичних маршрутів алгоритмами наслідування мурашиної колонії // Вісник ХПІ. – 2016. – № 21. – С. 47–60.

46. Юзевич В., Крап Н. Моделювання туристичних потоків з використанням мурашиних алгоритмів // Вісник Національного університету «Львівська політехніка». – 2011. – С. 98–103.

47. Visual Studio 19 [Електронний ресурс]. – Режим доступу: https://ko.com.ua/microsoft_vypustila_visual_studio_2019_128386. – Назва з екрану.

48. Встановлення Visual Studio 19 [Електронний ресурс]. – Режим доступу: https://www.youtube.com/watch?v=1t0WqbLzOcc. – Назва з екрану.

49. Reynolds M. Xamarin Mobile Application Development for Android. – Packt Publishing Ltd., 2014. – 145 p.

50. Peppers J. Xamarin Cross-platform Application Development. – Packt Publishing Ltd., 2014. – 239 p.

51. Kelly S., Tolvanen J.-P. Domain-Specific Modeling: Enabling Full Code Generation. – John Wiley & Sons, Inc., Hoboken, 2008. – 316 p.

52. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. – Вінниця: ВНТУ, 2021. – 42 с.

ДОДАТКИ

Додаток А
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: « Розробка системи автоматизованого вибору туристичних об'єктів »

Тип роботи: магістерська кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))

Підрозділ кафедра комп'ютерних систем управління

(кафедра, факультет (інститут), навчальна група)

Керівник Биков М.М., професор кафедри КСУ

(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	91%
Загальна схожість	9%

Аналіз звіту подібності (відмітити потрібне)

√ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Мирослава ТЕСЛЮК

(підпис) (прізвище, ініціали)

Опис прийнятого рішення

_____

_____

_____

_____

Особа, відповідальна за перевірку _____ Володимир ДУБОВОЙ

(підпис) (прізвище, ініціали)

Експерт _____ _____

(за потреби) (підпис) (прізвище, ініціали, посада)

Додаток Б
(обов'язковий)

ВНТУ

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., проф.

_____ Вячеслав КОВТУН

"14" 10 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Розробка системи автоматизованого вибору туристичних об'єктів

08-33.МКР.011.00.000 ТЗ

Студентка групи 2АКІТР-23м

_____ Мирослава ТЕСЛЮК
Підпис *Ім'я ПРІЗВИЩЕ*

Керівник проф. кафедри КСУ

_____ Микола БИКОВ
Підпис *Ім'я ПРІЗВИЩЕ*

Вінниця 2024

1. Назва та галузь застосування

1.1. Назва – Розробка системи автоматизованого вибору туристичних об’єктів.

1.2. Галузь застосування – Автоматизація, ком’ютерно-інтегровані технології та робототехніка.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ №310 від 17 вересня 2024 р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є підвищення ефективності процесу вибору та відвідування туристичних об’єктів для туристів та жителів міста для спрощення вибор місця відпочинку.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Ю. М. Ковальчук, "Аналіз систем автоматизованого пошуку туристичних маршрутів", Вісник Одеського національного університету, серія «Комп’ютерні науки», вип. 6, 2019, с. 34–42.
2. О. І. Чумак, "Розробка мобільного додатку для персоналізації туристичних маршрутів", Науковий вісник НТУУ «КПІ», серія «Інформаційні технології», вип. 25, 2020, с. 99–106.
3. M. Nasir and S. Hussain, "Development of tourism management systems using artificial intelligence", Journal of Tourism Research & Hospitality, vol. 9, no. 1, pp. 24-35, 2020.
4. Мулеса О.Ю. Основи мови запитів SQL. – Ужгород, 2015. –48 с.
5. C.H. Papadimitriou, K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity, Englewood Cliffs, New Jersey, Prentice-Hall Inc, 2011. - 510 p.
6. Штовба С. Д. Мурашині алгоритми оптимізації [Текст] / С. Д. Штовба, О. М. Рудий // Вісник Вінницького політехнічного інституту. - 2004. - № 4. - С. 62-69..

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- можливість використання карт та GPS навігації;

- можливість залишати коментарі на основі відвідання об’єктів;
- автоматичний вибір туристичного об’єкту;
- пошук та представлення у зручному для користувача форматі об’єктів для відвідання та відпочинку;
- автоматичний вибір кращих об’єктів на основі мурашиного алгоритму.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- Mobile device;
- Visual Studio 2019;
- Xamarin.

5.2.2. Умови експлуатації системи:

- робота на мобільних та веб-додатках;
- можливість цілодобового функціонування системи;
- дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

- Багатоваріантний аналіз способів для побудови системи автоматизованого вибору туристичних об’єктів, постановка задач. «07» 09 2024 р.
- Розробка технічного завдання «15» 09 2024 р.
- Розробка архітектури системи «17» 09 2024 р.
- Вибір технологій розробки системи «25» 09 2024 р.
- Розробка програмного забезпечення системи «03» 11 2024 р.
- Тестування програмного забезпечення «16» 11 2024 р.
- Оформлення пояснювальної записки «20» 11 2024 р.

6.2 Графічні матеріали:

- Розробка UML-діаграм системи «25» 11 2024 р.
- Розробка ER-моделі бази даних системи «28» 11 2024 р.
- Розробка демонстраційних плакатів «30» 11 2024 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «26» 11 2024 р.

7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист роботи провести до «01» 12 2024 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ДЕК. Захист бакалаврської дипломної роботи провести до «09» 12 2024 р.

Додаток В

(довідковий)

Лістинги програм

1. MenuActivity.cs

```
[Activity(Label = "TouristsObjects", MainLauncher = true)]
public class MenuActivity : Activity
{
    private Button orderButton;
    private Button cartButton;
    private Button aboutButton;
    private Button mapButton;
    private Button takePictureButton;

    protected override void OnCreate(Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);

        SetContentView(Resource.Layout.MainMenu);

        FindViews();

        HandleEvents();
    }

    private void FindViews()
    {
        orderButton = FindViewById<Button>(Resource.Id.orderButton);
        cartButton = FindViewById<Button>(Resource.Id.cartButton);
        aboutButton = FindViewById<Button>(Resource.Id.aboutButton);
        mapButton = FindViewById<Button>(Resource.Id.mapButton);
        takePictureButton = FindViewById<Button>(Resource.Id.takePictureButton);
    }

    private void HandleEvents()
    {
        orderButton.Click += OrderButton_Click;
        aboutButton.Click += AboutButton_Click;
        takePictureButton.Click += TakePictureButton_Click;
        mapButton.Click += MapButton_Click;
    }

    private void TakePictureButton_Click(object sender, EventArgs e)
    {
        var intent = new Intent(this, typeof(TakePictureActivity));
        StartActivity(intent);
    }

    private void AboutButton_Click(object sender, EventArgs e)
    {
        var intent = new Intent(this, typeof(AboutActivity));
        StartActivity(intent);
    }

    private void OrderButton_Click(object sender, EventArgs e)
    {
        var intent = new Intent(this, typeof(PlaceMenuActivity));
        StartActivity(intent);
```

```
        }

        private void MapButton_Click(object sender, EventArgs e)
        {
            var intent = new Intent(this, typeof(PlaceMapActivity));
            StartActivity(intent);
        }
     }
```

2. PlaceDetailActivity.cs

```
[Activity(Label = "Place Details", MainLauncher = false)]
    public class PlaceDetailsActivity : Activity
    {
        private ImageView placeImageView;
        private TextView placeNameTextView;
        private TextView shortDescriptionTextView;
        private TextView descriptionTextView;
        private TextView priceTextView;
        private EditText amountEditText;
        private Button cancelButton;
        private Button orderButton;


        private Place selectedPlace;
        private PlaceDataService dataService;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            SetContentView(Resource.Layout.PlaceDetailView);
            PlaceDataService dataService = new PlaceDataService();

            var selectedPlaceId = Intent.Extras.GetInt("selectedPlaceId");
            selectedPlace = dataService.GetPlaceById(selectedPlaceId);

            FindViews();
            BindData();
            HandleEvents();
        }

        private void FindViews()
        {
            placeImageView = FindViewById<ImageView>(Resource.Id.placeImageView);
            placeNameTextView = FindViewById<TextView>(Resource.Id.placeNameTextView);
            shortDescriptionTextView                                                    =
FindViewById<TextView>(Resource.Id.shortDescriptionTextView);
            descriptionTextView                                                         =
FindViewById<TextView>(Resource.Id.descriptionTextView);
            priceTextView = FindViewById<TextView>(Resource.Id.priceTextView);
            amountEditText = FindViewById<EditText>(Resource.Id.amountEditText);
            cancelButton = FindViewById<Button>(Resource.Id.cancelButton);
            orderButton = FindViewById<Button>(Resource.Id.orderButton);
        }

        private void BindData()
        {
            placeNameTextView.Text = selectedPlace.Name;
            shortDescriptionTextView.Text = selectedPlace.ShortDescription;
            descriptionTextView.Text = selectedPlace.Description;
            //priceTextView.Text = "Price: " + selectedPlace.Price;
```

```
            //var                                   imageBitmap                                   =
ImageHelper.GetImageBitmapFromUrl("http://gillcleerenpluralsight.blob.core.windows.net/fi
les/" + selectedPlace.ImagePath + ".jpg");

            //placeImageView.SetImageBitmap(imageBitmap);
        }

        private void HandleEvents()
        {
            orderButton.Click += OrderButton_Click;
            cancelButton.Click += CancelButton_Click;
        }

        private void CancelButton_Click(object sender, EventArgs e)
        {

        }

        private void OrderButton_Click(object sender, System.EventArgs e)
        {
            var amount = Int32.Parse(amountEditText.Text);

            var intent = new Intent();
            intent.PutExtra("selectedPlace", selectedPlace.PlaceId);
            intent.PutExtra("amount", amount);

            SetResult(Result.Ok, intent);

            this.Finish();
        }
               }
```

3. PlaceMapActivity.cs

```
[Activity(Label = "Map")]
    public class PlaceMapActivity : Activity
    {
        private Button externalMapButton;
        private FrameLayout mapFrameLayout;
        private MapFragment mapFragment;
        private GoogleMap googleMap;
        private LatLng placeLocation;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            placeLocation = new LatLng(49.231019, 28.471488);

            SetContentView(Resource.Layout.PlaceMapView);

            FindViews();

            HandleEvents();

            CreateMapFragment();

            UpdateMapView();
        }

        private void UpdateMapView()
        {
```

```
            var mapReadyCallback = new LocalMapReady();

            mapReadyCallback.MapReady += (sender, args) =>
            {
                googleMap = (sender as LocalMapReady).Map;

                if (googleMap != null)
                {
                    MarkerOptions markerOptions = new MarkerOptions();
                    markerOptions.SetPosition(placeLocation);
                    markerOptions.SetTitle("MyPlace");
                    googleMap.AddMarker(markerOptions);

                    CameraUpdate cameraUpdate = CameraUpdateFactory.NewLatLngZoom(placeLocation, 15);
                    googleMap.MoveCamera(cameraUpdate);
                }
            };
            mapFragment.GetMapAsync(mapReadyCallback);
        }


        private class LocalMapReady : Java.Lang.Object, IOnMapReadyCallback
        {
            public GoogleMap Map { get; private set; }

            public event EventHandler MapReady;

            public void OnMapReady(GoogleMap googleMap)
            {
                Map = googleMap;
                var handler = MapReady;
                if (handler != null)
                    handler(this, EventArgs.Empty);
            }
        }

        private void CreateMapFragment()
        {
            mapFragment = FragmentManager.FindFragmentByTag("map") as MapFragment;

            if (mapFragment == null)
            {
                var googleMapOptions = new GoogleMapOptions()
                    .InvokeMapType(GoogleMap.MapTypeNormal)
                    .InvokeZoomControlsEnabled(true)
                    .InvokeCompassEnabled(true);

                FragmentTransaction fragmentTransaction = FragmentManager.BeginTransaction();
                mapFragment = MapFragment.NewInstance(googleMapOptions);
                fragmentTransaction.Add(Resource.Id.mapFrameLayout, mapFragment, "map");
                fragmentTransaction.Commit();
            }
        }

        private void FindViews()
        {
            externalMapButton = FindViewById<Button>(Resource.Id.externalMapButton);
        }

        private void HandleEvents()
        {
            externalMapButton.Click += ExternalMapButton_Click;
```

```
        }

        private void ExternalMapButton_Click(object sender, EventArgs e)
        {
            Android.Net.Uri                          placeLocationUri                          =
Android.Net.Uri.Parse("geo:50.846704,4.352446");
            Intent mapIntent = new Intent(Intent.ActionView, placeLocationUri);
            StartActivity(mapIntent);
        }
            }
```

4. PlaceMenuActivity.cs

```
[Activity(Label = "Menu Navigation", MainLauncher = false)]
    public class PlaceMenuActivity : Activity
    {
        private ListView placeListView;
        private List<Place> allPlaces;
        private PlaceDataService placeDataService;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            // Create your application here
            SetContentView(Resource.Layout.PlaceMenuView);

            ActionBar.NavigationMode = ActionBarNavigationMode.Tabs;

            AddTab("Fav", Resource.Drawable.FavoriteIcon, new FavouritePlaceFragment());
            AddTab("Eat", Resource.Drawable.EatLoversIcon, new EatLoversFragment());
            AddTab("Art", Resource.Drawable.ArtLoversIcon, new ArtLoversFragment());
            AddTab("Sport",             Resource.Drawable.SportLoversIcon,             new
SportLoversFragment());
            AddTab("Travel",            Resource.Drawable.TravelLoversIcon,            new
TravelLoversFragment());
        }

        private void AddTab(string tabText, int iconResourceId, Fragment view)
        {
            var tab = this.ActionBar.NewTab();
            tab.SetText(tabText);
            tab.SetIcon(iconResourceId);

            tab.TabReselected += delegate (object sender, ActionBar.TabEventArgs e)
            {
                var                                 fragment                                 =
this.FragmentManager.FindFragmentById(Resource.Id.fragmentContainer);
                if (fragment != null)
                    e.FragmentTransaction.Remove(fragment);
                e.FragmentTransaction.Add(Resource.Id.fragmentContainer, view);
            };

            tab.TabUnselected += delegate (object sender, ActionBar.TabEventArgs e)
            {
                e.FragmentTransaction.Remove(view);
            };

            this.ActionBar.AddTab(tab);
        }

        private           void           PlaceListView_ItemClick(object           sender,
AdapterView.ItemClickEventArgs e)
```

```
        {
            var place = allPlaces[e.Position];

            var intent = new Intent();
            intent.SetClass(this, typeof(PlaceDetailsActivity));
            intent.PutExtra("selectedPlaceId", place.PlaceId);

            StartActivityForResult(intent, 100);
        }

        protected override void OnActivityResult(int requestCode, [GeneratedEnum] Result
resultCode, Intent data)
        {
            base.OnActivityResult(requestCode, resultCode, data);

            if (resultCode == Result.Ok && requestCode == 100)
            {
                var                              selectedPlace                              =
placeDataService.GetPlaceById(data.GetIntExtra("selectedPlaceId", 0));

                var dialog = new AlertDialog.Builder(this);
                dialog.SetTitle("Confirmation");
                dialog.SetMessage(string.Format("You've  added  {0}  time(s)  the  {1}",
data));
                dialog.Show();
            }
        }
     }
```

5. Place.cs

```
public class Place
    {
        public int PlaceId { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
        public string ShortDescription { get; set; }
        public string ImagePath { get; set; }
        public double Lng { get; set; }
        public double Lat { get; set; }
        public bool IsFavorite { get; set; }
        public string GroupName { get; set; }
        public List<string> Comments { get; set; }

    }
```

6. PlaceGroup.cs

```
public class PlaceGroup
    {
        public int PlaceGroupId { get; set; }
        public string Title { get; set; }
        public string ImagePath { get; set; }
        public List<Place> Places { get; set; }
    }
```

7. PlaceDataService.cs

```
public class PlaceDataService
    {
        private static PlaceRepository placeRepository = new PlaceRepository();
```

```
        public List<Place> GetAllPlaces()
        {
            return placeRepository.GetAllPlaces();
        }

        public List<PlaceGroup> GetGroupedPlaces()
        {
            return placeRepository.GetGroupedPlaces();
        }

        public List<Place> GetPlacesForGroup(int placeGroupId)
        {
            return placeRepository.GetPlaceForGroup(placeGroupId);
        }

        public List<Place> GetFavoritePlaces()
        {
            return placeRepository.GetFavoritePlace();
        }

        public Place GetPlaceById(int placeId)
        {
            return placeRepository.GetPlaceById(placeId);
        }
    }
```

8. PlaceRepository.cs

```
public class PlaceRepository
    {
        private static List<PlaceGroup> placeGroups = new List<PlaceGroup>();
        public List<Place> GetAllPlaces()
        {
            IEnumerable<Place> places =
                from placeGroup in placeGroups
                from place in placeGroup.Places

                select place;

            return places.ToList<Place>();
        }

        public Place GetPlaceById(int placeId)
        {
            IEnumerable<Place> places =
                from placeGroup in placeGroups
                from place in placeGroup.Places
                where place.PlaceId == placeId
                select place;

            return places.FirstOrDefault();
        }

        public List<PlaceGroup> GetGroupedPlaces()
        {
            return placeGroups;
        }

        public List<Place> GetPlaceForGroup(int placeGroupId)
        {
            var group = placeGroups.Where(h => h.PlaceGroupId ==
placeGroupId).FirstOrDefault();

            if (group != null)
```

```
            {
                return group.Places;
            }
            return null;
        }

        public List<Place> GetFavoritePlace()
        {
            IEnumerable<Place> places =
                from placeGroup in placeGroups
                from place in placeGroup.Places
                where place.IsFavorite
                select place;

            return places.ToList<Place>();
        }
    }
```

Додаток Г

(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО ВИБОРУ ТУРИСТИЧНИХ ОБ'ЄКТІВ

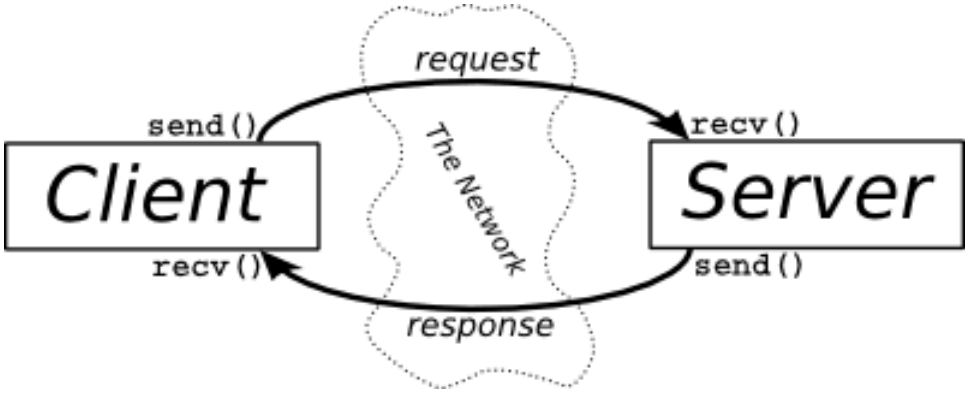
Студентка групи 2АКІТР-23м

Підпис Мирослава ТЕСЛЮК
Ім'я ПРІЗВИЩЕ

Керівник к.т.н., доцент, проф. кафедри КСУ

Підпис Микола БИКОВ
Ім'я ПРІЗВИЩЕ

СХЕМА ВЗАЄМОДІЇ КЛІЄНТСЬКОЇ ТА СЕРВЕРНОЇ ЧАСТИН РОЗРОБЛЮВАНОЇ СИСТЕМИ



UML-ДІАГРАМА ВИКОРИСТАННЯ

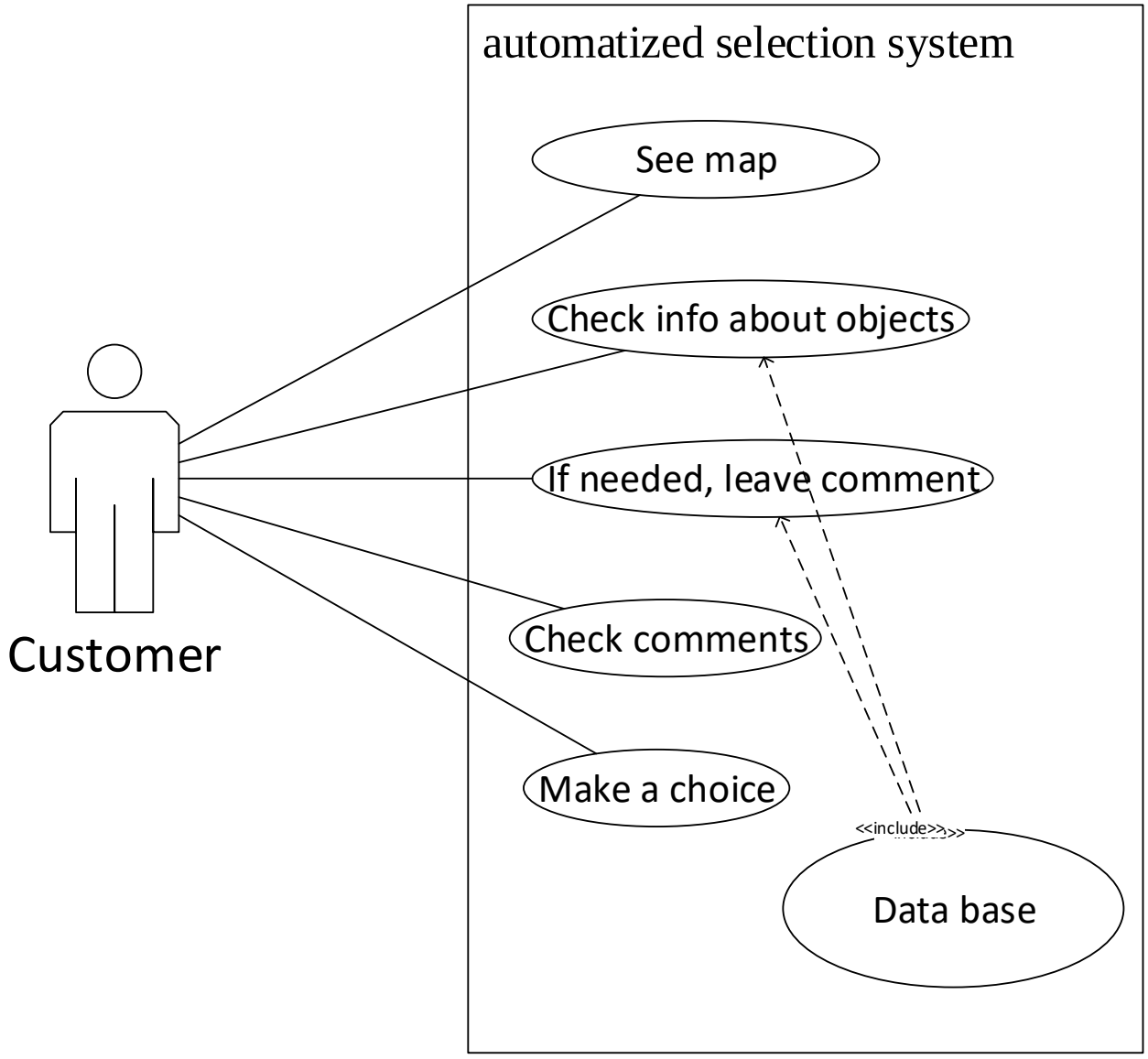
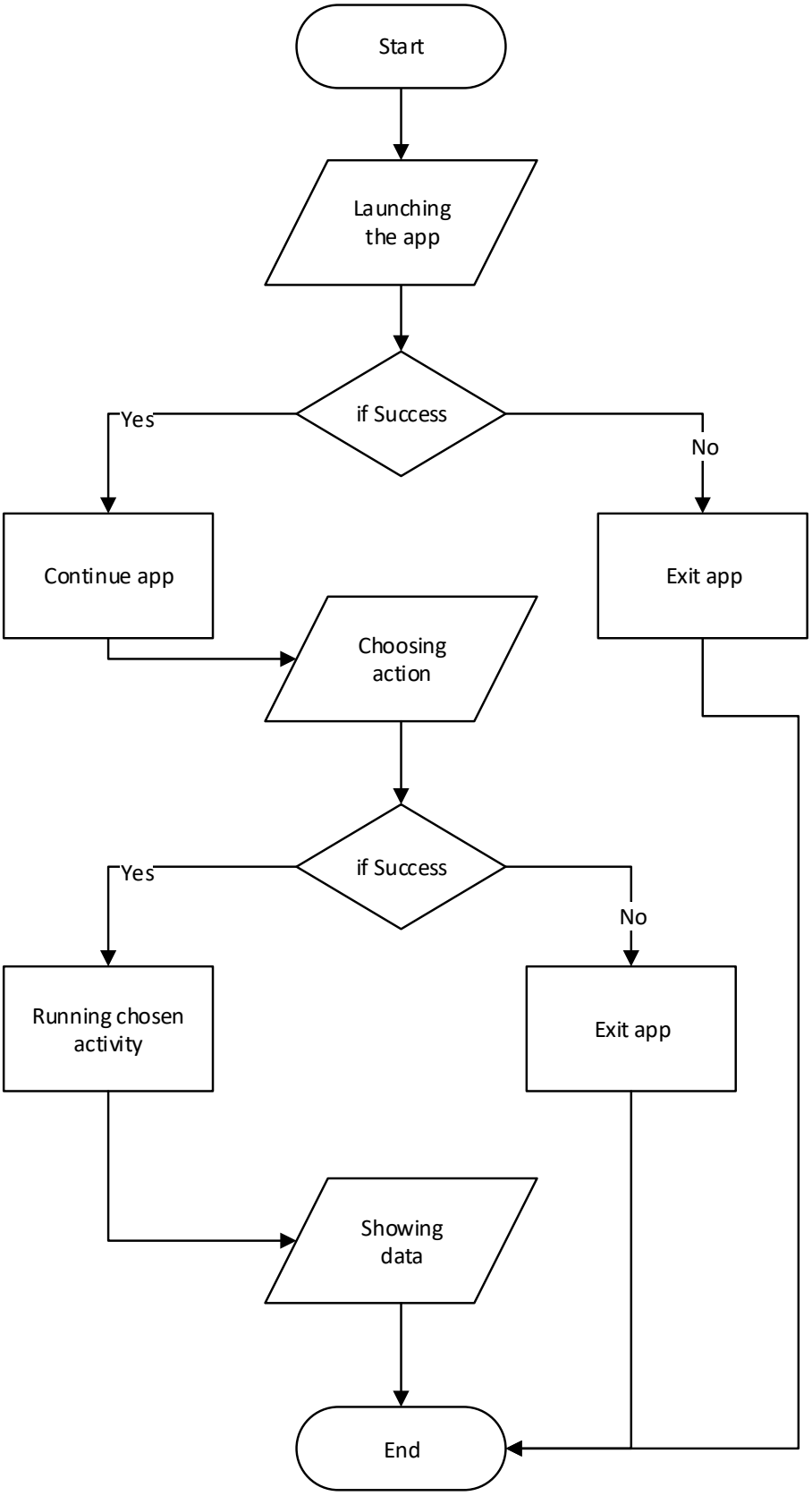
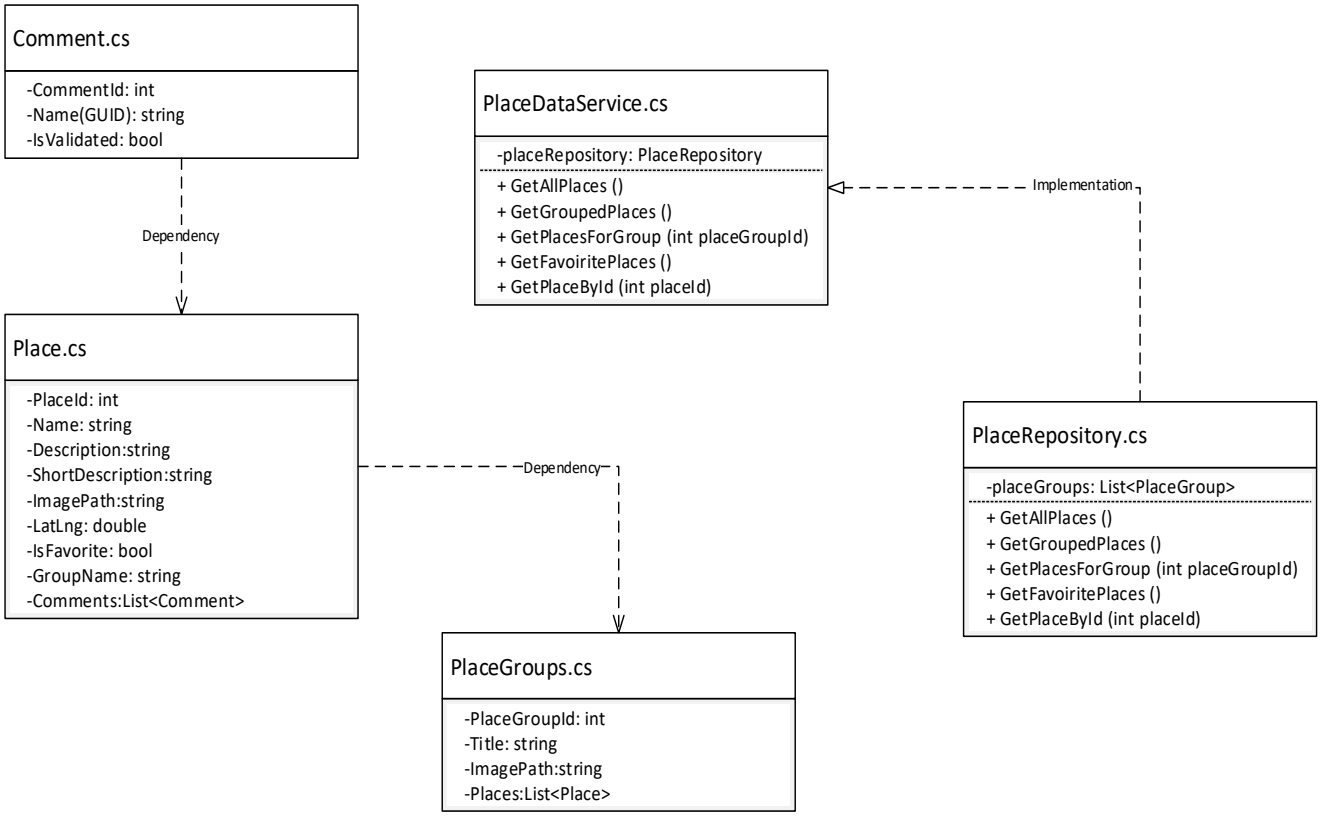


СХЕМА ПРОГРАМИ РОБОТИ СИСТЕМИ



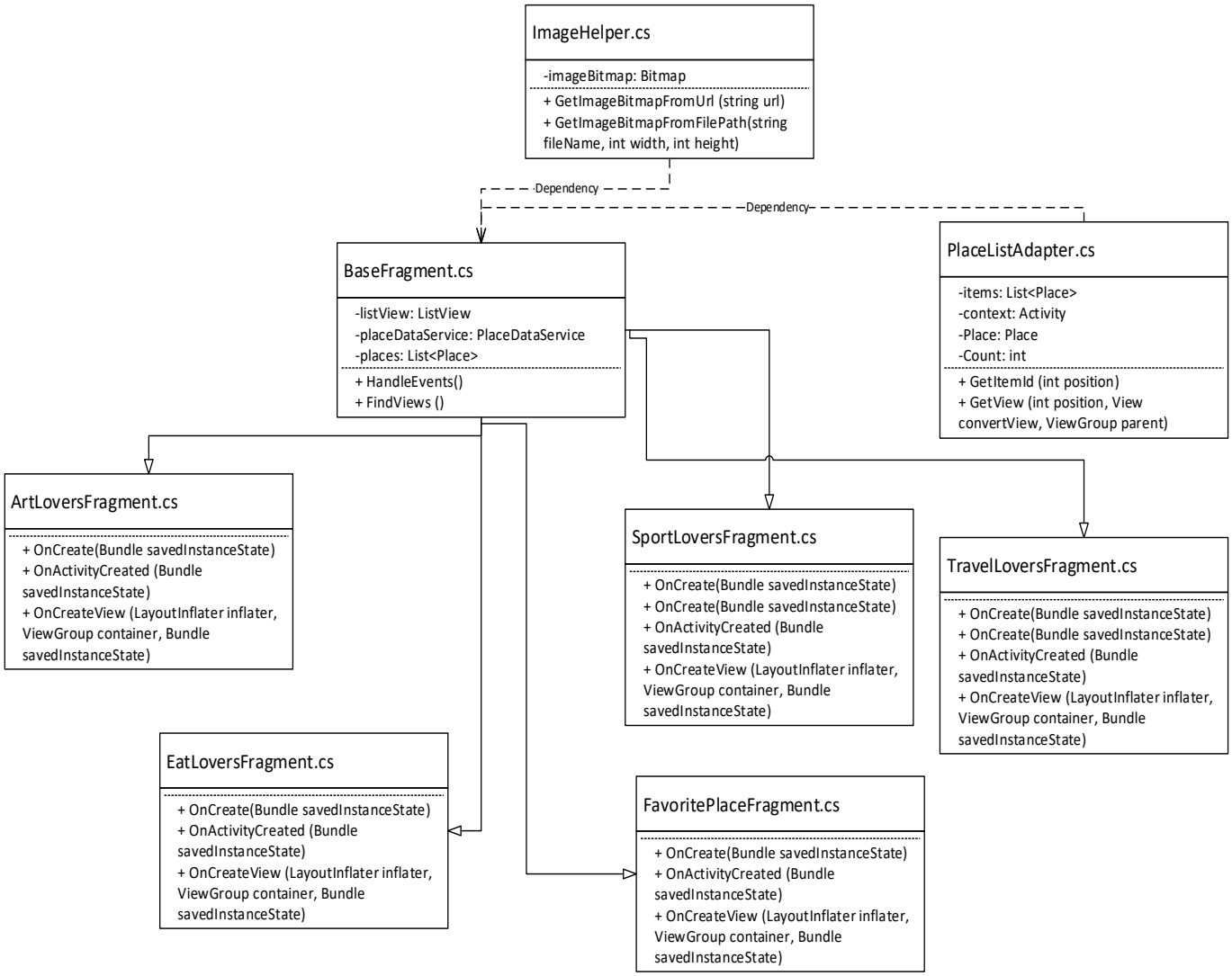
РОЗРОБКА ДІАГРАМИ КЛАСІВ СИСТЕМИ

Діаграма класів для головного проекту



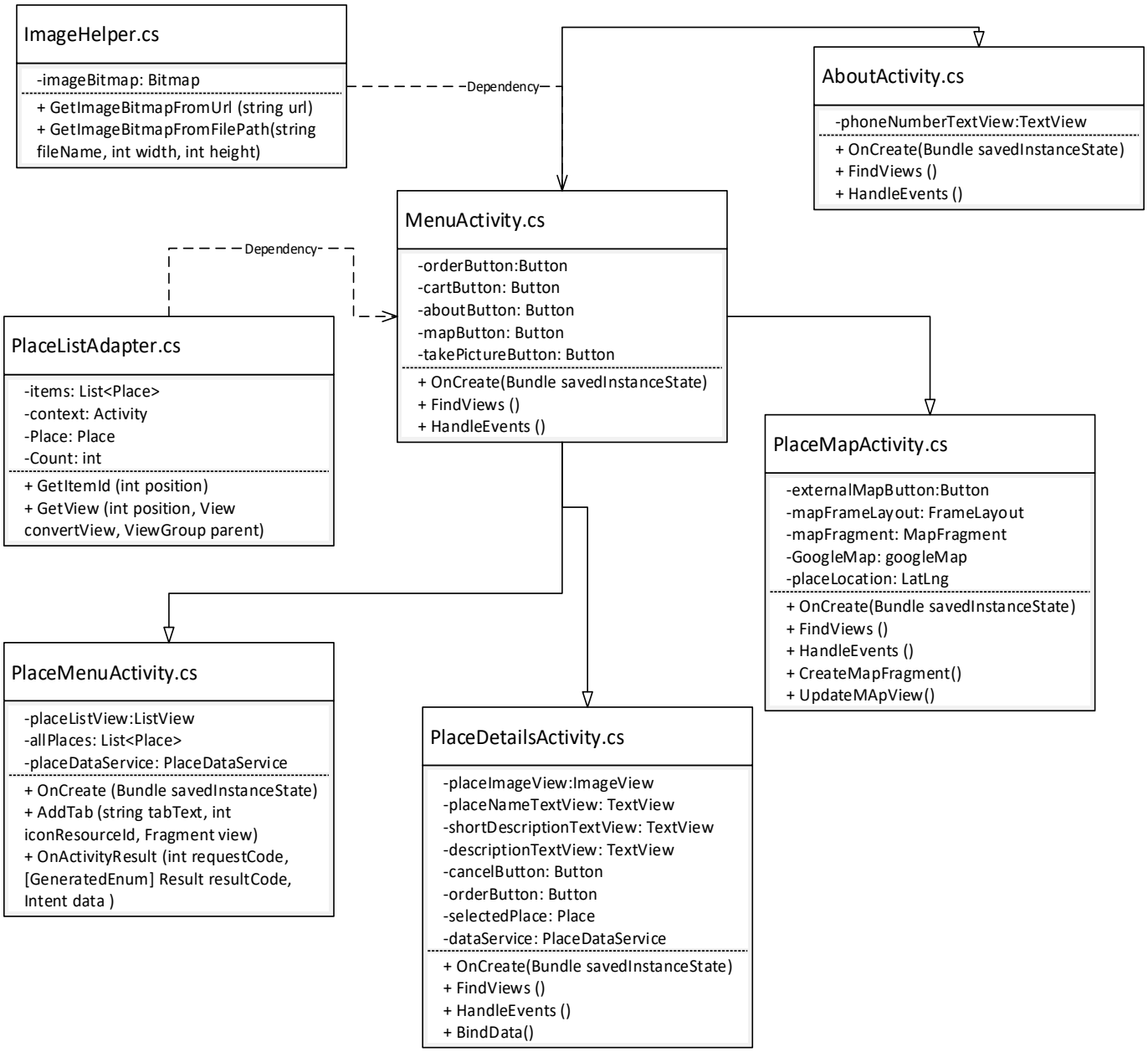
Діаграма класів для проекту Touristsobjects.Android

(частина перша)

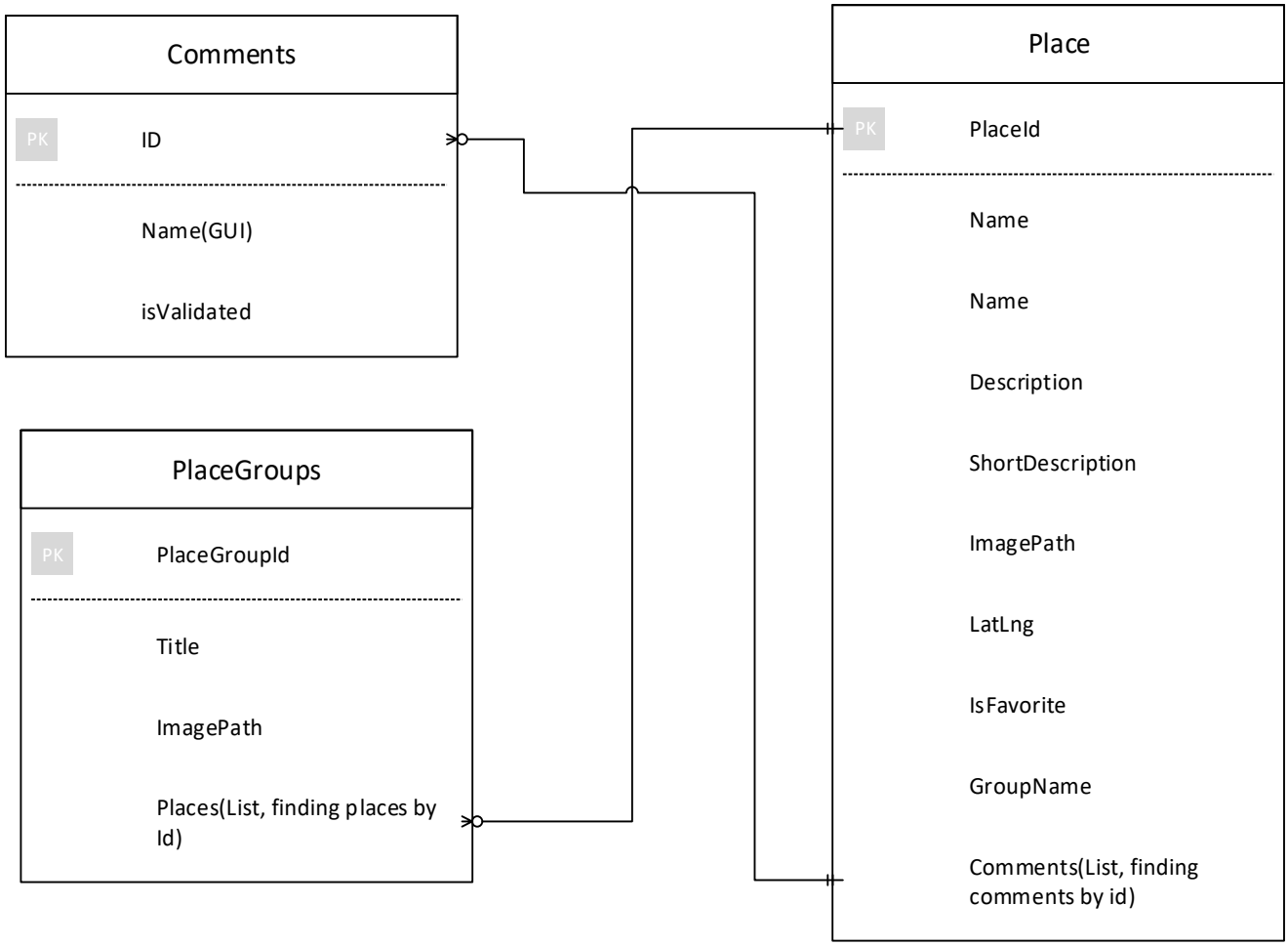


Діаграма класів для проекту TouristsObjects.Android

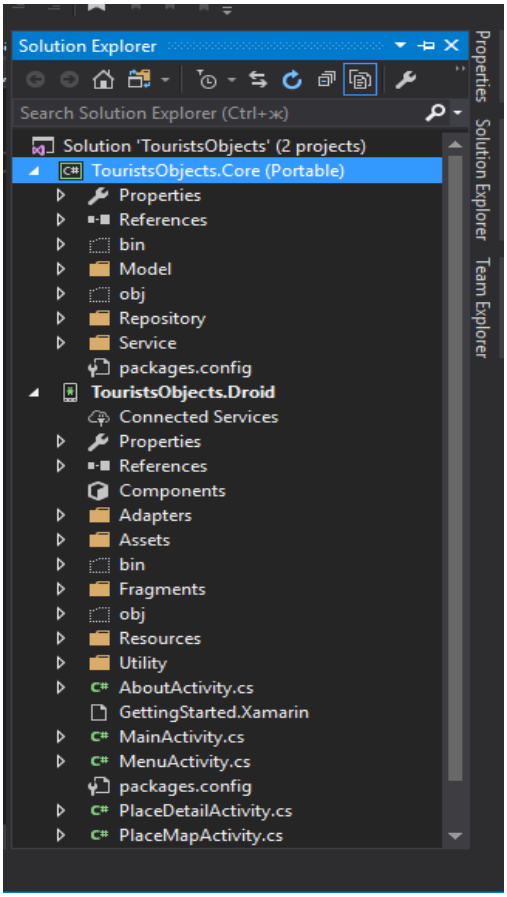
(частина друга)



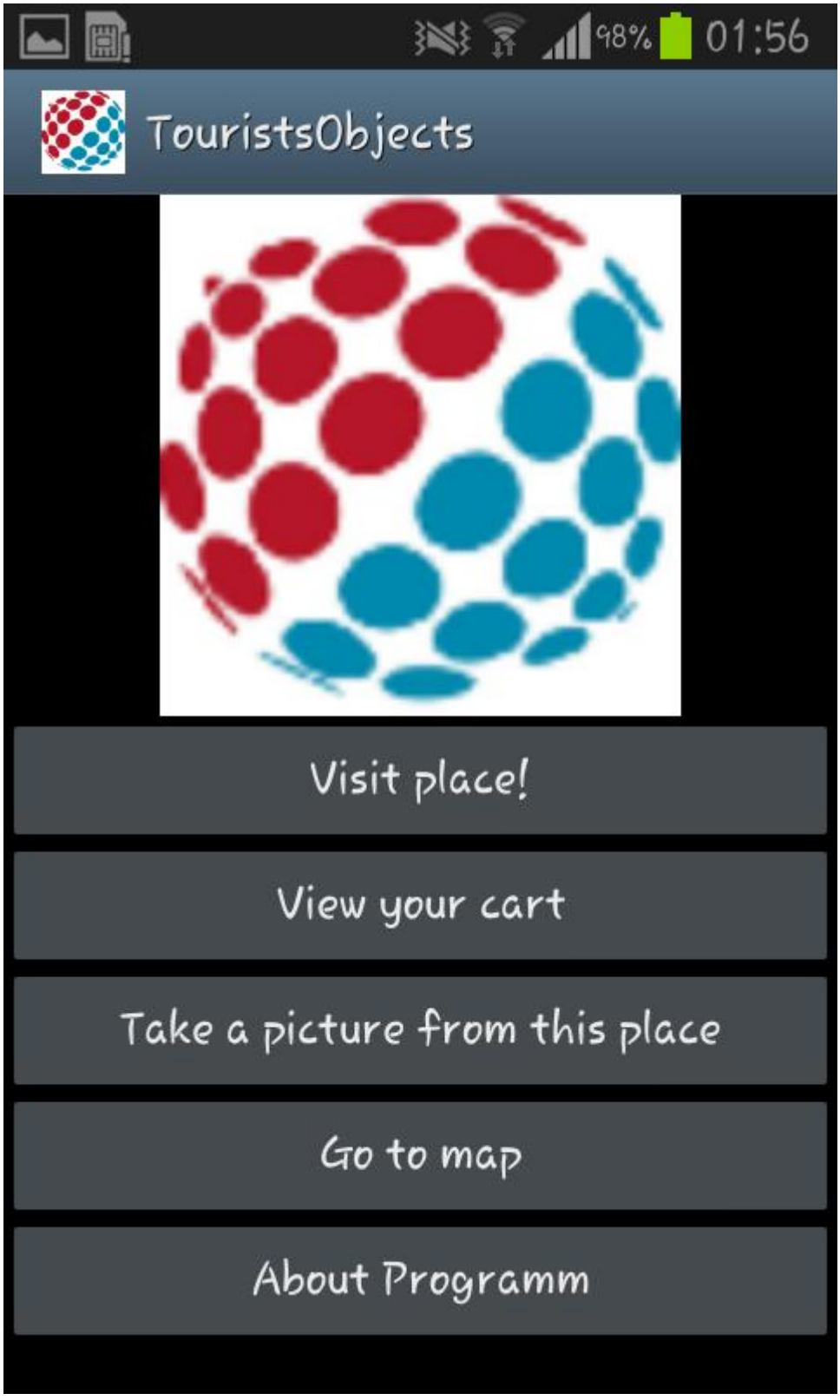
ER – МОДЕЛЬ БАЗИ ДАНИХ СИСТЕМИ АВТОМАТИЗОВАНОГО ПОШУКУ ТУРИСТИЧНИХ МАРШРУТІВ



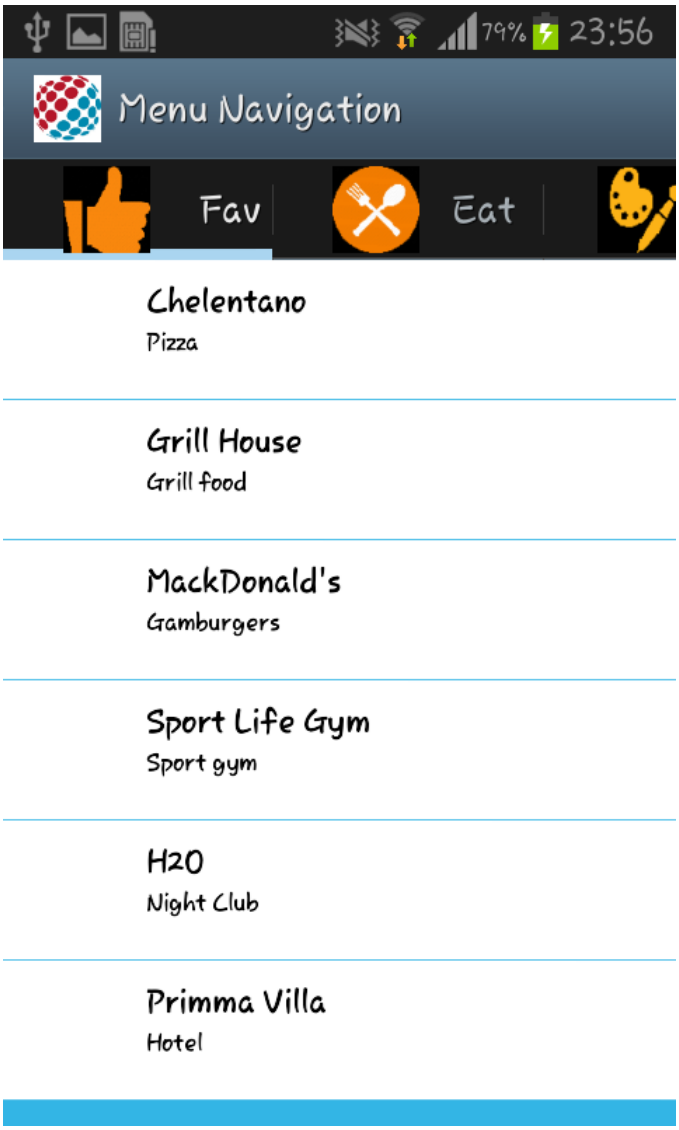
Структура проєкту



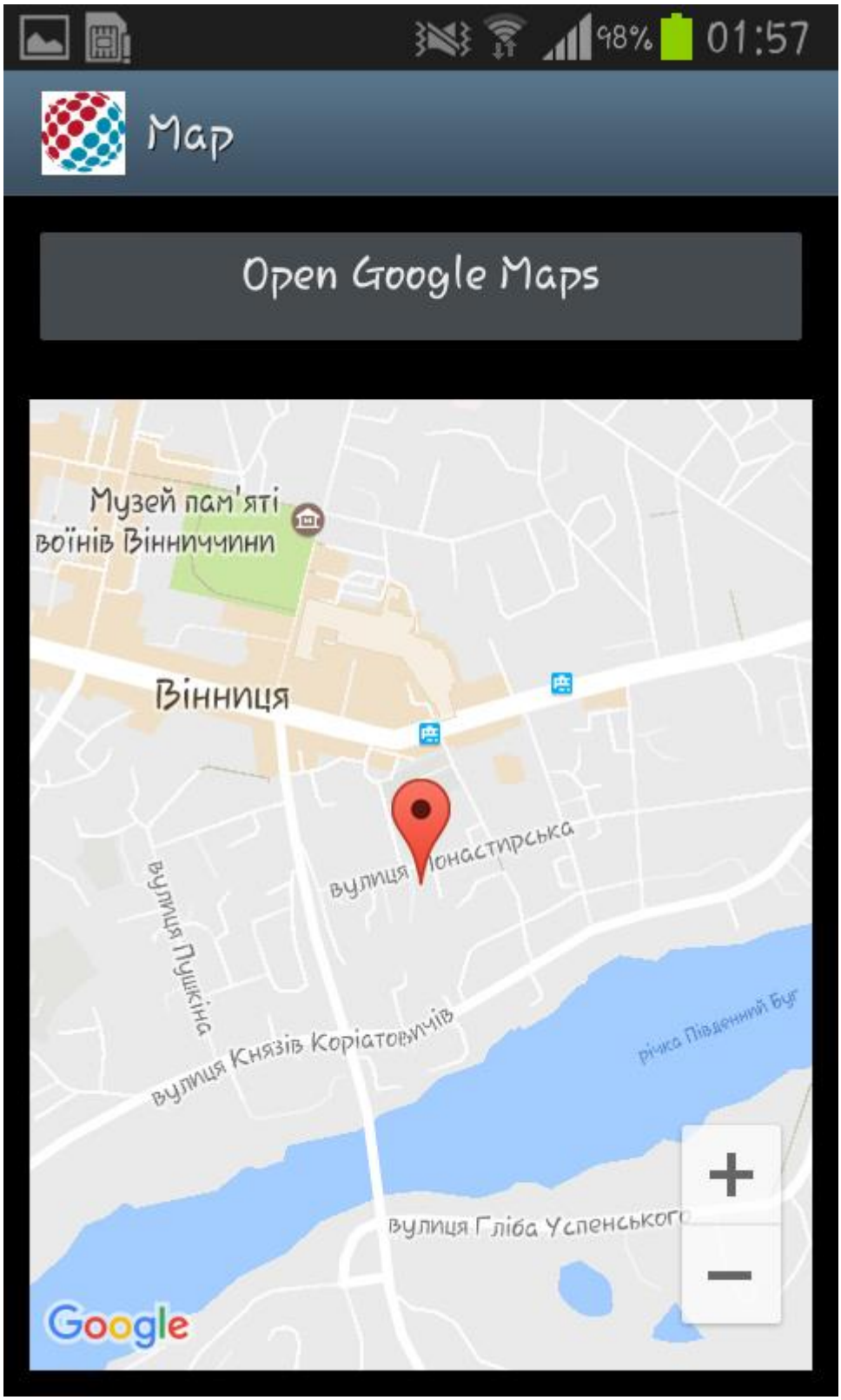
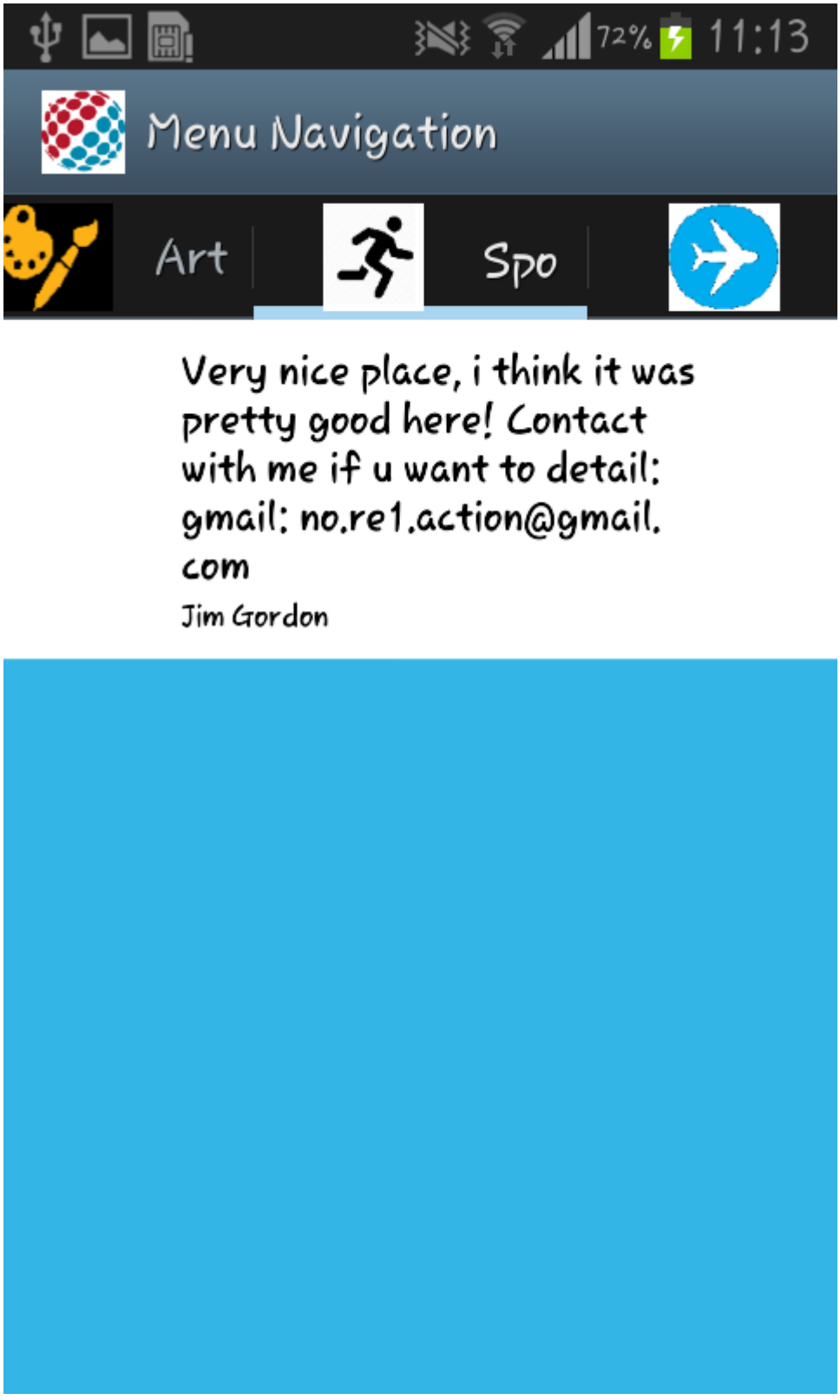
Головне меню програми



Навігація системи з внутрішніми об\єктами



Тестування Google Maps, Google Maps Api



Вікно коментарів