

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ОСТРОЗЬКА АКАДЕМІЯ”
Кафедра інформаційних технологій та аналітики даних**

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістра

на тему: **«Особливості технічного стеку та візуальної стилістики при розробці туристичного мобільного додатку»**

Виконала: студентка 2 курсу, групи МУП-2
другого (магістерського) рівня вищої освіти
спеціальності 122 Комп’ютерні науки

ОПП «Управління проєктами»

Кухарчук Яна Володимирівна

Керівник: *Красюк Б.В., доктор філософії з прикладної математики, викладач, фахівець-практик*

Рецензент: *кандидат технічних наук, доцент, доцент кафедри прикладної математики Донецького національного університету імені Василя Стуса*

Загоруйко Любов Василівна

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри інформаційних технологій та аналітики даних
_____ (проф., д.е.н. Кривицька О.Р.)

Протокол № 5 від «04» грудня 2025 р.

Острог, 2025

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня магістра

Тема: *«Особливості технічного стеку та візуальної стилістики при розробці туристичного мобільного додатку»*

Автор: *Кухарчук Яна Володимирівна*

Науковий керівник: *Красюк Б.В., доктор філософії з прикладної математики, викладач, фахівець-практик кафедри ІТБ*

Захищена «.....»..... 2025 року.

Пояснювальна записка до кваліфікаційної роботи: 68с., 16 рис., 5 табл., 2 додатки, 29 джерела.

Ключові слова: *туристичний мобільний додаток, технічний стек, UI/UX дизайн, брендбук, візуальна айдентика, User Flow, мобільний інтерфейс, цифровий туризм, MyLutsk.*

Короткий зміст праці:

У кваліфікаційній роботі досліджено особливості вибору технічного стеку та розробки візуальної стилістики туристичного мобільного додатку.

У першому розділі здійснено підбір технічного стеку серверної та клієнтської частин мобільного застосунку, а також обґрунтовано вибір бази даних. Проаналізовано доцільність використання сучасних кросплатформних рішень для реалізації туристичного мобільного сервісу. У другому розділі розглянуто методологію розробки програмного продукту, виконано порівняння основних підходів до розробки та визначено особливості впровадження змін на основі зворотного зв'язку від користувачів. У третьому розділі обґрунтовано вибір візуальної стилістики, розроблено брендбук та проведено тестування доступності інтерфейсу відповідно до сучасних стандартів.

Робота завершується висновками, у яких узагальнено основні результати виконаної роботи та визначено напрями практичного застосування запропонованого дизайну.

ANNOTATION
of the qualification work
for obtaining a master's degree

Theme: *«Features of the technical stack and visual style in the development of a tourist mobile application»*

Author: *Kukharchuk Yana*

Scientific supervisor: *Krasyuk B.V., Doctor of Philosophy in Applied Mathematics, teacher, practitioner of the ITB department*

Defended «.....»..... of 2025.

Explanatory note to the qualification work: *68 p., 16 pic., 5 tables., 2 attachments, 29 sources.*

Keywords: *tourist mobile application, technical stack, UI/UX design, brand book, visual identity, User Flow, mobile interface, digital tourism, MyLutsk.*

Summary of the paper:

The qualification work examines the features of choosing a technical stack and developing a visual style for a mobile tourism.

In the first section, the technical stack of the server and client parts of the mobile application is selected, and the choice of the database is justified. The feasibility of using modern cross-platform solutions for implementing a mobile tourism service is analyzed. In the second section, the methodology for developing a software product is considered, the main approaches to development are compared, and the features of implementing changes based on user feedback are determined. In the third section, the choice of visual style is justified, a brand book is developed, and interface accessibility testing is carried out in accordance with modern standards.

The work ends with conclusions, which summarize the main results of the work performed and determine the directions of practical application of the proposed design.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1	
ПІДБІР ТЕХНІЧНОГО СТЕКУ	8
1.1. Технічний стек серверної частини	8
1.2. Технічний стек клієнтської частини.....	12
1.3. Вибір бази даних	16
Висновки до розділу 1.....	21
РОЗДІЛ 2	
МЕТОДОЛОГІЯ РОЗРОБКИ	24
2.1. Порівняння методологій розробки	24
2.2. Особливості впровадження змін на основі відгуку користувача.....	33
Висновки до розділу 2.....	40
РОЗДІЛ 3	
ДИЗАЙН	42
3.1. Вибір візуальної стилістики.....	42
3.2. Accessibility testing	54
Висновки до розділу 3.....	57
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТКИ	64
ДОДАТОК А	64
ДОДАТОК Б	66

ВСТУП

Сучасні тенденції цифровізації та розвитку інформаційних технологій охоплюють усі сфери життя, зокрема й туристичну індустрію. У світі набуває поширення концепція смарт-туризму, яка інтегрується з розвитком «розумних міст» та передбачає використання цифрових сервісів для підвищення зручності, доступності та індивідуалізації подорожей. Мобільні застосунки стали невід'ємним інструментом мандрівників у всьому світі, оскільки дозволяють отримувати актуальну інформацію про маршрути, пам'ятки, події та сервіси, формувати індивідуальні туристичні маршрути та ділитися враженнями в інтерактивному форматі.

Особливої актуальності ці процеси набувають в Україні, де останні роки спостерігається стійке зростання внутрішнього туризму. Луцьк - одне з найдавніших міст країни, багате на історичні та культурні пам'ятки, проте досі не має єдиного комплексного мобільного сервісу, який би надавав туристам повну інформацію про можливості міста, забезпечував зручну навігацію та пропонував інтерактивний функціонал.

За статистикою, понад дві третини мандрівників у світі користуються мобільними сервісами для планування поїздок і отримання актуальної інформації безпосередньо під час подорожі. У багатьох європейських містах функціонують спеціалізовані мобільні сервіси для туристів, такі як *VisitBerlin*, *Paris City Guide*, *Krakow Guide*, які довели свою ефективність у популяризації туристичного потенціалу. Враховуючи ці тенденції, створення сучасного мобільного додатку для міста Луцьк є не лише доцільним, а й необхідним кроком у напрямі розвитку туристичної інфраструктури та формування позитивного іміджу міста.

Наукова новизна представленої роботи полягає у створенні комплексного рішення для Луцька, яке об'єднує кілька ключових сервісів - довідкові телефони, маршрути, фото-зони, транспортну інформацію та можливості для покупок - в єдиному мобільному застосунку. Це дозволяє поєднати теоретичні засади

проектування інформаційних систем із практичними аспектами створення сучасного цифрового продукту для туристів.

Для досягнення мети переддипломної практики були поставлені такі основні завдання:

- дослідити ринок туристичних мобільних застосунків і виявити потреби цільової аудиторії;
- проаналізувати існуючі аналоги та визначити їхні переваги й недоліки;
- сформулювати технічні та функціональні вимоги до майбутнього додатку;
- обґрунтувати вибір архітектури та технологій (серверна й клієнтська частини, база даних);
- розробити прототип інтерфейсу у середовищі Figma, враховуючи принципи UX/UI-дизайну та доступності.

Методологічною основою дослідження стали методи аналізу наукових джерел та світового досвіду, порівняння існуючих рішень, а також принципи UX/UI-дизайну, прототипування та тестування доступності.

Практична значущість роботи полягає у можливості впровадження розробленої концепції додатку «MyLutsk» як реального програмного продукту. Він здатен стати інноваційним інструментом для популяризації культурної спадщини міста, підвищити рівень туристичного сервісу, полегшити навігацію для відвідувачів, а також сприяти формуванню позитивного іміджу Луцька як сучасного туристичного центру. Крім того, результати дослідження можуть бути використані в контексті програм розвитку «розумного міста» та цифрової трансформації регіону.

Таким чином, робота має як теоретичне, так і практичне значення: теоретично вона дозволяє дослідити особливості розробки мобільних застосунків у сфері туризму, а практично – дає змогу створити основу для

впровадження реального додатку, який підвищить зручність та інформативність туристичних подорожей у Луцьку.

Об'єктом дослідження є ринок туристичних мобільних додатків, орієнтованих на забезпечення доступності для користувачів з особливими потребами.

Предметом дослідження є процес розробки туристичного мобільного додатку, що включає вибір технічного стеку, візуальної стилістики, методології розробки та забезпечення доступності.

Результати проведеного дослідження та розробки мають широке практичне застосування. У туристичній сфері Луцька мобільний додаток може стати ефективним інструментом для популяризації міста, забезпечуючи відвідувачів повною та зручно структурованою інформацією про історико-культурні пам'ятки, маршрути, події й сервіси. У закладах культури та музейних установах він відкриває можливість інтеграції віртуальних гідів і аудіоекскурсій, що дозволяє відвідувачам глибше зануритися в історичну спадщину та створює нові формати взаємодії з аудиторією. Крім того, результати роботи можуть бути інтегровані в програми розвитку міської інфраструктури, зокрема в концепцію «розумного міста», сприяючи цифровій трансформації регіону, підвищенню доступності інформації та покращенню туристичного досвіду як для гостей, так і для мешканців Луцька.

РОЗДІЛ 1

ПІДБІР ТЕХНІЧНОГО СТЕКУ

1.1. Технічний стек серверної частини

Серверна частина є фундаментом будь-якого сучасного мобільного застосунку, оскільки саме вона забезпечує коректну роботу функціоналу, збереження та обробку даних, а також інтеграцію з зовнішніми сервісами. Якщо клієнтська частина відповідає за візуальний інтерфейс і безпосередню взаємодію з користувачем, то серверна частина, яку також називають backend, виконує роль «невидимого двигуна» системи. У випадку з туристичним мобільним додатком міста Луцьк серверна частина повинна відповідати кільком критично важливим вимогам:

- надійність та продуктивність. Туристичний додаток має обслуговувати одночасно велику кількість користувачів, особливо в пікові туристичні сезони. Це означає, що сервер повинен бути здатним ефективно обробляти сотні й навіть тисячі запитів у реальному часі.
- масштабованість. У перспективі додаток може розширюватися новими функціями - від додавання віртуальних екскурсій та інтеграції з аудіогідами до системи бронювання квитків у музеї. Серверна частина має дозволяти таке розширення без значних перебудов архітектури.
- Інтеграція з картографічними сервісами. Оскільки одним із ключових завдань є побудова маршрутів та відображення пам'яток на карті, сервер має підтримувати роботу з API Google Maps, Mapbox чи іншими сервісами геолокації.
- безпека. Туристичний додаток може зберігати персональні дані користувачів (акаунти, історію відвіданих місць, маршрути), тому

захист від несанкціонованого доступу та кібератак є обов'язковим аспектом.

На сучасному ринку існує кілька найбільш поширених рішень для створення серверної частини. Кожне з них має свої особливості, переваги й недоліки, тому вибір залежить від специфіки проєкту[1].

Node.js (з фреймворками Express чи NestJS): Node.js - це серверне середовище, побудоване на основі JavaScript. Його головною перевагою є асинхронна архітектура, яка дозволяє ефективно обробляти велику кількість одночасних запитів. Це особливо важливо для туристичного застосунку, де користувачі можуть масово шукати маршрути чи переглядати інформацію про пам'ятки.

Django (Python): Django - це високорівневий фреймворк, який дозволяє швидко створювати серверні застосунки завдяки вбудованим інструментам. Він має готову адмін-панель, що спрощує керування даними. Django добре підходить для додатків, де ключовим є контент, однак він менш оптимальний для роботи з великою кількістю одночасних запитів у реальному часі.

Spring Boot (Java): Spring Boot - це рішення для створення масштабованих корпоративних систем. Воно забезпечує високу надійність, модульність і добре інтегрується з мікросервісною архітектурою. Водночас воно є більш складним у налаштуванні та вимагає більше ресурсів на розробку.

ASP.NET Core (C#): ASP.NET Core - сучасний кросплатформенний фреймворк від Microsoft, що відзначається високою продуктивністю, широкими можливостями безпеки та інтеграцією з корпоративними рішеннями. Його доцільно використовувати для великих enterprise-проєктів, проте він менш зручний для невеликих команд і потребує значних ресурсів. Для наочності наведемо порівняльну таблицю:

Таблиця 1.1

Порівняльна таблиця серверних технологій

Технологія	Переваги	Недоліки	Приклади застосування
Node.js (Express/NestJS)	Висока продуктивність, асинхронність, велика екосистема	Складність у підтримці великих проєктів без фреймворків	Чат-застосунки, REST API, туристичні додатки
Django (Python)	Швидка розробка, зручна адмін-панель	Нижча продуктивність	Сайти з контентом, CRM-системи
Spring Boot (Java)	Стабільність, масштабованість	Високий поріг входження, складніший процес розробки	Банківські системи, корпоративні портали
ASP.NET Core (C#)	Висока швидкодія, інтеграція з enterprise-рішеннями	Складність для невеликих команд	Корпоративні веб-додатки, ERP-системи

Для реалізації туристичного мобільного додатку міста Луцьк оптимальним є використання **Node.js з Express**, оскільки ця технологія поєднує високу продуктивність, простоту створення REST API та можливість гнучкої інтеграції з іншими сервісами. Node.js добре підходить для систем, які потребують роботи в реальному часі, а також дозволяє реалізувати push-сповіщення та швидко обробляти геолокаційні дані[2].

Архітектуру серверної частини проєкту можна подати у вигляді діаграми (рис. 1.1).

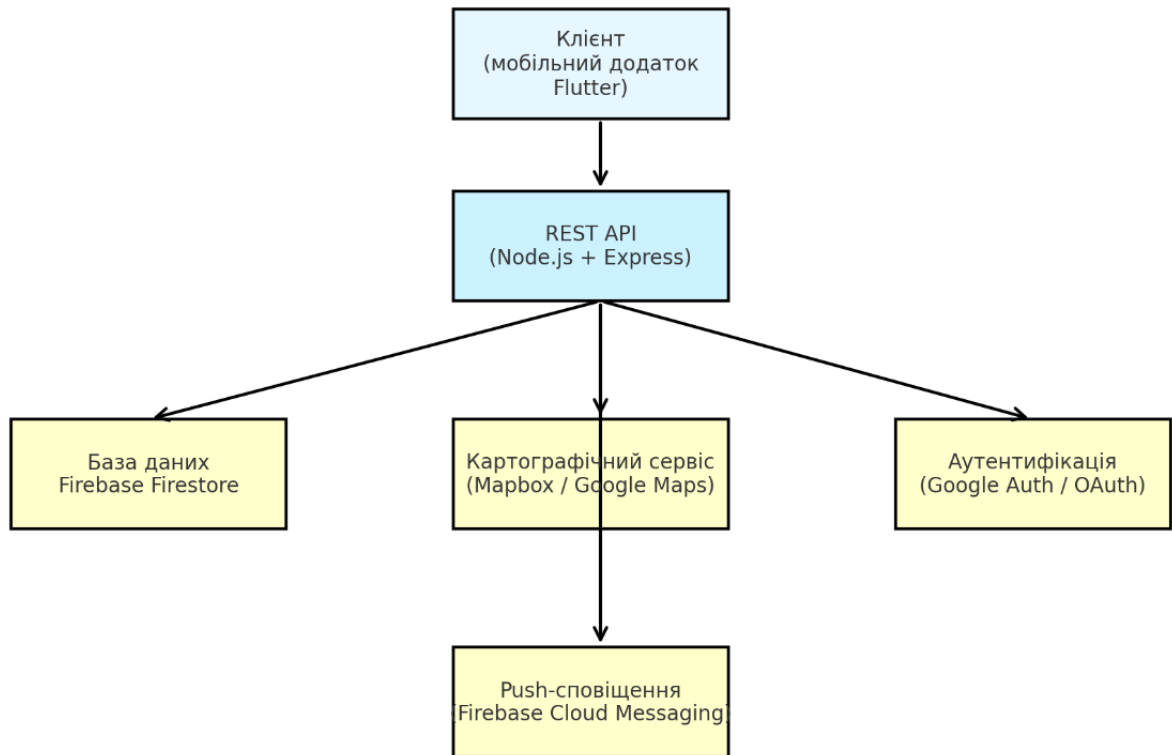


Рис. 1.1 Архітектура серверної частини

Джерело: розроблено автором

На рисунку 1.1 показано, що клієнтська частина (мобільний додаток, розроблений на Flutter) взаємодіє із сервером через REST API, побудований на Node.js з Express. Далі серверна частина розподіляє запити між ключовими модулями:

- базою даних Firebase Firestore, де зберігаються туристичні об'єкти, маршрути, користувачі, відгуки та мультимедійний контент;
- картографічними сервісами (**Google Maps, Mapbox**), що відповідають за побудову маршрутів і роботу з геолокацією;
- системою аутентифікації (**Google Auth, OAuth**), яка забезпечує реєстрацію та захист персональних даних;
- модулем push-сповіщень (**Firebase Cloud Messaging**), що дозволяє надсилати актуальні повідомлення туристам у реальному часі.

Завдяки такій архітектурі система є гнучкою, модульною та масштабованою, що дозволяє розширювати функціонал у майбутньому, наприклад додавати нові сервіси чи інтегрувати додаток з іншими платформами.

Отже, серверна частина є критично важливим компонентом мобільного туристичного додатку міста Луцьк. Вона визначає швидкість і стабільність роботи системи, безпеку персональних даних користувачів, а також можливості подальшого розширення функціоналу. Серед аналізованих технологій найбільш оптимальним вибором для даного проєкту є Node.js з Express, що дозволяє швидко створити ефективну архітектуру, забезпечити інтеграцію з картами, підтримати роботу з базою даних і залишає простір для подальшого розвитку проєкту[3].

1.2. Технічний стек клієнтської частини

Клієнтська частина мобільного застосунку - це саме те, що бачить і з чим взаємодіє користувач. Вона формує перше враження від роботи з програмою. Якщо інтерфейс зручний і зрозумілий, користувач швидко знаходить потрібну інформацію та із задоволенням повертається до додатку. Якщо ж інтерфейс перевантажений чи незручний, навіть найкращий функціонал може залишитися непоміченим.

Для туристичного мобільного застосунку міста Луцька важливо, щоб програма була легкою у використанні, мала привабливий дизайн і водночас підтримувала необхідні функції. Адже туристи зазвичай не мають часу довго розбиратися з інтерфейсом - вони хочуть швидко знайти пам'ятки, прокласти маршрут, подивитися фото чи отримати підказку, де поїсти[4].

Сучасні мобільні застосунки будуються двома основними шляхами:

Нативна розробка - коли для Android і iOS створюються окремі програми (мовами Kotlin і Swift). Це забезпечує найвищу продуктивність та повний доступ

до функцій пристрою. Але мінус у тому, що потрібно писати й підтримувати два різні додатки, що вдвічі збільшує витрати часу й ресурсів.

Кросплатформна розробка - коли один код працює і на Android, і на iOS. Це дозволяє значно пришвидшити розробку та полегшити підтримку продукту.

Найбільш поширеними кросплатформними технологіями є **React Native** і **Flutter**. React Native, хоча й перевірений часом, іноді поступається продуктивністю та узгодженістю UI. Flutter - сучасний фреймворк від Google, що використовує мову Dart і власний графічний рушій **Skia**. Завдяки цьому він забезпечує стабільний рендеринг інтерфейсу зі швидкістю 60/120 fps та однаковий вигляд додатку на всіх пристроях.

Для кращого розуміння наведемо порівняння підходів у таблиці:

Таблиця 1.2

Підхід / Фреймворк	Нативно (Swift/Kotlin)	React Native	Flutter
Продуктивність UI	Максимальна	Висока	Дуже висока
Доступ до API	Повна	Через модулі	Плагіни
Вартість підтримки	Висока	Середня	Низька
Швидкість розробки	Нижча	Висока	Висока
Узгодженість дизайну	Середня	Середня	Висока

У нашому проєкті було прийнято рішення використати **Flutter** як основний інструмент для розробки клієнтської частини. Flutter - це кросплатформний фреймворк, розроблений компанією Google, що дозволяє створювати мобільні застосунки для Android та iOS з єдиної кодової бази, використовуючи мову програмування **Dart**.

Flutter дозволяє швидко створити якісний додаток, який однаково добре працює і на Android, і на iOS. Крім того, Flutter підтримує технологію **Hot Reload**

- це означає, що розробник одразу бачить зміни в інтерфейсі без перезапуску додатку. Така можливість значно пришвидшує розробку й тестування.

Щоб підкреслити практичність вибору, наведемо кілька прикладів компаній, що застосували Flutter у своїх продуктах:

- **Google Ads** - офіційний додаток для керування рекламними кампаніями, де важливі продуктивність та якість UI.
- **Alibaba** - використання Flutter дозволило масштабувати мобільні сервіси для мільйонів користувачів.
- **BMW** - застосовує Flutter для створення інтерфейсів мультимедійних систем у автомобілях.

Ці приклади підтверджують, що Flutter може витримувати серйозні навантаження і застосовуватися навіть у глобальних продуктах.

Архітектура клієнтської частини в нашому проєкті побудована за принципом поділу на кілька шарів. Такий підхід дозволяє чітко відокремити відповідальність кожного компонента, що спрощує розробку та тестування.

- **презентаційний шар** - відповідає за інтерфейс: екрани, кнопки, стилі;
- **шар управління станами** (наприклад, Riverpod) - стежить, щоб інтерфейс завжди відповідав поточним даним;
- **доменний шар** - зберігає бізнес-логіку: пошук пам'яток, додавання у «Вибране» тощо;
- **шар даних** - працює з API, Firebase та локальним кешем.

Це дозволяє окремо перевіряти кожен компонент і спрощує підтримку в майбутньому.

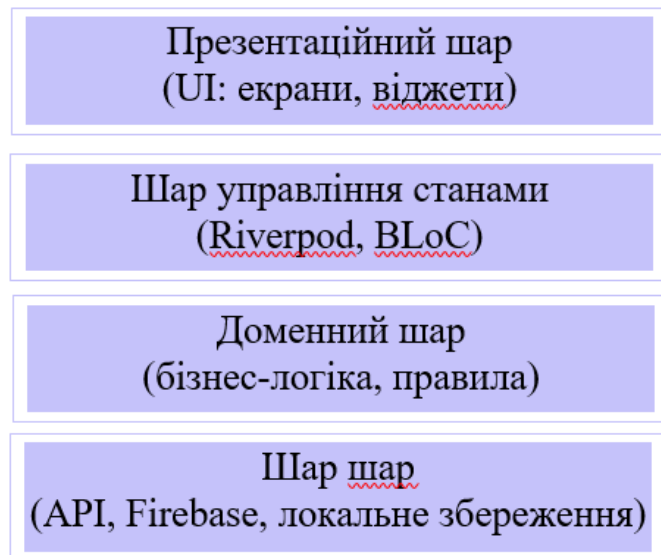


Рис. 1.2 Архітектура клієнтської частини

Джерело: розроблено автором

Для туристичного мобільного застосунку карта є центральним елементом, адже саме через неї користувач взаємодіє з містом, шукає цікаві локації та будує маршрути. У процесі розробки було розглянуто два основні рішення для реалізації картографічної складової - **Google Maps SDK** та **Mapbox**. Перший варіант має очевидну перевагу завдяки своїй широкій популярності: більшість користувачів добре знайомі з інтерфейсом Google Maps, що робить роботу з додатком інтуїтивно зрозумілою. Водночас цей сервіс пропонує розвинену інфраструктуру для побудови маршрутів, пошуку закладів і відображення об'єктів на карті. Альтернативою є Mapbox, який хоч і менш відомий пересічним користувачам, проте має низку вагомих переваг. Він відзначається високим рівнем кастомізації дизайну карт, що дозволяє адаптувати їх під стиль додатку, а також підтримкою офлайн-режиму - функціоналу, вкрай важливого для туристів, які можуть залишитися без доступу до інтернету під час подорожі. У нашому проєкті застосовано комбінований підхід: **онлайн-навігація реалізована через Google Maps, а офлайн-карти - на базі Mapbox**, що забезпечує баланс між зручністю і функціональністю.

Не менш важливою складовою клієнтської частини є **забезпечення якості та стабільності роботи застосунку**. У Flutter передбачено кілька рівнів

тестування, що дозволяють перевіряти як окремі елементи, так і систему в цілому. **Unit-тести** орієнтовані на перевірку бізнес-логіки та коректність роботи алгоритмів. **Widget-тести** дозволяють протестувати інтерфейсні компоненти та перевірити їхню реакцію на різні вхідні дані. Завершальним етапом є **інтеграційні тести**, які охоплюють увесь функціонал додатку та дають змогу оцінити його роботу в умовах, максимально наближених до реальних. Такий підхід мінімізує ризик виникнення критичних помилок та гарантує стабільність системи у продуктивному середовищі.

Отже, клієнтська частина туристичного застосунку для міста Луцька поєднує сучасний технологічний стек із зручними архітектурними рішеннями. Використання Flutter у поєднанні з Riverpod, картографічними SDK та офлайн-кешуванням забезпечує надійність, швидкодію і комфорт для користувачів. Така система не лише задовольняє потреби туристів сьогодні, але й дозволяє легко розширювати функціонал у майбутньому, наприклад додавати аудіогіди, маршрути з доповненою реальністю чи рекомендації за інтересами.

1.3. Вибір бази даних

У сучасних мобільних додатках база даних є «серцем», яке зберігає і постачає всю інформацію. Саме від правильно обраної бази даних залежить, наскільки швидко працюватиме програма, чи зможе вона обробляти великі обсяги даних і чи буде зручною для користувачів. Для туристичного додатку міста Луцька база даних має особливе значення, адже додаток покликаний стати цифровим помічником для гостей міста.

Уявімо ситуацію: турист приїжджає до Луцька і хоче знайти найближчі пам'ятки, прокласти маршрут до замку Любарта, дізнатися про фестивалі чи культурні заходи. Усі ці запити повинні виконуватися швидко й без затримок. Якщо база даних працюватиме повільно або некоректно, користувач може втратити інтерес до застосунку. Тому вибір правильної технології для зберігання інформації є ключовим завданням на етапі проектування.

Додатковою вимогою є **масштабованість**. Туристичний додаток може починатися як локальний проєкт для Луцька, але згодом розширитися на інші міста чи навіть регіони. Це означає, що база даних має бути готовою обробляти як сотні, так і тисячі одночасних запитів. Ще один важливий аспект - **підтримка мультимедіа**. Сучасний туристичний сервіс неможливо уявити без фотографій пам'яток, карт, афіш подій. Відтак обрана база даних має бути зручною для зберігання та швидкої обробки не лише текстової, а й графічної інформації. Важливим фактором є **зручність інтеграції з мобільними технологіями**. Оскільки клієнтська частина нашого додатку реалізована у Flutter, база даних повинна мати прості інструменти для підключення та роботи саме з цим фреймворком. Це зменшить час розробки і зробить застосунок більш стабільним[5].

Завдання, які повинна вирішувати база даних у нашому проєкті:

- зберігання інформації про туристичні об'єкти (назва, опис, фото, координати);
- підтримка роботи з картами та маршрутами (координати, точки інтересу);
- збереження налаштувань користувачів (вибрана мова, «улюблені» місця);
- можливість роботи в офлайн-режимі;
- швидка синхронізація із сервером, коли є доступ до Інтернету.

У світі ІТ існує два основних підходи до організації даних.

Перший - реляційні бази (SQL). Це традиційні системи, де вся інформація зберігається в таблицях із чіткою структурою. Наприклад, у таблиці «Пам'ятки» можуть бути колонки «Назва», «Адреса», «Опис», «Координати». Такі бази дуже надійні та зручні для складних запитів, коли потрібно швидко знайти дані за кількома критеріями. Приклади: MySQL, PostgreSQL, SQLite.

Другий - нереляційні (NoSQL). Тут усе працює більш гнучко. Дані можуть зберігатися у вигляді документів, схожих на JSON. Це дуже зручно для

мобільних застосунків, де часто потрібно швидко віддавати дані у зручному форматі. Приклади: Firebase Firestore, MongoDB, CouchDB.

У сучасній практиці вибір залежить від того, які саме дані потрібно обробляти. Якщо їх легко структурувати в таблиці - доречно використовувати SQL. Якщо дані різноманітні та часто змінюються - зручніше працювати з NoSQL.

Для нашого проєкту важливо було знайти баланс між **простотою використання, швидкістю обробки даних та зручністю інтеграції з мобільним додатком.**

На сучасному ринку існує велика кількість баз даних, кожна з яких має свої особливості, переваги та недоліки. Для туристичного мобільного додатку ми розглядали кілька найпопулярніших варіантів, які найчастіше використовуються у веб та мобільній розробці.

MySQL є однією з найвідоміших реляційних баз даних. Вона добре підходить для роботи з табличними даними, забезпечує високу стабільність та має величезну спільноту розробників. Завдяки цьому проблеми чи питання, які виникають під час розробки, можна швидко вирішити. Проте MySQL менш гнучка у випадках, коли потрібно зберігати мультимедіа чи часто змінювану інформацію, що може створювати певні обмеження.

PostgreSQL - це ще одна потужна реляційна база даних, яка вважається більш сучасною та функціональною у порівнянні з MySQL. Вона підтримує складні запити та великі обсяги даних, що робить її популярною у великих корпоративних проєктах.

Однак інтеграція з мобільними застосунками може виявитися складнішою, і це збільшує час та зусилля для налаштування.

MongoDB належить до нереляційних баз даних (NoSQL) і зберігає дані у форматі документів. Це дозволяє працювати з більш гнучкими структурами та легше зберігати дані, які часто змінюються, наприклад маршрути чи мультимедійний контент. MongoDB також добре підходить для роботи з

геолокаційними даними, що може бути актуально для нашого додатку. Проте для мобільних проєктів інтеграція потребує додаткових налаштувань.

Firebase Realtime Database / Firestore - це хмарна база даних від Google, яка створена спеціально для швидкої інтеграції з мобільними застосунками. Вона підтримує синхронізацію даних у реальному часі, дозволяє легко оновлювати інформацію на всіх пристроях користувачів одночасно та має вбудовані механізми безпеки. Головним мінусом можна вважати залежність від сервісів Google та поступове збільшення витрат у разі значного росту обсягів даних[6].

Таблиця 1.3

Порівняльна характеристика баз даних за ключовими технічними параметрами

База даних	Тип	Продуктивність	Простота інтеграції	Масштабованість	Найкраще підходить для
MySQL	SQL	Висока	Середня	Висока	Табличні дані, класичні довідники
PostgreSQL	SQL	Дуже висока	Складніша	Висока	Великі проєкти з аналітикою
MongoDB	NoSQL (документи)	Висока	Середня	Висока	Геодані, маршрути, мультимедіа
Firebase Firestore	NoSQL (хмара)	Висока	Дуже проста	Висока	Мобільні застосунки, синхронізація онлайн

Як видно з таблиці, кожне рішення має свої сильні сторони. MySQL та PostgreSQL добре працюють із даними, які мають чітку табличну структуру. MongoDB дозволяє ефективно працювати з геолокацією та мультимедіа, але вимагає більше часу на інтеграцію. Натомість Firebase Firestore відзначається максимальною простотою використання у мобільних додатках, що особливо важливо для нашого проєкту.

Оскільки наш туристичний застосунок міста Луцьк буде орієнтований на роботу в реальному часі та вимагатиме швидкого доступу до інформації про пам'ятки, маршрути та відгуки, вибір саме Firestore виглядає найбільш доцільним. Його хмарна структура дозволить уникнути складних налаштувань серверів, а готові інструменти синхронізації забезпечать актуальність даних для всіх користувачів одночасно. Саме тому оптимальним вибором стала **Firestore** - хмарна NoSQL база даних від Google.

Її основні переваги:

- легка інтеграція з Flutter (наш обраний фреймворк клієнтської частини);
- автоматична синхронізація даних між усіма користувачами;
- гнучка структура (колекції та документи), що дозволяє зберігати маршрути, списки пам'яток, фото та інші об'єкти;
- вбудовані механізми безпеки та авторизації;
- можливість масштабування без додаткових зусиль зі сторони розробників;

Для ефективної роботи мобільного туристичного застосунку структура даних у Firestore організована за допомогою **колекцій** та **документів**. Кожна колекція містить набір документів, які відповідають певному типу інформації.

- **Колекція places (пам'ятки)**: зберігає інформацію про туристичні об'єкти. Поля: id, назва, опис, координати, категорія, фотоURL. *Приклад*: документ «Замок Любарта» → назва: «Замок Любарта», координати: 50.739, 25.335, опис, посилання на фото.
- **Колекція routes (маршрути)**: містить дані про туристичні маршрути. Поля: id, назва, список_точок, тривалість, рівень_складності. *Приклад*: «Історичний центр Луцька» → включає 6 точок, середня тривалість — 2 години.
- **Колекція events (події/заходи)**: описує актуальні культурні події та фестивалі. Поля: id, назва, дата, місце, опис, афішаURL. *Приклад*:

Фестиваль «Ніч у Луцькому замку» → дата: 12 серпня, опис, афіша.

- **Колекція users (користувачі):** використовується для персоналізації та налаштувань. Поля: id, ім'я, email, список_вибраного, налаштування. *Приклад:* користувач може зберігати у «Вибраному» улюблені пам'ятки чи маршрути.

Таким чином, вибір бази даних став ключовим етапом у створенні клієнтської частини туристичного мобільного додатку міста Луцьк. Порівняння різних варіантів показало, що найкращим рішенням є **Firebase Firestore**. Ця база поєднує простоту, швидкість, безпечність і гнучкість, що робить її оптимальною для мобільного додатку, який працює в реальному часі та орієнтований на туристів.

Висновок до розділу 1

У першому розділі було здійснено детальний аналіз технічних рішень, які можуть бути використані для розробки туристичного мобільного застосунку, та обґрунтовано вибір оптимального стеку технологій. Проведене дослідження дозволило оцінити різні варіанти з погляду їх продуктивності, масштабованості, простоти інтеграції та відповідності специфіці майбутнього продукту.

На рівні **серверної частини** були розглянуті найбільш поширені фреймворки та середовища: Node.js, Django та Spring Boot. Django вирізняється високим рівнем захищеності та чіткою структурою, однак його інтеграція з мобільними додатками є складнішою, а розробка потребує більше часу. Spring Boot забезпечує масштабованість та стабільність у великих корпоративних проєктах, проте є складним у налаштуванні. У свою чергу, Node.js із використанням фреймворку Express продемонстрував оптимальне поєднання продуктивності, швидкості розробки та гнучкості. Асинхронна модель обробки запитів та підтримка великої кількості одночасних підключень роблять його найкращим вибором для нашого застосунку, де важлива швидка взаємодія з користувачем.

Щодо **клієнтської частини**, було порівняно нативні підходи (Kotlin/Java для Android, Swift для iOS) та кросплатформні фреймворки (React Native, Flutter). Нативна розробка хоч і забезпечує максимальну продуктивність та повний доступ до функцій пристрою, проте вимагає створення двох окремих додатків, що ускладнює підтримку і збільшує витрати. React Native дозволяє створювати кросплатформні рішення, проте має обмеження щодо складних інтерфейсів. Flutter, натомість, відзначається високою швидкістю, єдиним кодом для двох платформ, сучасним підходом до побудови інтерфейсів і великою спільнотою розробників. Крім того, можливість швидкого тестування через Hot Reload робить його надзвичайно зручним інструментом. Саме тому Flutter обрано як основу клієнтської частини.

При виборі **системи управління базами даних** було проаналізовано реляційні (MySQL, PostgreSQL) та нереляційні (MongoDB, Firebase Firestore) рішення. MySQL добре підходить для класичних структурованих даних, проте менш гнучкий у роботі з мультимедійними та динамічними об'єктами. PostgreSQL забезпечує високу надійність та складні запити, але є більш ресурсомістким і складним у налаштуванні. MongoDB дозволяє працювати з гнучкими структурами, включно з геоданими, однак потребує додаткових налаштувань для мобільної інтеграції. Найкращим вибором для нашого проєкту стала хмарна база даних **Firebase Firestore**, яка забезпечує синхронізацію у реальному часі, просту інтеграцію з мобільним застосунком, вбудовані інструменти захисту даних та масштабованість. Саме ці характеристики є ключовими для туристичного застосунку, що працює з картами, маршрутами та подіями.

Таким чином, обґрунтований технічний стек майбутнього застосунку виглядає наступним чином:

- **Node.js (Express)** - серверна частина, що забезпечує швидкий і надійний обмін даними;
- **Flutter** - клієнтська частина, яка гарантує кросплатформність та

зручний інтерфейс;

- **Firebase Firestore** - система управління даними, що підтримує масштабованість, синхронізацію та безпеку.

Обрані технології не лише задовольняють функціональні вимоги, а й відкривають широкі можливості для подальшого розвитку застосунку. Завдяки такому підходу продукт матиме конкурентоспроможність, високу якість обслуговування користувачів і перспективи масштабування в майбутньому.

РОЗДІЛ 2

МЕТОДОЛОГІЯ РОЗРОБКИ

2.1. Порівняння методологій розробки

У сучасній індустрії розробки програмного забезпечення вибір методології є одним із ключових рішень, що визначає успіх усього проєкту. Методологія задає рамки та правила, за якими працює команда, визначає послідовність етапів створення продукту, підхід до управління ресурсами, комунікації та контролю якості. Вона безпосередньо впливає на те, наскільки швидко і гнучко команда реагує на зміни, чи здатна враховувати потреби користувачів у процесі створення продукту та наскільки стабільним буде кінцевий результат.

Особливо актуальним питання вибору методології є у сфері розробки мобільних застосунків. Цей ринок характеризується високою динамікою: технології швидко змінюються, з'являються нові сервіси, а користувачі очікують від додатків не лише стабільності, а й регулярних оновлень із новим функціоналом. Тому розробники повинні працювати у таких умовах, де є потреба в гнучкості, швидкому прийнятті рішень та активному зворотному зв'язку від аудиторії[9].

У практиці ІТ-індустрії застосовуються як **традиційні моделі**, що орієнтовані на чітку послідовність дій, так і **гнучкі методології**, що акцентують увагу на швидкій адаптації та співпраці з користувачами.

Традиційні підходи: каскадна модель

Каскадна модель (Waterfall) вважається класичним і найбільш формалізованим підходом до розробки програмного забезпечення. Вона виникла ще у 70-х роках ХХ століття і тривалий час була єдиним офіційно описаним методом, який використовували великі компанії та державні установи. Головна ідея Waterfall полягає у суворій послідовності виконання етапів: від збору вимог і проєктування до тестування та впровадження. Всі кроки йдуть «каскадом»,

тобто завершення одного етапу є обов'язковою передумовою для початку наступного.

У класичному варіанті модель включає п'ять основних фаз:

1. Аналіз вимог - на цьому етапі замовник разом із командою детально описує всі функціональні й нефункціональні вимоги до майбутньої системи. Результатом є технічне завдання.
2. Проектування - створення архітектури, моделювання даних, проектування інтерфейсів та розробка детальних планів реалізації.
3. Реалізація - програмісти починають написання коду відповідно до документації. Цей етап зазвичай займає найбільше часу.
4. Тестування - готова система перевіряється на відповідність вимогам. На цьому етапі виявляються помилки, які виправляються до релізу.
5. Впровадження та супровід - продукт передається користувачам, а команда займається його підтримкою.

Перевагою каскадної моделі є її структурованість і прогнозованість: замовник заздалегідь знає, що отримає у кінці, а менеджери мають змогу планувати бюджет і терміни з високою точністю. Крім того, Waterfall дає можливість створювати детальну документацію, що важливо у великих організаціях і проєктах, які потребують довготривалої підтримки[10].

Однак цей підхід має й суттєві обмеження. Його головний недолік - відсутність гнучкості. Якщо під час розробки виникають нові потреби користувачів або замовник хоче змінити функціонал, інтегрувати ці зміни практично неможливо без переробки значної частини системи. Також у Waterfall немає можливості раннього тестування на користувачах: продукт стає доступним тільки після завершення всіх етапів. Це створює ризик того, що кінцевий результат може не відповідати очікуванням.

Не випадково каскадна модель здобула популярність у сферах, де зміна вимог є мінімальною або небажаною. Наприклад, у сфері оборони, авіації чи космічних досліджень, де будь-яка помилка може мати критичні наслідки. Саме

тому NASA при створенні програмного забезпечення для космічних апаратів використовує Waterfall. Там надзвичайно важливо, щоб усі вимоги були чітко задокументовані й реалізовані без відхилень, а будь-яке тестування проводиться тільки після завершення кодування.

У контексті проєкту «MyLutsk» каскадна модель виглядає менш придатною. Причина полягає в тому, що розробка мобільного застосунку для туристів — це процес, який постійно еволюціонує. Вимоги користувачів ще формуються: під час тестування прототипу можуть з'явитися нові ідеї (наприклад, інтеграція з аудіогідами, доповнена реальність для маршрутів, розширені налаштування для іноземних туристів). У каскадній моделі внесення подібних змін було б надзвичайно дорогим і затягнутим, адже довелося б повертатися на попередні етапи й перебудовувати систему.

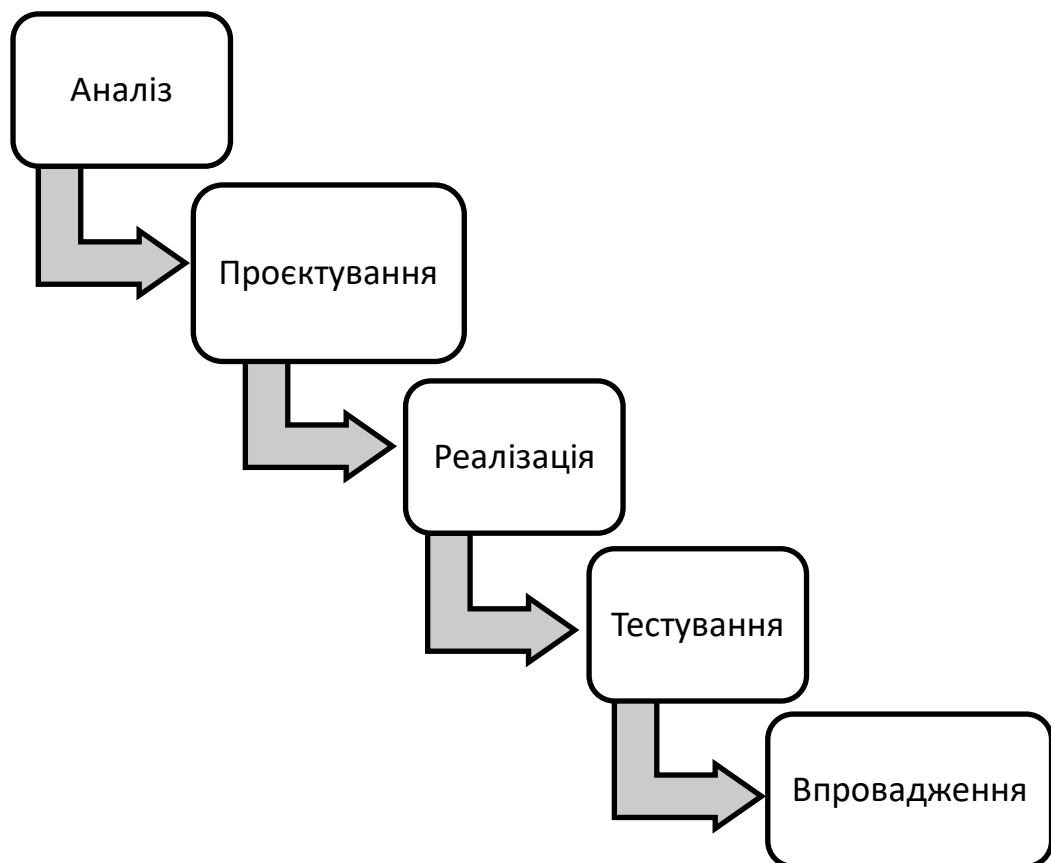


Рис. 2.1 Каскадна модель (Waterfall)

Джерело: розроблено автором

Таким чином, Waterfall можна вважати надійним і перевіреним підходом для великих консервативних систем, однак для гнучких і динамічних мобільних продуктів, таких як «MyLutsk», більш доцільним буде застосування сучасних гнучких методологій, які дозволяють швидко реагувати на потреби ринку та вдосконалювати продукт поступово.

Традиційні підходи: ітераційна модель

Ітераційна модель стала логічним продовженням каскадного підходу і виникла як спроба подолати його головний недолік - відсутність гнучкості. Якщо у Waterfall продукт створюється один раз, наприкінці всіх етапів, то ітераційна модель дозволяє поступово вдосконалювати систему через серію повторюваних циклів (ітерацій).

Кожна ітерація є міні-проєктом, який містить етапи аналізу, проєктування, реалізації та тестування. Результатом кожної ітерації є робочий прототип або версія системи з обмеженим, але функціональним набором можливостей. Таким чином, замовник отримує можливість ознайомитися з продуктом на ранньому етапі, оцінити його якість і надати зворотний зв'язок, а команда може швидко вносити зміни, ґрунтуючись на реальних потребах.

Основні характеристики ітераційної моделі:

- Поступовість. Продукт створюється невеликими кроками, які послідовно розширюють функціонал.
- Гнучкість. Легко інтегрувати нові вимоги чи змінювати старі.
- Прозорість. Замовник бачить проміжні результати після кожної ітерації.
- Вищі витрати на планування. Потребує постійної координації між усіма учасниками процесу.

Ітераційна модель активно застосовувалася у великих ІТ-компаніях у 80-90-х роках. Наприклад, Microsoft розробляла нові версії Windows саме ітераційно. Кожна версія ОС спиралася на попередню, вдосконалюючи інтерфейс, додаткові функції та систему безпеки. Аналогічно працювали й

компанії на кшталт IBM, що створювали корпоративні системи з регулярними релізами.

Цей підхід дозволяв компаніям не чекати роками на повністю готовий продукт, а вже через кілька місяців випускати проміжні версії, які могли використовуватися реальними користувачами.

Для туристичного мобільного додатку «MyLutsk» ітераційний підхід виглядає доволі привабливим. Він дозволяє почати з невеликого набору функцій (наприклад, базова карта з позначеними пам'ятками), а потім поступово додавати маршрути, систему «Вибране», відображення культурних подій, офлайн-режим чи навіть інтеграцію з аудіогідами. Таким чином, уже після першої ітерації користувачі могли б тестувати робочу версію додатку, а команда отримувала б цінний зворотний зв'язок для подальшого розвитку. Це мінімізує ризики невідповідності кінцевого продукту очікуванням і робить розробку більш гнучкою та прозорою.

Разом із тим, ітераційний підхід вимагає більшої кількості ресурсів для управління проєктом, оскільки кожна ітерація потребує ретельного планування й тестування. Тому для невеликих команд він може створювати додаткове навантаження.

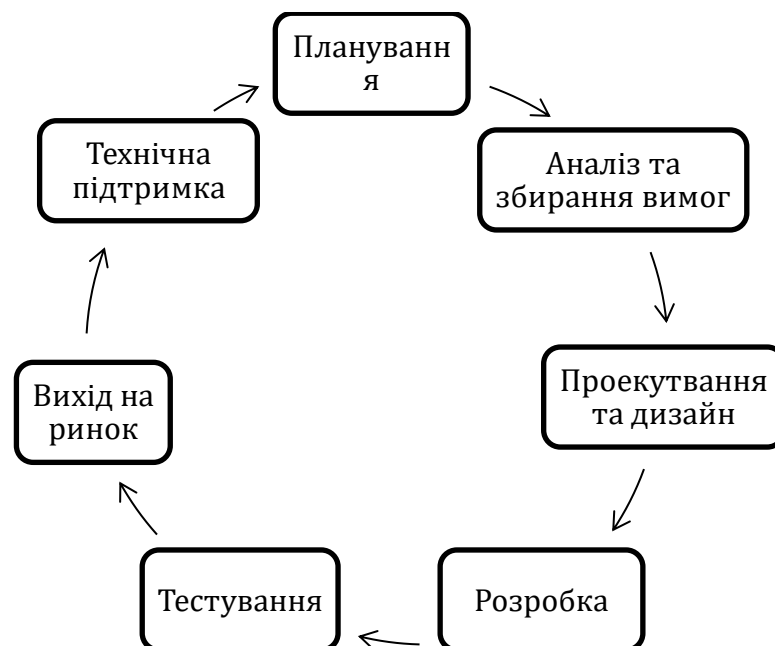


Рис 2.2 Ітераційна модель розробки

Джерело: розроблено автором

Отже, ітераційна модель поєднує структурованість із більшою гнучкістю, ніж Waterfall, і може слугувати хорошим компромісом для команд, які хочуть уникнути жорстких рамок, але водночас мати контроль над етапами розробки.

Agile

Agile є сучасним підходом до організації процесу розробки програмного забезпечення, який виник як альтернатива традиційним моделям, що базувалися на жорсткому плануванні. Якщо у каскадній і навіть ітераційній моделі акцент робився на суворій послідовності етапів і детальній документації, то в Agile у центрі уваги знаходяться **користувач і його потреби**, а також **гнучкість у прийнятті рішень**.

У 2001 році група провідних інженерів сформулювала так званий **Маніфест гнучкої розробки програмного забезпечення (Agile Manifesto)**, що став основою сучасних гнучких методологій. У ньому викладені чотири ключові принципи:

- люди та взаємодія важливіші за процеси й інструменти;
- робочий продукт важливіший за детальну документацію;
- співпраця із замовником важливіша за контрактні переговори;
- готовність до змін важливіша за слідування початковому плану.

Ці принципи докорінно змінили уявлення про розробку програмних продуктів. Якщо раніше створення програмного забезпечення могло тривати роками, а результат часто не відповідав реальним потребам користувачів, то Agile дозволив робити це **поетапно**: швидко випускати перші версії, отримувати відгуки й на їх основі вдосконалювати продукт. Це зробило розробку більш динамічною, живою та орієнтованою на результат.

Scrum є однією з найпопулярніших методологій Agile. Він ґрунтується на роботі невеликих команд, які організовують діяльність у вигляді **спринтів** - коротких циклів тривалістю від двох до чотирьох тижнів. Наприкінці кожного

спринту команда повинна надати **інкремент продукту** - робочу частину системи, яка вже може використовуватися.

Особливістю Scrum є його **структурованість**. Тут чітко визначені ролі:

- **Product Owner** відповідає за бачення продукту і визначає пріоритети завдань.
- **Scrum Master** контролює процес і допомагає усувати перешкоди.
- **Команда розробників** реалізує функціонал.

Для організації роботи у Scrum застосовуються спеціальні артефакти:

- **Backlog продукту** - загальний список усіх вимог і завдань.
- **Backlog спринту** - підмножина завдань, які виконуються саме у цьому спринті.
- **Інкремент** - готовий результат, створений за підсумками спринту.

Процес супроводжується регулярними зустрічами: плануванням спринту, щоденними короткими обговореннями (stand-up), демонстрацією результатів і ретроспективою, де команда аналізує власну роботу.

Компанія **Spotify** активно використовує Scrum, поділяючи команди на невеликі автономні «скводи» (squads). Кожен із них відповідає за окрему частину функціоналу, наприклад рекомендаційні алгоритми чи плейлісти. Така організація дозволяє регулярно випускати нові можливості, які відразу тестуються мільйонами користувачів.

У **«MyLutsk»**: Scrum підходить як основна методологія для створення застосунку. Наприклад, у першому спринті команда може реалізувати карту з пам'ятками, у другому - додати маршрути та функцію «Вибране», у третьому - інтегрувати календар подій, а в четвертому - впровадити офлайн-режим. Кожна версія додатку виходила б у світ із новими можливостями, що дає змогу туристам користуватися ним ще на етапі розробки, а команді - отримувати живий відгук.

Kanban - це інший, не менш ефективний підхід у межах Agile, який акцентує увагу на **безперервності роботи та прозорості процесів**. Він менш

формалізований, ніж Scrum, і більше орієнтований на гнучке управління завданнями.

Основним інструментом Kanban є **дошка завдань**, яка зазвичай складається з трьох колонок:

- «**Заплановано**» - завдання, які очікують виконання;
- «**У процесі**» - завдання, над якими зараз працюють;
- «**Завершено**» - виконані завдання.

Завдання подаються у вигляді карток і переміщуються дошкою залежно від їхнього стану. Така наочна візуалізація дозволяє команді бачити реальний прогрес, уникати перевантаження та швидко змінювати пріоритети.

Метод Kanban уперше був застосований у виробництві компанією **Toyota**, а згодом його перейняли ІТ-компанії. Сьогодні Kanban широко використовується у таких сервісах, як **Trello**, який побудований саме на цій системі.

У «MyLutsk»: Kanban доцільно застосовувати для управління оновленням контенту. Наприклад, завдання «додати опис нової події», «завантажити фотографії пам'ятки» чи «оновити розклад маршруту» можуть легко відслідковуватися на дошці. Це дає можливість оперативно реагувати на зміни та забезпечувати актуальність даних для туристів.

Попри те, що обидві методології належать до Agile, вони мають різні сфери застосування:

- **Scrum** орієнтується на короткі цикли (спринти), тому найкраще підходить для створення нового функціоналу і поступового розвитку продукту.
- **Kanban** забезпечує безперервний потік роботи, що робить його оптимальним для підтримки та регулярних оновлень.

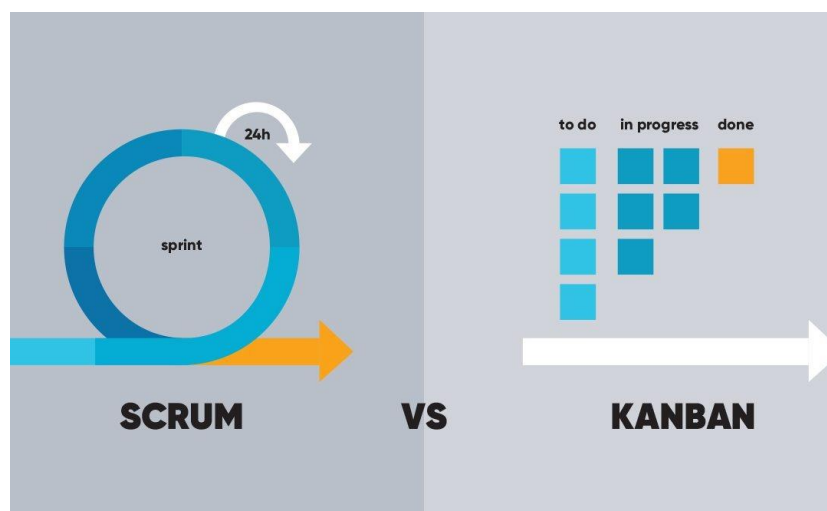


Рис 2.3 Порівняння Scrum і Kanban [<https://pmtips.com.ua/post/scrum-abo-kanban-shcho-vybraty-dlia-vashoyi-komandy>]

Для проєкту «MyLutsk» оптимальним є поєднання двох методів: Scrum дозволить плановірно реалізувати ключові функції застосунку (карти, маршрути, події), а Kanban забезпечить безперервне оновлення контенту (нові події, афіші, описи пам'яток). Такий підхід створить баланс між швидкістю розробки, гнучкістю та стабільністю.

Для кращого розуміння особливостей різних методологій було складено порівняльну таблицю, у якій наведені їхні переваги, недоліки, та можливості застосування у проєкті «MyLutsk».

Таблиця 2.1

Порівняльна характеристика методологій розробки програмного забезпечення

Метедологія	Переваги	Недоліки	Використання у «MyLutsk»
Waterfall	Структурованість, чітка документація	Відсутність гнучкості, складність змін	Недоцільна: вимоги користувачів змінюються
Ітераційна	Проміжні результати, раннє тестування	Складність координації	Поетапна розробка: спершу пам'ятки, потім маршрути й події
Scrum	Гнучкість, прозорість, швидкі релізи	Потребує високої дисципліни	Найкраще рішення: спринти з тестуванням на туристах

Kanban	Прозорість, контроль завантаженості	Відсутність чітких строків	Оптимально для підтримки й оновлення контенту
---------------	---	-------------------------------	---

Аналіз показує, що для проєкту «*MyLutsk*» найбільш ефективним є поєднання Scrum і Kanban:

- Scrum стане основною методологією для розробки нового функціоналу, адже дозволяє створювати додаток поступово, з регулярним тестуванням на реальних користувачах.
- Kanban доцільно застосовувати для управління поточними завданнями - оновленням подій, афіш і описів пам'яток, що забезпечить актуальність інформації у додатку.

Таким чином, комбіноване використання Scrum і Kanban створює баланс між гнучкістю, швидкістю розробки та стабільністю оновлень, що відповідає завданням проєкту «*MyLutsk*».

2.2. Особливості впровадження змін на основі відгуку користувача

У сучасній розробці програмного забезпечення роль користувачів не обмежується лише використанням готового продукту. Вони стають активними учасниками процесу його вдосконалення. Саме завдяки їхнім відгукам команди розробників отримують цінну інформацію про зручність інтерфейсу, необхідність нових функцій, а також проблеми, з якими стикаються реальні користувачі у щоденному застосуванні програми.

Особливо важливо враховувати відгук у мобільних туристичних додатках, таких як «*MyLutsk*». Туристи - це аудиторія з різноманітними потребами: хтось шукає культурні пам'ятки, хтось - актуальні події, інші потребують офлайн-карт чи багатомовної підтримки. Врахування цих вимог дає змогу зробити застосунок більш гнучким та універсальним, що напряму впливає на рівень задоволеності користувачів і загальну конкурентоспроможність продукту[11].

У цьому контексті важливим є використання принципів Agile-розробки. На відміну від традиційних методологій, які передбачають внесення змін лише після завершення проєкту, Agile дозволяє швидко реагувати на фідбек, інтегруючи його у кожен нову ітерацію. Завдяки цьому продукт постійно розвивається та вдосконалюється, зберігаючи актуальність і відповідність очікуванням користувачів.

Таким чином, упровадження змін на основі відгуку користувача у проєкті «*MyLutsk*» не є додатковим завданням, а стає невід’ємною частиною розробки. Це забезпечує не лише технічний розвиток застосунку, а й створює основу для формування позитивного досвіду користувача, підвищення його лояльності та залучення нових туристів.

Для того щоб мобільний додаток залишався зручним та актуальним, розробникам необхідно постійно отримувати інформацію про те, як ним користуються реальні люди. У випадку з туристичним застосунком «*MyLutsk*» це має особливе значення, адже потреби туристів є динамічними: змінюються маршрути, з’являються нові події, а користувачі мають різний досвід і рівень цифрових навичок. Саме тому важливо передбачити кілька каналів збору відгуків, які дадуть змогу комплексно оцінювати роботу програми[12].

1. Аналітика використання додатку.

За допомогою сервісів **Firebase Analytics** чи **Google Analytics** можна збирати статистику про поведінку користувачів: які екрани вони відкривають найчастіше, скільки часу проводять у додатку, які маршрути обирають. Ці дані дозволяють зрозуміти, які функції користуються найбільшим попитом, а які потребують доопрацювання. Для «*MyLutsk*» це означає можливість виявляти найцікавіші туристичні точки й визначати слабкі місця інтерфейсу[13].

2. Форма зворотного зв’язку в додатку.

Користувач може безпосередньо залишити свій коментар або оцінку щодо конкретної пам’ятки чи маршруту. Це забезпечує якісний зворотний зв’язок - думки й пропозиції реальних людей, які щойно скористалися функціоналом.

Такий інструмент ефективно працює у багатьох сервісах, наприклад у Booking.com чи Airbnb, де користувачі залишають відгуки про готелі чи квартири. У «MyLutsk» це можна реалізувати через просту форму: «Оцініть маршрут» або «Чи була ця інформація корисною?».

3. Відгуки у магазинах додатків (App Store та Google Play).

Рейтинги й коментарі в офіційних маркетах є одним із найцінніших каналів. По-перше, вони впливають на загальне сприйняття застосунку новими користувачами. По-друге, коментарі часто містять конструктивні поради: що працює добре, а що потребує вдосконалення. Наприклад, у сервісі Uber значна частина розвитку функцій ґрунтувалася на аналізі саме таких відгуків. Для «MyLutsk» це стане джерелом незалежного фідбеку від широкої аудиторії.

4. Соціальні мережі та онлайн-спільноти.

Багато користувачів не пишуть відгуки у самому додатку, але активно діляться думками у Facebook, Telegram-каналах чи туристичних форумах. Цей канал є менш структурованим, але дозволяє вловлювати загальні тенденції й «живий» досвід людей. Для «MyLutsk» це може бути моніторинг туристичних груп Луцька, де користувачі діляться враженнями від поїздок.



Рис. 2.4 Канали збору відгуків користувачів

Джерело: розроблено автором

Використання різних каналів збору відгуків дозволяє побачити картину з різних боків: кількісні дані аналітики доповнюються якісними коментарями користувачів, а офіційні відгуки в маркетах - живим спілкуванням у соцмережах.

Впровадження змін на основі відгуків користувачів - це безперервний цикл, що дозволяє команді оперативно реагувати на потреби аудиторії та вдосконалювати продукт. Для мобільного додатку «*MyLutsk*» цей процес є особливо важливим, адже його успіх напряму залежить від того, наскільки швидко враховуються нові вимоги туристів[14].

Все починається зі збору даних: аналітика (наприклад, *Firebase Analytics*) показує статистику використання функцій і маршрутів, тоді як форми зворотного зв'язку чи коментарі в соцмережах дають якісні відгуки від реальних користувачів. У *MyLutsk* це може відбуватися автоматично - за допомогою *Firebase Analytics*, де фіксується, які маршрути переглядають найчастіше, скільки часу користувачі проводять на сторінках пам'яток. Водночас через вбудовану форму зворотного зв'язку турист може залишити оцінку чи коментар («Чи був цей маршрут зручним?», «Чи була інформація достатньою?»).

Далі відбувається аналіз зібраної інформації. Зібрані дані перевіряються на повторюваність. Наприклад, якщо багато користувачів пишуть, що карти не працюють офлайн, це сигнал до пріоритетного вдосконалення. Аналогічно, якщо більшість користувачів відвідують одні й ті ж маршрути, можна розширити категорії та створити нові добірки.

Після цього настає етап пріоритизації, коли команда визначає, які зміни слід реалізувати в першу чергу, а які можна відкласти. Це забезпечує баланс між швидкістю розвитку і якістю функціоналу. У *MyLutsk* команда може віднести запит на офлайн-мапи до завдань найвищого пріоритету, оскільки він критично впливає на досвід туриста. Натомість менш термінові запити, як-от «додати доповнену реальність (AR) для замків», можуть бути заплановані на наступні релізи.

Далі відбувається планування: у Scrum завдання потрапляють у backlog спринту, а в Kanban - на дошку в колонку «Заплановано». Це дозволяє організувати роботу максимально прозоро.

Наступний крок - реалізація та тестування. Тут створюються нові функції або виправляються помилки, після чого все проходить як технічне, так і юзабіліті-тестування, щоб гарантувати якість кінцевого результату. Наприклад, додається можливість фільтрувати маршрути за рівнем складності - і далі проводиться тестування з реальними користувачами (туристи можуть протестувати цю функцію під час подорожі)[15].

Завершальним етапом є оновлення, коли нова версія застосунку виходить у Google Play чи App Store. Туристи отримують оновлений функціонал, бачать враховані пропозиції і залишають нові коментарі. Після цього цикл починається знову: користувачі тестують новий функціонал, залишають відгуки, і процес повторюється, забезпечуючи безперервний розвиток продукту.

Вивчення досвіду провідних компаній дає змогу зрозуміти, наскільки важливим є системне врахування відгуків користувачів у процесі розвитку мобільних сервісів. Багато глобальних застосунків довели, що саме фідбек стає основним рушієм змін, і приклади цього добре ілюструють можливості, які можуть бути реалізовані й у «MyLutsk».



Рис. 2.5 Приклади застосунків

[<https://www.dreamstime.com/illustration/booking-airbnb-logo.html>]

Duolingo:

Цей освітній мобільний застосунок є одним із найяскравіших прикладів використання відгуків для розвитку продукту. Команда Duolingo постійно

впроваджує А/Б-тести, щоб перевірити, яка версія інтерфейсу чи функції краще сприймається користувачами. Наприклад, перед додаванням нових елементів гейміфікації (бонусні життя, рівні, нагороди) розробники тестують кілька варіантів і залишають лише той, що показав найвищий рівень залученості. Крім того, коментарі в App Store та Google Play регулярно стають джерелом ідей для покращень. У випадку «*MyLutsk*» подібний підхід можна застосувати, тестуючи різні способи відображення маршрутів чи категоризації пам'яток.

Airbnb:

Спочатку цей сервіс був орієнтований переважно на бронювання житла. Однак численні запити користувачів на унікальні туристичні активності - кулінарні майстер-класи, екскурсії з місцевими жителями, творчі воркшопи - змусили команду розширити функціонал. Так з'явився розділ «Досвіди» (Experiences), що перетворив Airbnb на комплексну платформу для подорожей. Це приклад того, як масові відгуки користувачів формують новий напрямок розвитку продукту. Для «*MyLutsk*» аналогом може бути додавання інтерактивних аудіогідів чи віртуальних екскурсій, якщо користувачі активно цього вимагатимуть.

Google Maps:

Один із найбільш відомих сервісів для навігації та пошуку локацій, який активно використовує відгуки та контент від самих користувачів. Можливість додавати рейтинг, фото та коментарі до закладів чи пам'яток зробила цей сервіс максимально зручним і практичним. Завдяки цьому Google Maps постійно оновлюється найактуальнішою інформацією, яку надають самі користувачі. Для «*MyLutsk*» це може стати натхненням для впровадження функції «Відгуки про маршрути» або «Фото від туристів», що збільшить залученість і зробить контент більш живим та достовірним.

Таким чином, приклади Duolingo, Airbnb та Google Maps доводять: успіх мобільного додатку залежить не лише від початкової ідеї, а й від здатності команди постійно інтегрувати зворотний зв'язок користувачів у розвиток

продукту. Для «*MyLutsk*» це означає потребу створити механізми швидкої реакції на запити туристів, що дозволить перетворити додаток на сучасний, гнучкий і затребуваний сервіс.

Практичне застосування відгуків користувачів у проєкті «*MyLutsk*» може суттєво вплинути на якість сервісу та рівень задоволеності туристів. Кожен зафіксований коментар або повторюваний запит може трансформуватися у конкретну функцію чи вдосконалення.

По-перше, якщо користувачі вказують на складність пошуку маршрутів, доцільно впровадити систему фільтрації за категоріями. Туристи зможуть обирати маршрути відповідно до своїх інтересів: історичні пам'ятки, культурні об'єкти чи гастрономічні точки. Це не лише підвищить зручність використання, а й допоможе індивідуалізувати туристичний досвід.

По-друге, серед поширених потреб є офлайн-режим. Туристи часто залишаються без доступу до інтернету під час подорожей. Реалізація цього запиту можлива завдяки інтеграції Mapbox та системи кешування маршрутів. Таким чином, навіть у «мертвих зонах» користувач зможе прокласти маршрут чи переглянути опис пам'ятки.

По-третє, важливим елементом є багатомовність. Якщо відгуки свідчать про необхідність розширення мовної підтримки, слід реалізувати локалізацію принаймні трьома мовами: українською, англійською та польською. Це зробить додаток доступним не лише для жителів Луцька, а й для міжнародних туристів, що активно відвідують місто.

По-четверте, відгуки користувачів часто стосуються складнощів у реєстрації та вході. Щоб уникнути бар'єрів на цьому етапі, слід інтегрувати спрощену авторизацію через Google або Facebook. Це дозволить користувачам швидко створювати профілі та переходити до використання додатку без зайвих затримок.

Таким чином, кожен відгук має потенціал стати основою для покращення функціоналу «*MyLutsk*». Реагування на такі пропозиції створює у користувачів

відчуття залученості до розвитку продукту та підвищує рівень їхньої довіри до сервісу.

Упровадження змін на основі відгуків користувачів є критично важливим для довгострокового успіху «*MyLutsk*». Постійний цикл «**зворотний зв'язок** → **аналіз** → **реалізація** → **оновлення**» дає змогу підтримувати додаток актуальним, інклюзивним і зручним. Це створює сильну конкурентну перевагу, адже саме користувач визначає, які функції будуть корисними, і формує попит на нові можливості. У результаті «*MyLutsk*» може стати не просто довідковим ресурсом, а повноцінним інтерактивним помічником для туристів[16].

Висновок до розділу 2

У другому розділі було розглянуто та порівняно основні методології розробки програмного забезпечення, що використовуються на сучасному етапі розвитку ІТ-індустрії, а також визначено їх доцільність для створення туристичного мобільного застосунку «**MyLutsk**». Аналіз каскадної, ітераційної та гнучких методологій показав, що традиційні підходи, зокрема модель Waterfall, характеризуються високим рівнем структурованості та передбачуваності, проте не забезпечують достатньої адаптивності до змін у вимогах користувачів. Це обмежує їх ефективність у проєктах, що мають динамічний характер та орієнтовані на поступове вдосконалення функціоналу.

Ітераційна модель розробки частково вирішує проблему гнучкості, оскільки дозволяє формувати продукт поетапно, отримуючи проміжні результати та коригуючи напрям подальшої розробки відповідно до відгуків користувачів.

Гнучкі підходи, зокрема методології Agile, Scrum та Kanban, виявилися найбільш відповідними для реалізації проєкту «*MyLutsk*». Scrum забезпечує ефективну організацію процесу розробки через поділ роботи на короткі спринти з постійним удосконаленням функціоналу, контролем пріоритетів та активною взаємодією з користувачами. Kanban, своєю чергою, є доцільним інструментом

для управління поточними завданнями, зокрема щодо оновлення контенту та підтримки актуальності інформації у додатку.

Особливе значення в рамках методології має зворотний зв'язок із користувачем, який виступає основним джерелом для удосконалення інтерфейсу та функціональних можливостей. Інтеграція відгуків у процес розробки дозволяє адаптувати продукт до реальних потреб аудиторії, забезпечує його конкурентоспроможність і сприяє підвищенню рівня користувацької задоволеності.

Отже, оптимальним підходом для реалізації проєкту стало поєднання методологій Scrum та Kanban, що забезпечує баланс між гнучкістю розробки, можливістю оперативного реагування на зміни, структурованістю процесу та стабільністю подальшого супроводу додатку.

РОЗДІЛ 3

ДИЗАЙН

3.1. Вибір візуальної стилістики

Візуальна стилістика мобільного додатку відіграє ключову роль у формуванні користувацького досвіду, адже саме зовнішній вигляд інтерфейсу, кольорова система, композиції елементів та типографіка визначають, наскільки комфортним і зрозумілим буде взаємодія користувача з продуктом. У туристичних мобільних додатках це має особливе значення, оскільки користувачі взаємодіють із застосунком у русі, часто в незнайомому середовищі, з обмеженим часом для прийняття рішень. Саме тому дизайн додатку «MyLutsk» був розроблений із фокусом на три основні аспекти: простота інтерфейсу, емоційна привабливість та зв'язок із локальною культурою міста Луцька[18].

Першочергово постало завдання сформувати візуальну мову, яка не лише виконуватиме інформаційну функцію, але й відображатиме ідентичність міста. Луцьк є містом із глибокою історією, культурною спадщиною й унікальною архітектурою, тому важливо було створити дизайн, який не просто подає інформацію, а формує емоційний досвід відкриття міста. Тому було обрано стилістику, що поєднує сучасний мінімалізм із теплими, м'якими візуальними акцентами, які передають атмосферу гостинності.

Для забезпечення цілісності візуального образу застосунку було створено брендбук - офіційний системний документ, у якому зібрані всі ключові елементи візуальної та комунікаційної айдентики проєкту, а також визначені правила їхнього використання. Брендбук виконує функцію стилістичної основи, що гарантує послідовність оформлення у будь-якому середовищі: в інтерфейсі мобільного додатку, на веб-сторінках, у друкованих та рекламних матеріалах, а також у сувенірній продукції.

У випадку «MyLutsk» брендбук має особливо значущу роль, оскільки додаток виступає цифровим провідником містом. Це не лише інструмент для

навігації або пошуку локацій, а спосіб познайомити користувача з Луцьком через враження, емоції та історичні сенси, що збереглися у просторі міста. Тому візуальна айдентика була побудована так, щоб передавати атмосферу спокою, тепла, відкритості, які є характерними для Луцька як міста зі збереженою історичною пам'яттю та затишною міською ритмікою[23].

Посилання на брендбук:

<https://www.figma.com/design/JGf7OELzOcYXK2YYX124fy/Untitled?node-id=415-78&t=EH3W8wiYs9YVdxX6-1>.

Після визначення концептуальних засад айдентики були сформовані її ключові візуальні компоненти: логотип та його варіанти, основна та акцентна кольорова палітра, шрифтові рішення, правила композиції та використання графічних елементів у різних середовищах. Створення брендбуку дозволило сформуванню цілісного та впізнаваного візуального образу додатку, який не розпадається на окремі частини і зберігається незалежно від контексту застосування.

Логотип є центральним елементом бренд-айдентики мобільного додатку «MyLutsk», оскільки саме він формує перше враження про проєкт і забезпечує миттєву впізнаваність у цифровому та фізичному середовищах. У своїй основі логотип поєднує графічний символ і текстову назву, що утворюють єдину композицію, орієнтовану на простоту, чіткість і збереження змістовності.

Графічна частина логотипу виконана у вигляді стилізованого зображення Замку Любарта - найвідомішої історичної та архітектурної домінанти міста.. Саме тому його включення до айдентики було принциповим рішенням: логотип відразу пов'язує додаток із простором, який він представляє. Зображення замку подане контурно та мінімалістично, без надмірної деталізації. Такий підхід дозволяє зберегти історичний зміст образу, одночасно інтегруючи його у сучасну візуальну систему.

Композиційно логотип побудований таким чином, щоб бути масштабованим і придатним до використання у різних форматах: від великих

рекламних площин до маленьких інтерфейсних значків. Використання простих геометричних форм, обмеженої кількості ліній та чіткої пропорційності забезпечує стійкість графічного образу під час зміни розміру, фону чи носія. Завдяки цьому логотип однаково добре відтворюється на мобільному екрані, у друкованих матеріалах, на сувенірах, у міській навігації та у соціальних мережах.

Важливо також, що логотип був розроблений у кількох варіативних форматах, передбачених брендбуком: повна композиція (графіка + назва), компактна версія (лише графічний символ) та монохромні адаптації. Це дозволяє використовувати логотип гнучко, не втрачаючи впізнаваності, зберігаючи стиль і відчуття бренду незалежно від контексту.



Рис. 3.1. Логотип MY LUTSK

Джерело: розроблено автором

Кольорова палітра є одним із визначальних елементів візуальної айдентики, оскільки саме колір першим формує емоційне враження і задає загальний тон взаємодії користувача з продуктом.

При розробці айдентики «MyLutsk» було обрано спокійну, глибоку та збалансовану палітру, що передає камерність, теплоту та історичну насиченість Луцька. Основу становлять два домінантні кольори - глибокий синій та білий. Вони задають загальний стиль, створюють контраст і забезпечують читабельність у цифровому середовищі. Для створення емоційних акцентів були обрані бірюзовий (#60ADB7) та світло-блакитний (#DAE6EE).

Таким чином, обрана кольорова система працює не як декоративний елемент, а як **функціонально-сміслова структура**, що:

- задає емоційний тон взаємодії,
- підсилює зв'язок із культурним середовищем міста,
- забезпечує комфортне сприйняття інформації,
- формує стабільні та впізнавані асоціації з брендом.

Палітра допомагає створити атмосферу, у якій користувач не просто отримує інформацію, а «проживає» знайомство з містом, сприймаючи його теплота, спокій і глибину через колір і візуальний ритм.



Рис. 3.2. Кольорова палітра MY LUTSK

Джерело: розроблено автором

Типографіка посідає особливе місце у візуальній айдентиці, оскільки шрифт не лише передає текстову інформацію, але й створює емоційний тон комунікації, визначає стиль спілкування додатку з користувачем та впливає на відчуття ритму і комфорту під час читання. Важливо, щоб шрифти гармонійно доповнювали загальну стилістику бренду, були функціональними у цифровому середовищі та підсилювали характер міста, який прагне передати додаток.

У бренд-айдентиці «MyLutsk» було свідомо обрано поєднання двох шрифтів, кожен із яких виконує свою окрему роль у структурі візуального сприйняття:

- **SF Compact** - як основний шрифт інтерфейсу;
- **Caveat** - як акцентний, емоційно-проявний шрифт.

Основним шрифтом став **SF Compact**, оскільки він вирізняється чіткістю, геометричною впорядкованістю і високою читабельністю на мобільних екранах. Його форми лаконічні, врівноважені, без декоративних перевантажень, що дозволяє користувачу зосередитися на змісті.

Акцентним шрифтом було обрано **Caveat** - рукописний, м'який і емоційно теплий. Caveat з'являється в коротких висловах, заголовках або емоційних мікротекстах.

У підсумку шрифтове рішення в айдентиці «MyLutsk» виконує не лише технічну, але й комунікативну функцію. Воно формує стиль мовлення бренду, впливає на відчуття атмосфери додатку і допомагає передати характер міста.



Рис. 3.4. Типографіка MY LUTSK

Джерело: розроблено автором

Окрім основних графічних елементів, таких як логотип, кольорова палітра та шрифтове рішення, бренд-айдентика мобільного додатку «MyLutsk» включає цілісну систему візуальних принципів, які визначають характер і спосіб подачі інформації в усіх точках взаємодії з користувачем. Важливим завданням було досягти того, щоб додаток не лише виконував інформаційну та навігаційну функції, але й репрезентував образ міста - його атмосферу, ритм, унікальність та емоційне звучання.

Задля забезпечення візуальної впізнаваності у різних середовищах у брендбуку розроблено приклади застосування айдентики на маркетингових матеріалах. Сюди входять макети постерів, соціальних карток, інформаційних навігаційних банерів, сувенірної атрибутики та рекламних носіїв. Вони показують, як бренд живе поза межами інтерфейсу та взаємодіє з ширшим

інформаційним простором міста. Наявність таких матеріалів дозволяє будувати єдиний образ міста у свідомості туриста, підсилюючи зв'язок між цифровим та реальним середовищем.

Після визначення загальної візуальної стилістики та ключових елементів айдентики наступним етапом стало формування логіки навігації додатку, яка визначає, яким чином користувач взаємодіє з інтерфейсом. Схема навігації виконує роль «внутрішньої карти» додатку, де кожен екран і кожна дія мають своє місце та призначення. Вона допомагає забезпечити послідовність користувацького досвіду, роблячи взаємодію простою, інтуїтивною та передбачуваною.

Навігація побудована так, щоб користувач не відчував необхідності вчитуватися чи роздумувати над тим, що робити далі. Розміщення розділів не є випадковим. Воно відображає типові потреби людини, яка щойно приїхала в місто: спершу дізнатися про маршрути, знайти красиві місця для фото, ознайомитися з культурною спадщиною, а вже потім — подумати про сувеніри та подарунки. Така послідовність не лише логічна, але й психологічно комфортна, оскільки вона відповідає природному способу сприйняття нового простору[24].

Кожен розділ має однаковий спосіб побудови: **перегляд списку → детальна картка локації → можливість відкрити її на карті та продовжити шлях**. Завдяки цьому користувач не потребує «пере-навчання» щоразу, коли переходить до нового розділу. Додаток працює як надійний супутник, що супроводжує, але не нав'язується.

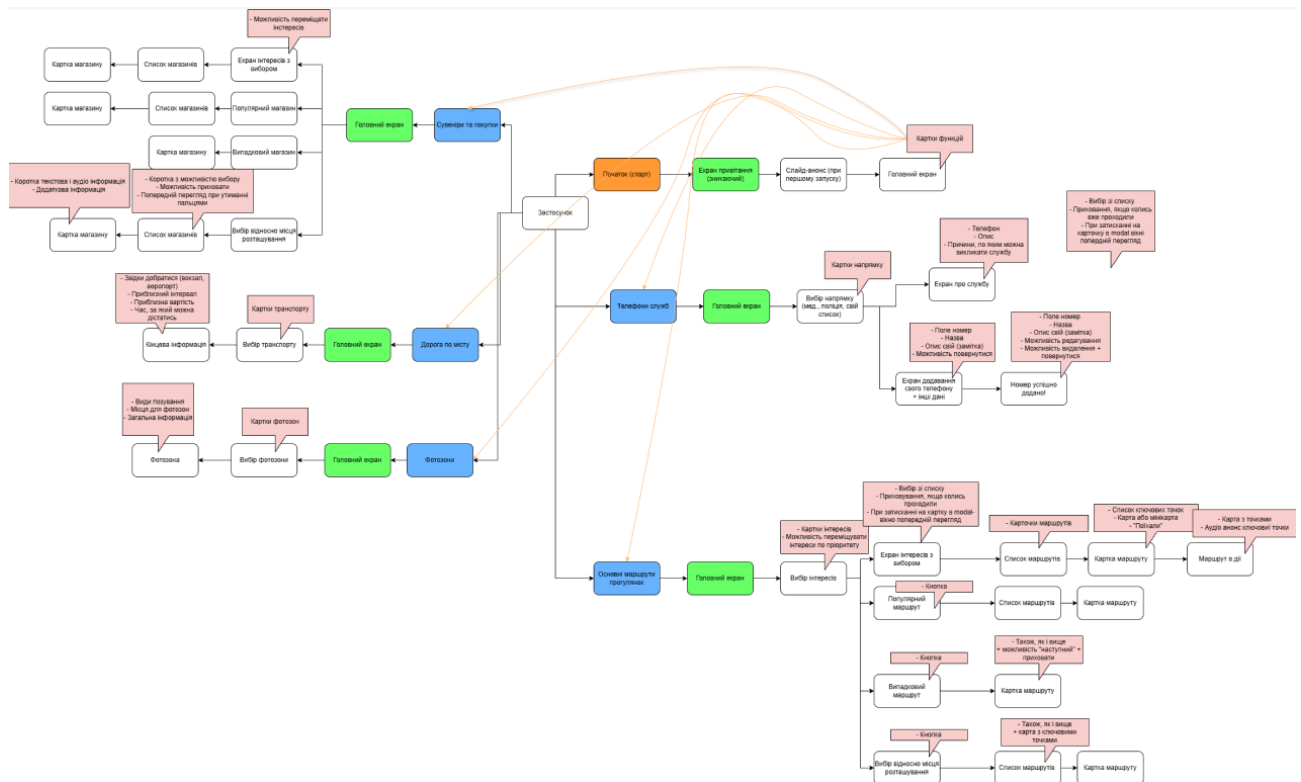


Рис. 3.4. Схема навігації мобільного додатку

Джерело: розроблено автором

На основі цієї навігаційної логіки спроектовано інтерфейс ключових екранів, які забезпечують безперервний сценарій користування: від першого знайомства зі змістом - до маршруту, фотографування, відвідання музеїв і придбання сувенірів.

Тому інтерфейс був побудований як **послідовна, логічно впорядкована система взаємопов'язаних екранів**, у якій кожен елемент має свою функцію і розташований так, щоб користувач міг знайти потрібне швидко та без напруги. Візуальні принципи - достатні відступи, повітря у композиції, контрастні заголовки, плавні акцентні кольори - безпосередньо продовжують стиль бренд-айдентики.

Посилання на мобільний додаток:

<https://www.figma.com/design/JGf7OELzOcYXK2YYX124fy/Untitled?node-id=0-1&t=EH3W8wiYs9YVdxX6-1>.

Головний екран виступає початковою точкою взаємодії користувача з додатком і задає загальний тон візуального досвіду. На цьому екрані відображено

логотип, що є ключовим елементом айдентики, та слоган «Луцьк - приємне відкриття!», виконаний акцентним шрифтом Caveat. Така комбінація створює емоційний вступ, який не лише інформує, а й формує настрій знайомства з містом.

Після стартового екрану користувач потрапляє на слайдер-гід, який знайомить його з історичним контекстом міста. Ілюстрації виконані у фірмовій кольоровій палітрі та стилістиці м'яких контурних форм, що підтримує загальний візуальний характер бренду.

Текстові блоки подані короткими, змістовними і легко читабельними - це важливо, оскільки користувач сприймає інформацію в русі та не має можливості читати великі фрагменти. Кнопка «Прокласти маршрут» одразу пропонує конкретну дію, що скорочує шлях взаємодії та стимулює інтерес до міста як до «живого маршруту», а не набору окремих точок.

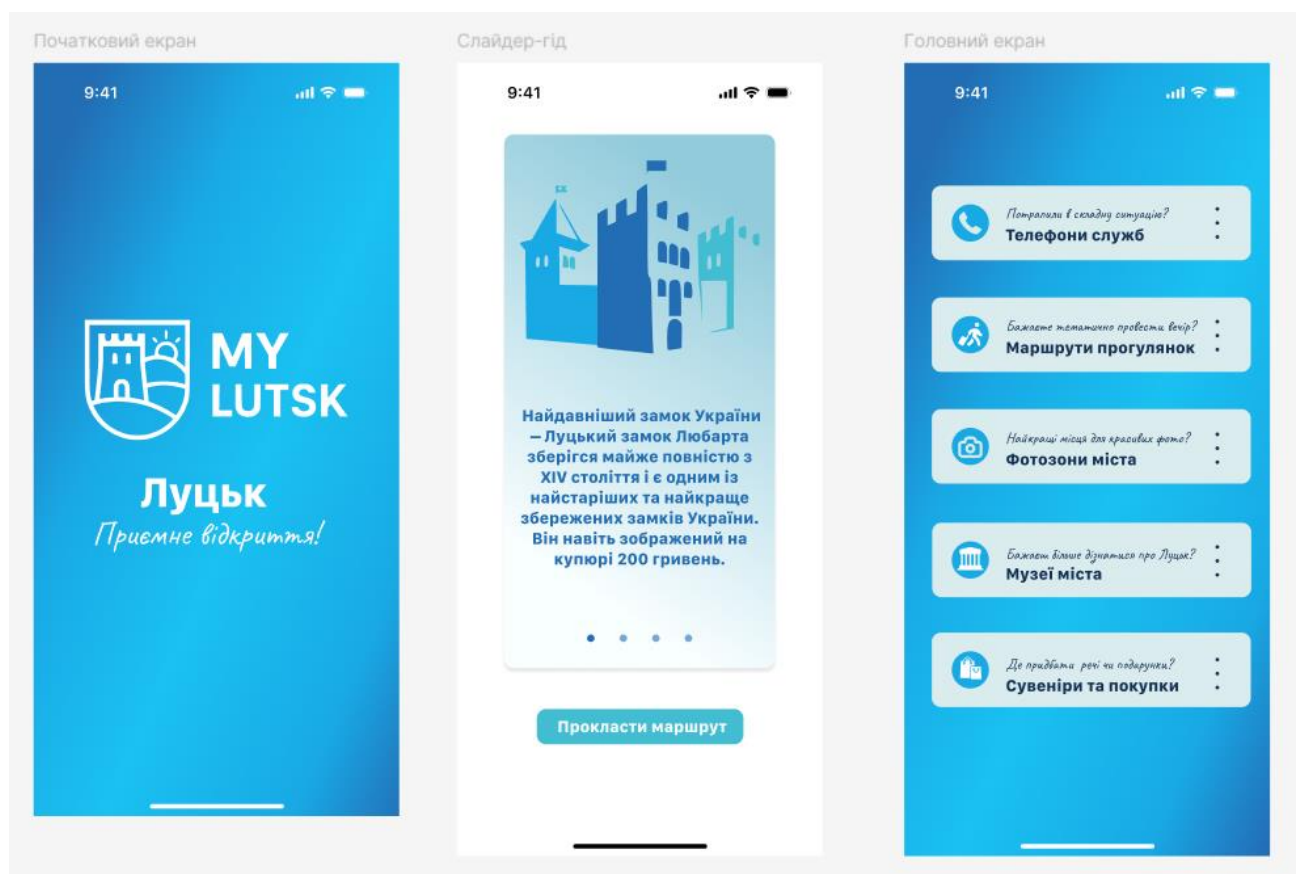


Рис. 3.5 Початковий та головний екрани мобільного додатку

Джерело: розроблено автором

Блок «Маршрути» - це центральний елемент додатку «MyLutsk», створений для того, щоб знайомство з містом було не випадковим, а цілісним і послідовним. Тут маршрути подані як готові прогулянкові шляхи, що поєднують ключові локації Луцька в єдину історію.

Каталог маршрутів оформлений у вигляді карток із великими зображеннями, що дозволяє користувачу обирати шлях, орієнтуючись насамперед на атмосферу і враження. Назва маршруту та позначка популярності допомагають швидко зрозуміти його стиль і складність.

Після відкриття маршруту інформація подається зрозуміло і послідовно: головне фото → тривалість → список зупинок → короткий опис → кнопки взаємодії. Така побудова дозволяє користуватися додатком прямо під час руху, не зупиняючись і не витрачаючи час на орієнтацію.

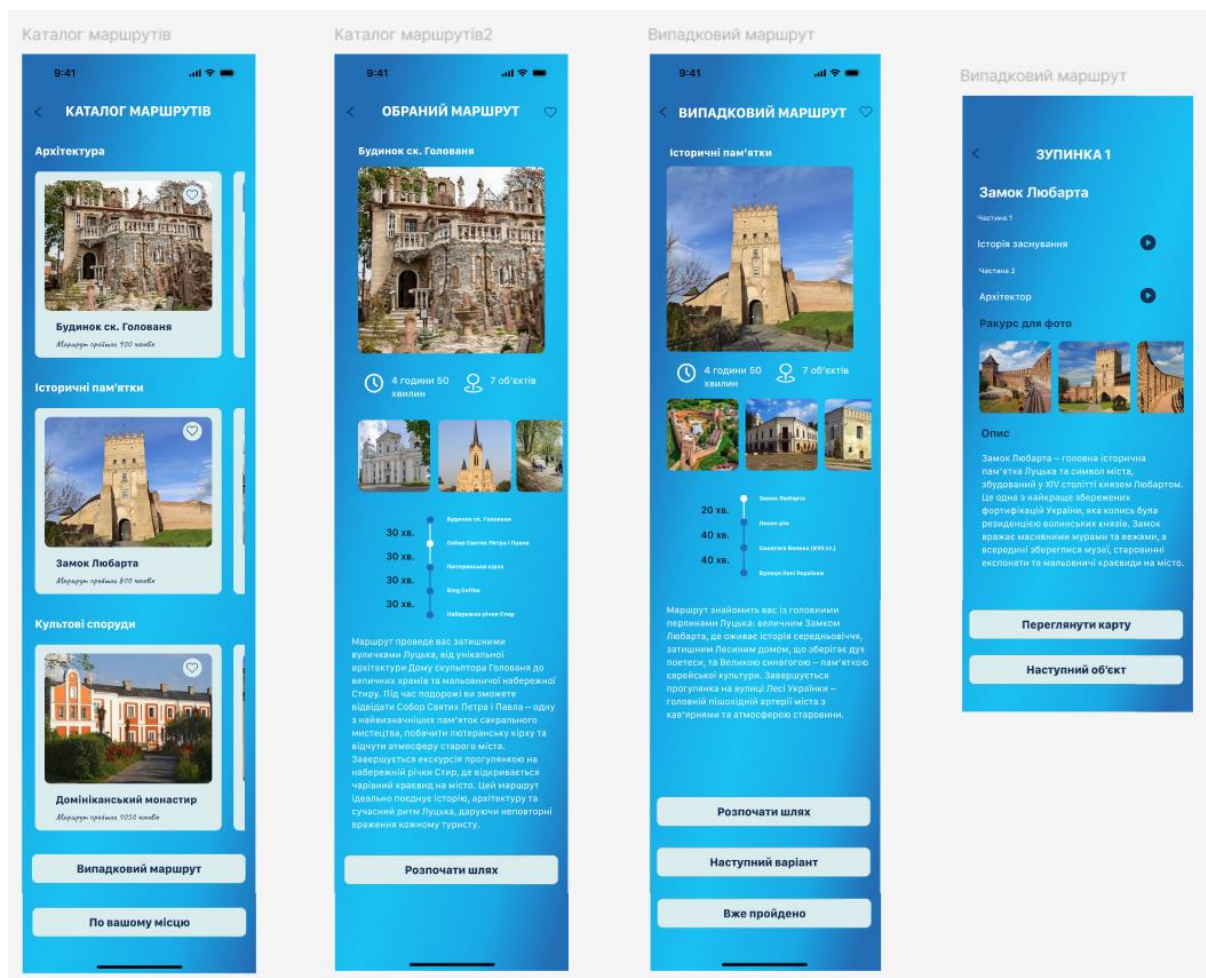


Рис. 3.6 – Каталог доступних маршрутів у додатку «MyLutsk».

Джерело: розроблено автором

Блок «Фотозони міста» допомагає користувачу побачити Луцьк як візуальний простір - місто світла, ракурсів та деталей, що розкривають його характер. Каталог фотозон організований у вигляді карток із великими зображеннями, що дає можливість обирати локацію емоційно, відчуючи її настрій ще до відвідування.

При переході до конкретної фотозони відкривається детальна сторінка з:

- основним фото,
- галереєю рекомендованих ракурсів,
- адресою і графіком роботи,
- порадами щодо часу зйомки та освітлення,
- можливістю перегляду локації на карті.

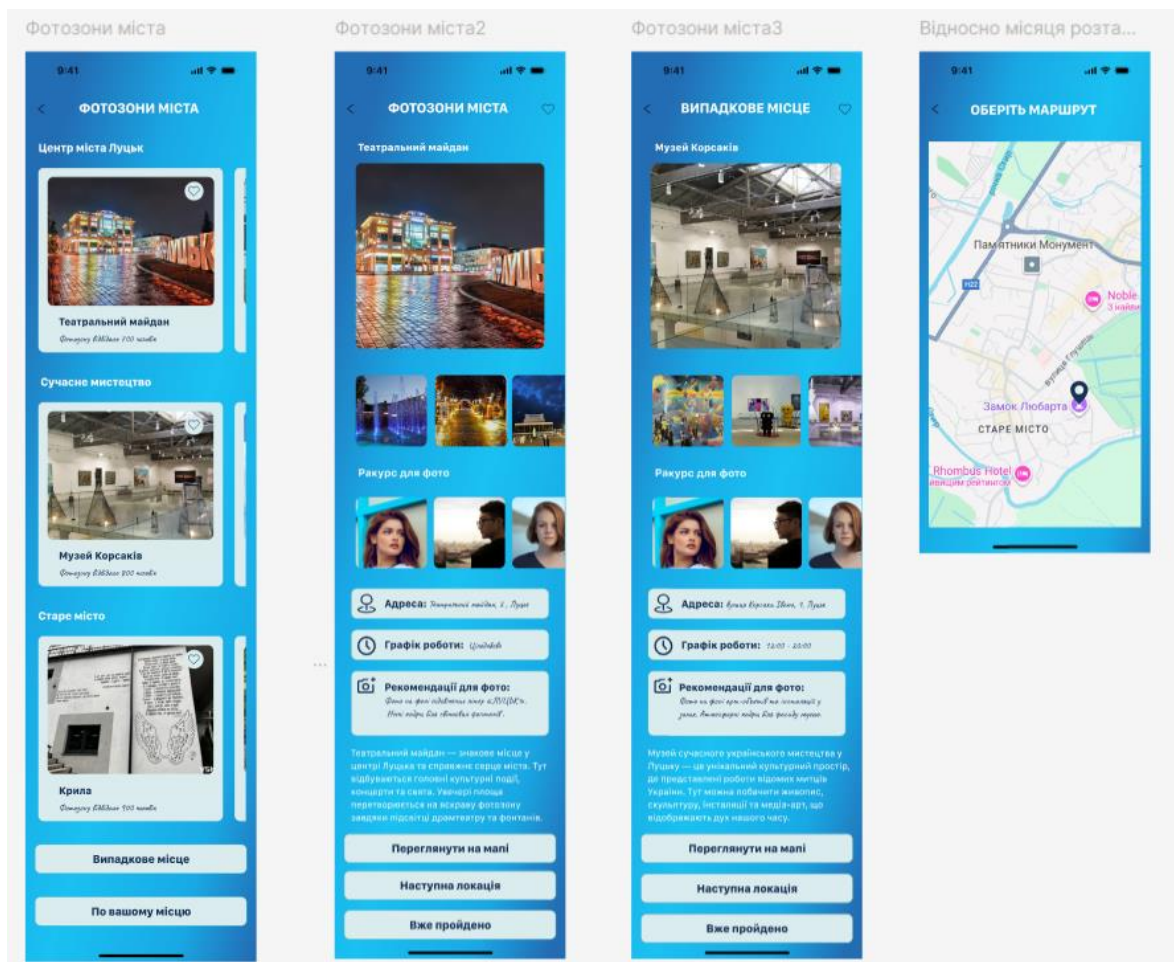


Рис. 3.7 - Каталог фотозон у додатку «MyLutsk».

Джерело: розроблено автором

Блок «Сувеніри та покупки» створений для того, щоб користувач міг знайти місця у Луцьку, де можна придбати локальні вироби та подарунки. Каталог сувенірів подано у вигляді карток із великими фотографіями. Такий підхід забезпечує швидке візуальне сприйняття та дозволяє користувачу обирати товар інтуїтивно - за стилем та виглядом, а не через довгі описи.

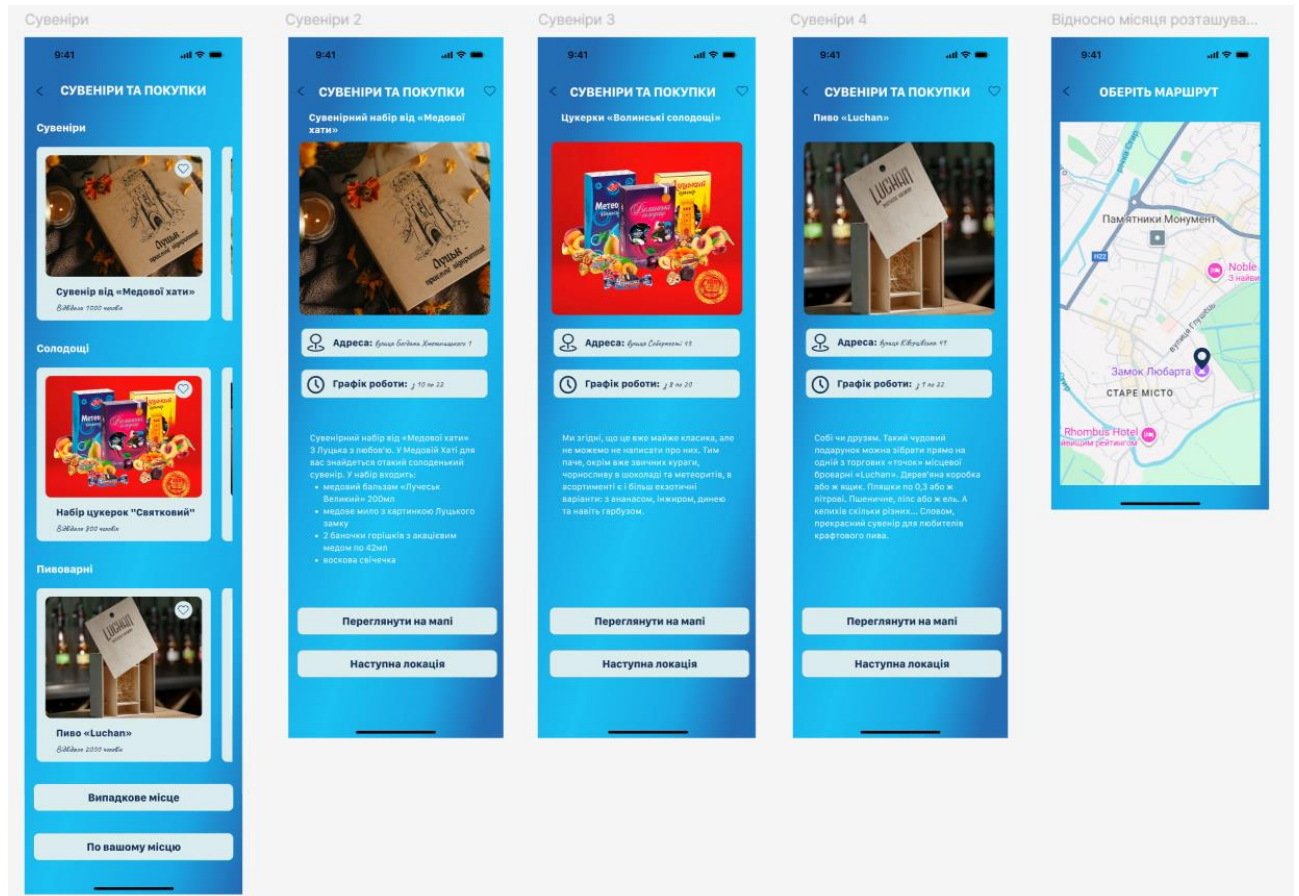


Рис. 3.8. Інтерфейс розділу «Сувеніри та покупки» в мобільному додатку «MyLutsk».

Джерело: розроблено автором

Блок «Каталог музеїв» призначений для ознайомлення користувача з основними культурними локаціями Луцька. Його завдання - надати зручний і швидкий доступ до інформації про музеї та допомогти спланувати відвідування.

Інформація подана стисло, без надлишкових текстів - так, щоб її можна було прочитати «на ходу», під час прогулянки.

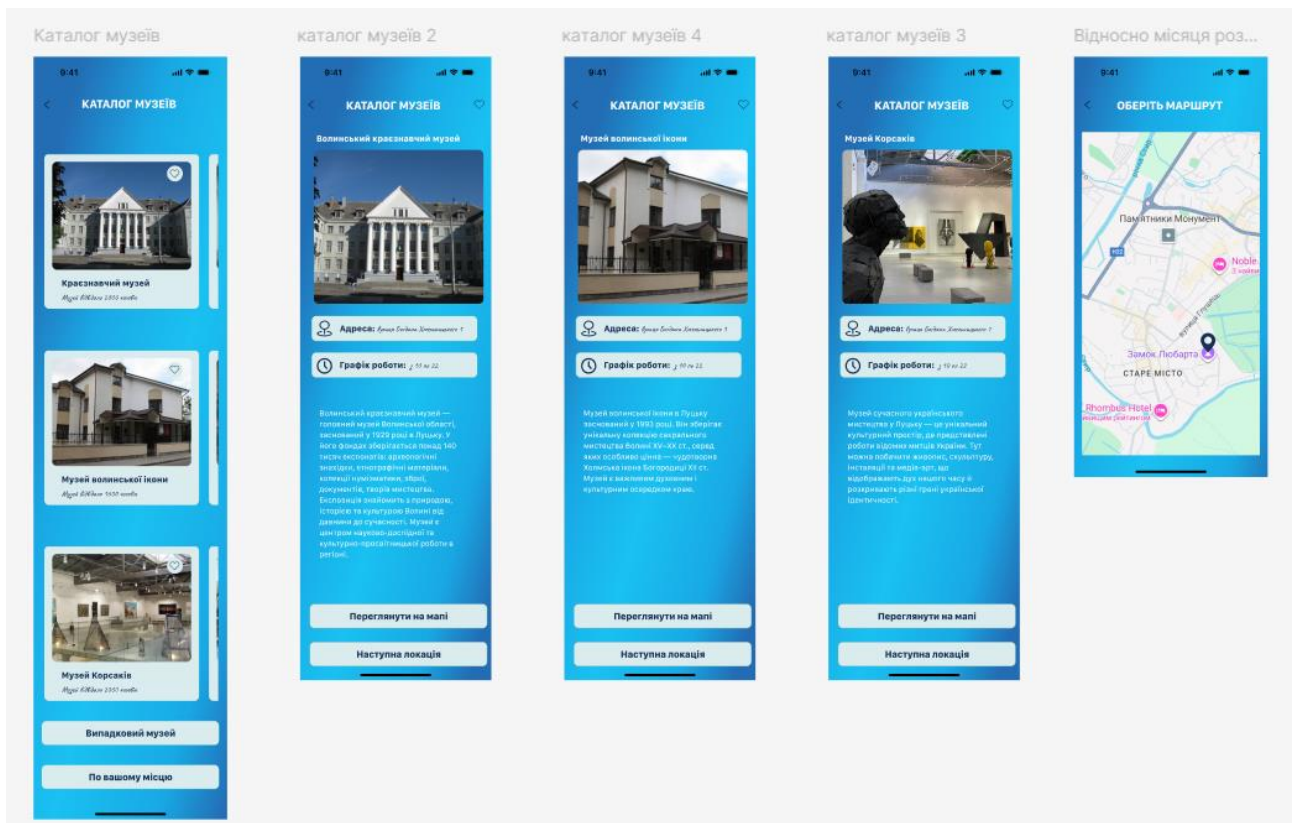


Рис. 3.9 - Каталог музеїв у додатку «MyLutsk».

Джерело: розроблено автором

MyLutsk» сформований як цілісна система, у якій візуальна айдентика, навігаційна логіка та дизайн інтерфейсів працюють узгоджено для однієї мети - перетворити знайомство з містом на послідовний, емоційно насичений і зручний досвід. Брендбук задає єдиний стиль мовлення додатку (колір, типографіка, іконографіка, фотостиль) і гарантує впізнаваність на всіх носіях - від екранів додатку до друкованих та рекламних матеріалів.

Таким чином, опис додатку підтверджує головну тезу проєкту: якісний туристичний застосунок - це синтез технічної простоти, візуальної послідовності та культурного змісту. Саме ця єдність робить «MyLutsk» зрозумілим з першого погляду, приємним у користуванні та корисним у реальних міських сценаріях, а отже - готовим до масштабування функціоналу та подальшого розвитку.

3.2. Accessibility testing

Проведення тестування доступності мало на меті визначити, наскільки інтерфейс мобільного додатку «MyLutsk» є комфортним, інтуїтивним і зрозумілим для різних категорій користувачів, у тому числі людей із порушеннями зору, моторики або просторового сприйняття, а також тих, хто використовує додаток у русі чи в умовах змішаного освітлення. Відповідно, оцінювалися читабельність, контрастність, послідовність навігації, зручність взаємодії та адаптивність інтерфейсу до різних сценаріїв використання.

У тестуванні застосовувалися рекомендації міжнародного стандарту WCAG 2.1, який визначає вимоги до доступності для людей із різними видами зорових і когнітивних особливостей. Додатково було враховано принципи Apple Human Interface Guidelines та Material Design Accessibility Guidelines, що регулюють оптимальні розміри інтерактивних елементів, ієрархію типографіки, логіку руху користувача між екранами та поведінку елементів при масштабуванні інтерфейсу. Це забезпечило не лише відповідність міжнародним нормам інклюзивності, але й практичну зручність використання додатку у реальних міських умовах - під час пересування, у транспорті або під відкритим небом.

Одним із ключових параметрів доступності інтерфейсу є забезпечення достатнього контрасту між текстом і фоновими площинами. Це особливо важливо для мобільних туристичних додатків, якими користуються в умовах змінного освітлення та руху. У додатку «MyLutsk» для побудови текстових та інформаційних блоків використано поєднання глибокого синього кольору (#153156) як фонові основи та білого кольору (#FFFFFF) для текстових елементів.

Контрастність цієї пари становить 7:1, що перевищує рекомендований мінімум 4.5:1, встановлений стандартом WCAG 2.1 для користувачів із частковим зниженням гостроти зору. Завдяки цьому текст залишається легко читабельним під час використання додатку:

- на відкритому повітрі при яскравому сонячному освітленні,
- у громадському транспорті,
- у вечірній час або при слабкому штучному світлі,
- під час руху, коли уважність користувача знижена.

Акцентні елементи інтерфейсу - кнопки, інтерактивні посилання, позначки - виконані у бірюзовому кольорі (#60ADB7). Його насиченість дозволяє впевнено виділяти інтерактивні області на темному тлі, проте він не створює зорової втоми, оскільки не є надмірно яскравим або агресивним для сприйняття.

Приклад перевірки контрастності в реальних умовах: Було проведено тест інтерфейсу на пристроях із різними рівнями яскравості. Навіть при зниженні яскравості екрану до 30% читабельність основного тексту та інтерактивних елементів залишилася стабільною. Це свідчить про адаптованість кольорової системи до варіативних умов використання, що є важливою складовою доступності.

Типографіка відіграє ключову роль у забезпеченні доступності, оскільки саме через текст користувач отримує основну інформацію. У додатку «MyLutsk» використовується шрифт SF Compact, який спроектований для високої читабельності на мобільних пристроях та зберігає чіткість при різних розмірах екрану і рівнях масштабування.

Щоб зменшити когнітивне навантаження, текстові елементи структуровані за ієрархією розмірів і ваги:

Таблиця 3.1

Шрифтові розміри та їх призначення

Рівень тексту	Розмір шрифту	Призначення
Заголовки	20–24 px	Назви екранів і ключових розділів
Підзаголовки	16–18 px	Опис підсекцій та елементи структури
Основний текст	14–16 px	Опис локацій, рекомендацій, інструкцій

Допоміжні підписи	12–14 рх	Коментарі, службова інформація
-------------------	----------	--------------------------------

Важливим є і характер подачі тексту: фрази короткі, змістові, побудовані за принципом *«одна думка - одне речення»*. Це дозволяє не «читати» текст, а сканувати його поглядом, що критично для туристичного використання в русі.

Для створення емоційних акцентів використано другий шрифт Caveat - рукописний, теплий і персоналізований. Його застосування обмежено заголовками-сентенціями чи фразами-привітаннями, що дозволяє додати інтонації «живого міста» без втрати читабельності. Таким чином, типографічна система допомагає сформувати збалансований ритм читання, при якому користувач отримує інформацію швидко, впевнено та без перевантаження.

Не менш важливою складовою доступності є порядок подачі інформації. У додатку «MyLutsk» всі екрани побудовані за принципом вертикальної логіки сприйняття, яка відповідає природному руху погляду:

1. Візуальний образ (фото або іконка локації) - створює перше емоційне враження.
2. Короткий текстовий опис (3–4 рядки ключової інформації) - дає змістовний орієнтир.
3. Дія / навігація (кнопки переходу або карта) - пропонує наступний крок.

Це виключає ситуації, коли користувачеві потрібно «шукати», куди натиснути чи як продовжити рух. Він завжди бачить наступну дію одразу, без необхідності запам'ятовувати структуру додатку.

Дублювання ключових навігаційних переходів («Перейти на мапу», «Наступна локація», «Повернутися») забезпечує сталість патернів, що особливо важливо для користувачів з труднощами концентрації.

Тестування доступності підтвердило, що дизайн «MyLutsk» є інклюзивним, передбачуваним і стійким до різних умов використання. Контрастність, зважена типографіка, комфортні розміри інтерактивних зон та

структурована подача інформації дозволяють зменшити когнітивне та фізичне навантаження на користувача. Це робить додаток не просто зручним, а таким, що адаптується під темп, контекст і особливості користувача.

Висновок до розділу 3

У процесі розробки дизайну мобільного додатку «MyLutsk» було сформовано комплексну візуальну систему, яка поєднує естетичну привабливість, емоційну виразність і функціональну зручність для користувача. Вибір візуальної стилістики ґрунтувався на прагненні передати унікальність міського середовища Луцька - його історичну глибину та атмосферу гостинності. Саме тому ключовими в дизайні стали м'які поєднання кольорів, мінімалістична іконографія, фотографічна автентичність та акцент на чистоті композиції.

Важливою складовою створення цілісного образу додатку став брендбук, який систематизував усі елементи айдентики: логотип, кольорову палітру, типографіку, стилістику фотографій, принципи розміщення елементів та графічні акценти. Брендбук забезпечив узгодженість візуального стилю у різних середовищах - від інтерфейсу додатку до рекламних матеріалів і сувенірної продукції. Завдяки цьому «MyLutsk» отримав чітку впізнавану ідентичність і став не лише цифровим продуктом, а частиною культурної репрезентації міста.

Проектування інтерфейсу здійснювалося із позиції сценаріїв реального використання, тому структура додатку є послідовною, передбачуваною та інтуїтивною. В основі навігації лежить логіка «від загального до конкретного» та повторюваність патернів взаємодії: список → детальний перегляд → карта. Це дозволяє користувачу не витратити час на освоєння інтерфейсу і зосередитися на самому досвіді знайомства з містом. Кожен екран виконує конкретну функцію і водночас підтримує емоційне сприйняття простору через фотографії, кольорові акценти і теплу типографіку.

Окрему увагу приділено доступності інтерфейсу. Дизайн було протестовано відповідно до рекомендацій WCAG 2.1, Apple Human Interface

Guidelines та Material Design Accessibility Guidelines. Забезпечено достатню контрастність, читабельність текстів у різних умовах освітлення, зручний розмір інтерактивних елементів та логічну ієрархію інформації. Це робить додаток доступним для користувачів із різним рівнем зорового сприйняття та моторної координації, а також комфортним у використанні під час активного пересування містом.

Отже, дизайн мобільного додатку «MyLutsk» виступає не лише декоративною складовою, а є повноцінним інструментом комунікації між містом і користувачем. Він формує емоційний настрій, підтримує змістовний досвід і забезпечує легкість навігації. Цілісна айдентика, продумані сценарії взаємодії та дотримання принципів доступності дозволили створити додаток, який є зручним, зрозумілим і виразним, а головне - здатним передати атмосферу Луцька як міста, що відкривається людині через враження, простір і живий діалог.

ВИСНОВКИ

У кваліфікаційній роботі було здійснено комплексне дослідження процесу проектування та дизайну туристичного мобільного додатку «MyLutsk» як інструменту підтримки туристичного розвитку міста Луцька. В ході роботи було визначено теоретичні засади створення цифрових продуктів у туристичній сфері, проаналізовано сучасні тенденції у сфері UI/UX дизайну та бренд-комунікації, а також розроблено практичне рішення, яке поєднує інформативність, доступність та емоційність користувацького досвіду.

Дослідження дозволило підтвердити, що сучасний туристичний сервіс потребує не лише надання інформації, але й формування атмосфери, враження та персоналізації взаємодії. Це зумовило вибір підходу, у якому користувач не просто отримує список об'єктів для відвідування, а переживає місто як емоційний простір, в якому маршрути, фотолокації, музеї та локальні сувеніри постають частиною особистої історії. Саме тому дизайн додатку був орієнтований на естетичну виразність, інтуїтивність і цілісність візуальної айдентики.

У результаті проведеного аналізу було визначено ключові елементи бренд-айдентики, що відповідають характеру міста Луцька: гармонійна кольорова палітра, сучасна та доступна типографіка, лаконічний та змістовний логотип, заснований на архітектурному символі міста - Замку Любарта. Створений брендбук забезпечив узгодженість ідентичності на всіх рівнях комунікації: від інтерфейсу додатку до маркетингових матеріалів та сувенірної продукції.

Проектування інтерфейсу мобільного додатку здійснювалося з урахуванням принципів зручності та передбачуваності користувацької взаємодії. Навігаційна структура була побудована за логікою простого поступального переходу між розділами, що забезпечує плавність та природність користувацького шляху. Додаток передбачає зручне використання «на ходу», що є важливим чинником для туристів, які переміщуються містом у реальному часі.

Практична цінність магістерської роботи полягає у створенні повноцінного прототипу мобільного додатку, який може бути впроваджений як у туристичній інфраструктурі міста Луцька, так і адаптований для інших міст України. Додаток сприяє розвитку туристичного бренду міста, підсилює його інформаційну доступність та формує позитивне емоційне сприйняття серед відвідувачів.

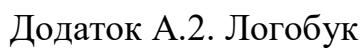
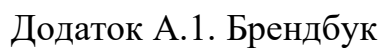
Отже, можна зробити висновок, що розроблений мобільний додаток «MyLutsk» виконує функцію сучасного та зручного цифрового провідника, який поєднує інформування, навігацію та емоційне залучення. Він не лише підтримує туристичну комунікацію, але й сприяє формуванню сталого інтересу до міста, поглибленого пізнання його культурного простору та створенню власної історії перебування в Луцьку. Робота має як теоретичне, так і практичне значення, відкриваючи перспективи подальшого розвитку, масштабування та вдосконалення проєкту в майбутньому.

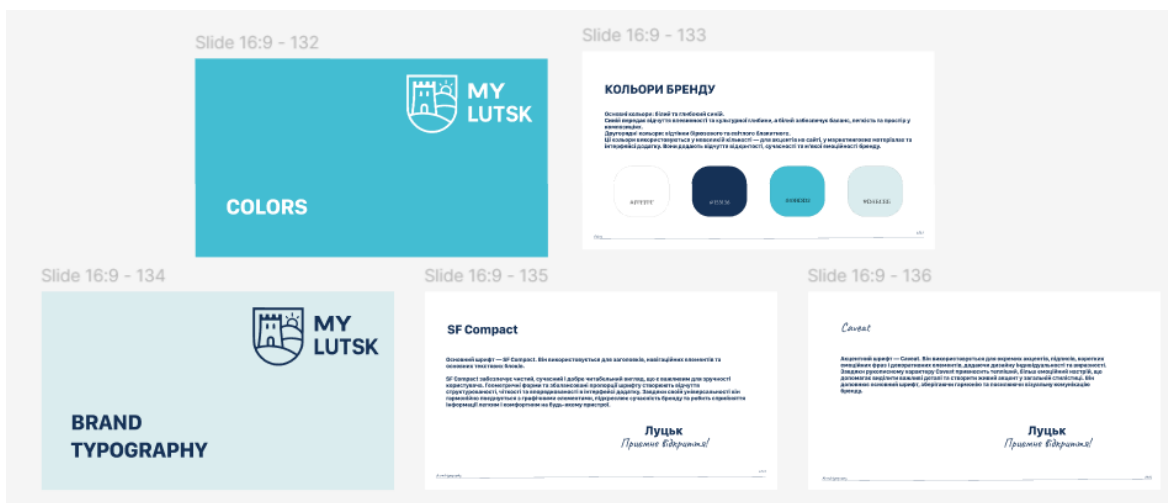
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шиколенков, Т. А. Розробка і робота мобільних додатків: навч. посіб. Київ: Видавничий центр КНУКіМ, 2014. 379 с.
2. Воробйов, О. О. Розробка сучасних веб-додатків: основи та приклади. Дніпро: Журфонд, 2019. 157-158 с.
3. Акіла, М. Т. Початковий Node.js і Express.js: навч. посіб. Київ: Видавничий центр КНУКіМ, 2016. 334 с.
4. Бабарицька, В. К., Малиновська, О. Ю. Менеджмент туризму. Туроперейтинг. Понятійно-термінологічні основи, сервісне забезпечення турпродукту: навч. посіб. Київ: Альтерпрес, 2014. 288 с.
5. Бабарицька, В. К., Любіцева, О. О. Організація туризму : навч. посіб. Ч. 1. Київ, 1998. 82 с.
6. Двуліт, З. П., Наумчук, С. І. Контент-маркетинг як інструмент успішного функціонування вітчизняних компаній. Економіка та держава, 2018, № 10, с. 10–13.
7. Дрокіна, Н. І. SEO-оптимізація сайту підприємства як інструмент інтернет-маркетингу. Науковий вісник Ужгородського національного університету. Серія: Міжнародні економічні відносини та світове господарство, 2018, вип. 19, Ч. 1, с. 127–132.
8. Емброуз, Г., Леонард, Н. Основи. Графічний дизайн 02. Дизайнерське дослідження. Пошук успішних креативних рішень. Київ: ArtHuss, 2019. 192 с.
9. Етапи створення сайту з нуля. My Master. URL: <https://mymaster.net.ua/ua/etapi-stvorenniya-sajtu-z-nulya/> (дата звернення: 27.02.2025).
10. Інструмент маркетингової та рекламної стратегії. siteGist. URL: <https://www.sitegist.com/en/services/web/landing> (дата звернення: 20.02.2025).

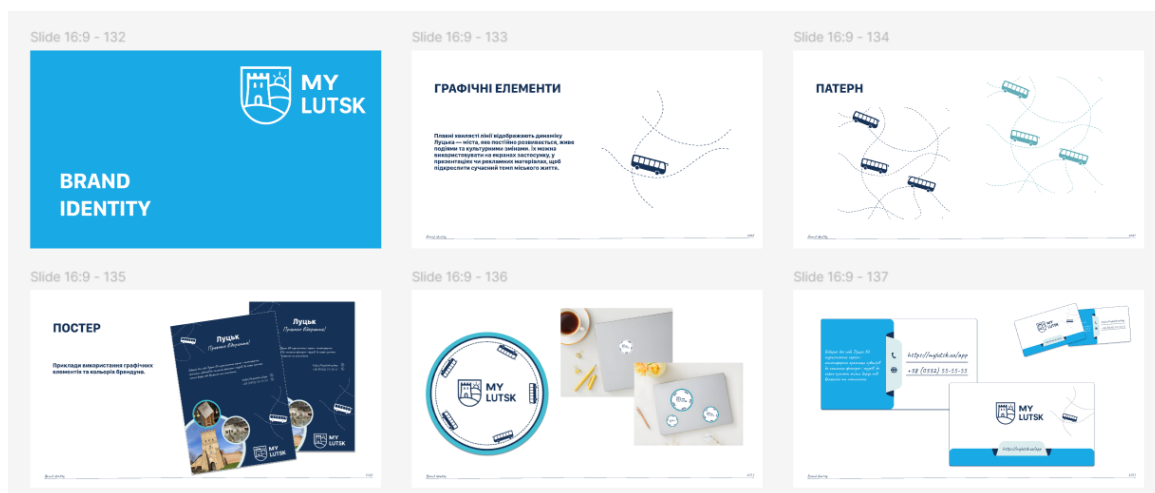
11. Ледбрук, Л. Управління проектом водоспадної моделі: огляд. 13 липня, 2012. URL: <http://projectcommunityonline.com/waterfall-project-management-an-overview.html> (дата звернення: 27.02.2025).
12. Маніфест Agile. Agilealliance. URL: <https://www.agilealliance.org/agile101/the-agilemanifesto/> (дата звернення: 27.02.2025).
13. Матеріали X науково-технічної конфіції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 7–8 грудня 2022 р.). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2022. 30 с.
14. Актуальні задачі сучасних технологій : зб. тез доповідей XI міжнар. наук.-практ. конф. Молодих учених та студентів, (Тернопіль, 7-8 грудня 2022) / М-во освіти і науки України, Тернопільський національний технічний університет імені Івана Пулюя [та ін.]. Тернопіль: ФОП Паляниця В. А., 2022. 169 с.
15. Матеріальна підтримка культурних проектів: Світовий досвід та можливості застосування в Україні. Колективна монографія. Інститут культурології академії мистецтв України. Київ, 2019.
16. Мардан, А. К. React швидко. Мобільні додатки на React Native, JSX, Redux: навч. посіб. Львів: OPілі, 2019. 560 с.
17. Мельник, А. М. Архітектура сучасних веб-додатків. Київ: Видавничий дім «Києво-Могилянська академія», 2021. 55 с.
18. Пасічник, О. Г., Пасічник, О. В., Стеценко, І. В. Основи веб-дизайну: навч. посіб. Київ: Вид. група BNV, 2009. 336 с.
19. Повна, С. В. Світовий досвід запровадження гнучких методологій управління конкурентоспроможністю підприємств та організацій. Проблеми і перспективи економіки та управління, № 1(21), 2020. URL: <http://ppeu.stu.cn.ua/article/view/211371> (дата звернення: 22.02.2025).

20. Рантюк, І. І. Огляд гнучких методологій в управлінні ІТ проектами. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2021/01/141.pdf> (дата звернення: 22.02.2025).
21. Чемерис, Г. Ю., Осадча, К. П. Проектування користувацького інтерфейсу: навч. посіб. для викладачів та студентів закладів вищої освіти. Мелітополь: ФОП Однорог Т., 2019. 300 с.
22. Київський Університет Культури. *Київський Університет Культури*. URL: <https://kuk.edu.ua/> (дата звернення: 29.09.2025).
23. Camren Browne. What are user flows in UX design?. *CareerFoundry*. URL: <https://careerfoundry.com/en/blog/ux-design/what-are-user-flows/>. (дата звернення: 05.06.2025).
24. Information architecture: a UX designer's guide - Justinmind. *Justinmind*. URL: <https://www.justinmind.com/wireframe/information-architecture-ux-guide> (дата звернення: 10.06.2025).
25. Website structure: how to plan the perfect site layout | figma. *Figma*. URL: <https://www.figma.com/resource-library/website-structure/> (дата звернення: 10.06.2025).
26. How to create an engaging university website design in 2025. *Seahawk*. URL: <https://seahawkmedia.com/design/university-website-design/> (дата звернення: 11.06.2025).
27. Ein kompletter Leitfaden für die Erstellung einer Website | Mailchimp. *Mailchimp*. URL: <https://mailchimp.com/resources/how-to-make-a-website/> (дата звернення: 11.06.2025).
28. Mitrovikj M. UX/UI trends 2025. *Medium*. URL: <https://medium.com/codeart-mk/ux-ui-trends-2025-818ea752c9f7> (дата звернення: 12.06.2025).
29. The biggest UX design trends shaping the industry in 2025. *UX Design Institute*. URL: <https://www.uxdesigninstitute.com/blog/ux-design-trends-in-2025/> (дата звернення: 14.06.2025).

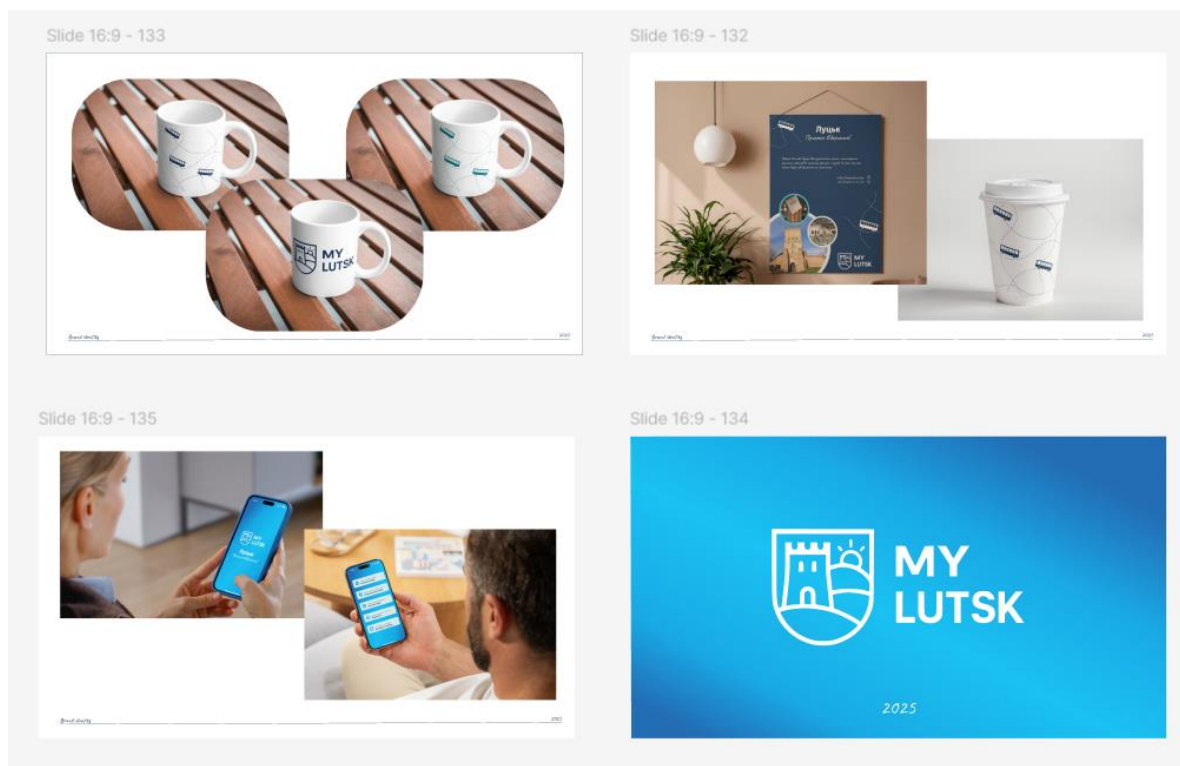




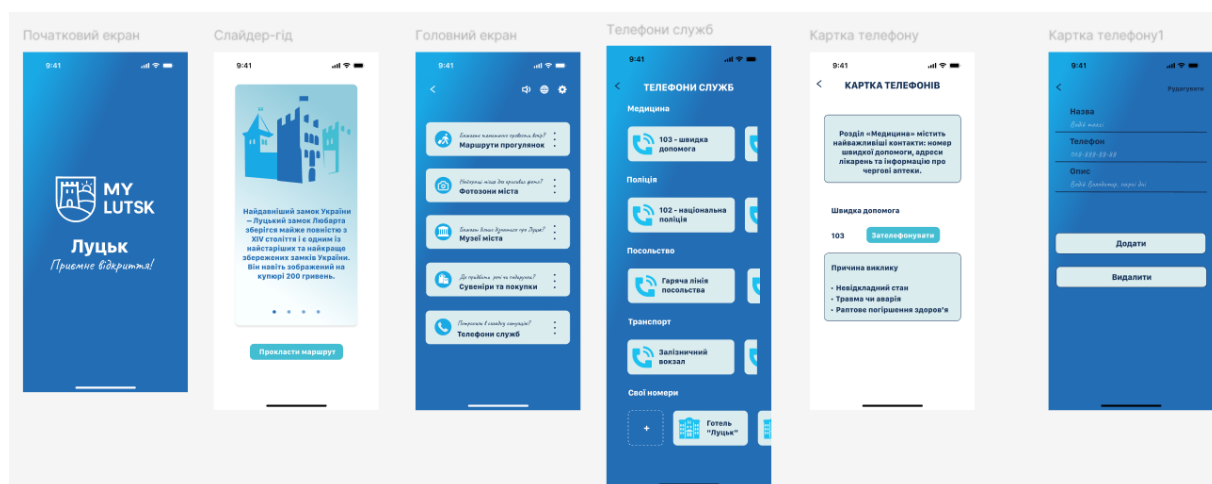
Додаток А.3. Брендбук



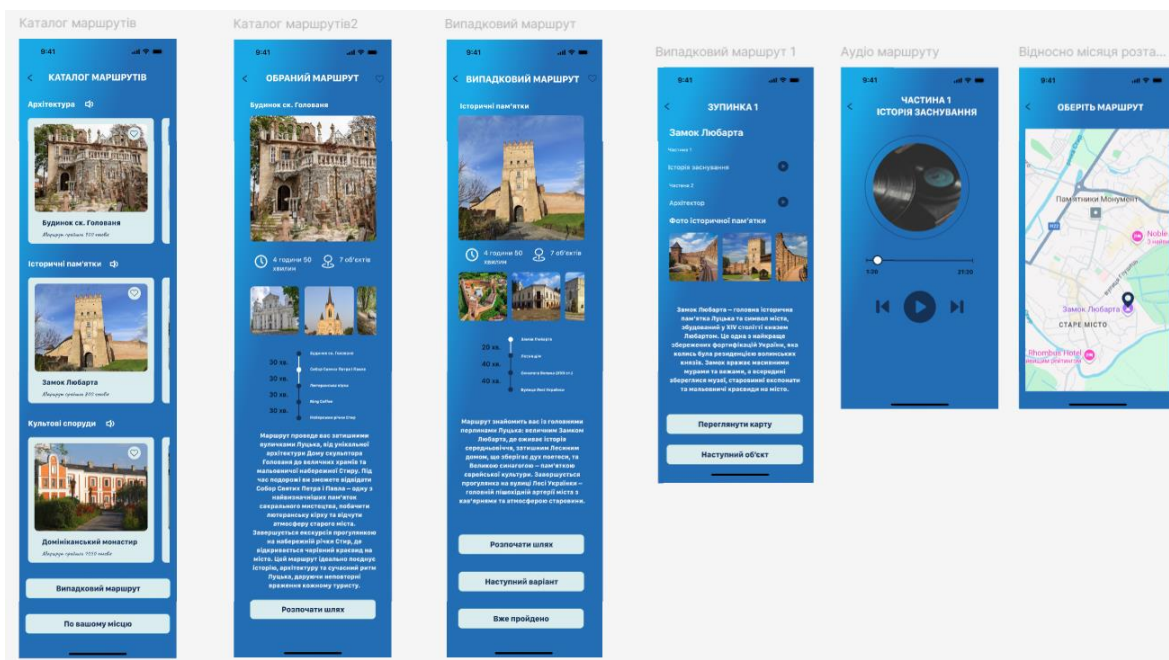
Додаток А.4. Брендбук



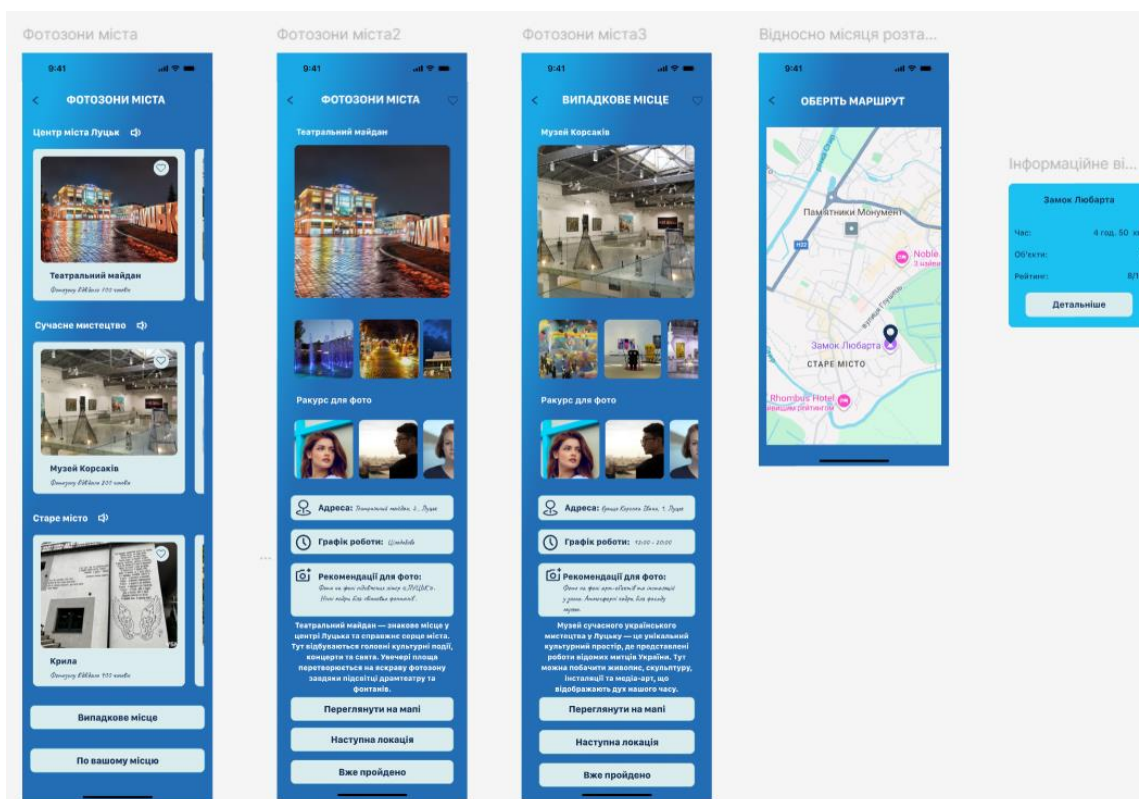
Додаток А.5. Бренбук



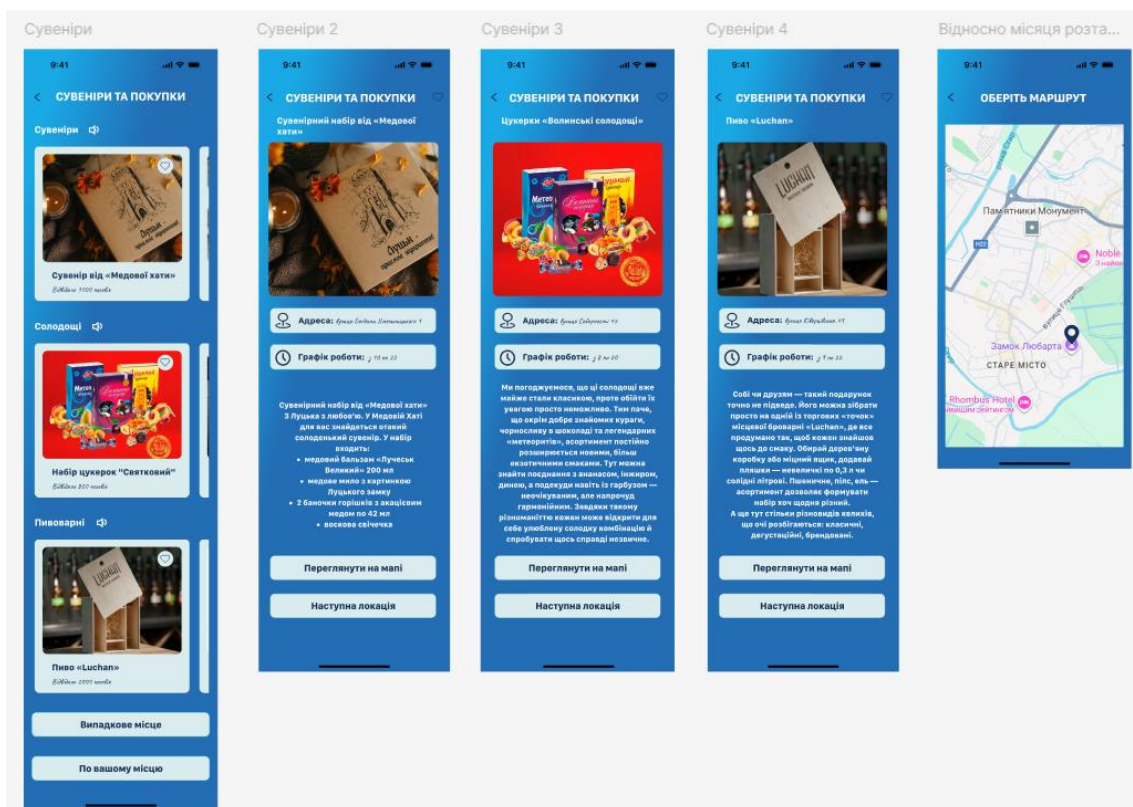
Додаток Б.1. Застосунок «MY LUTSK»



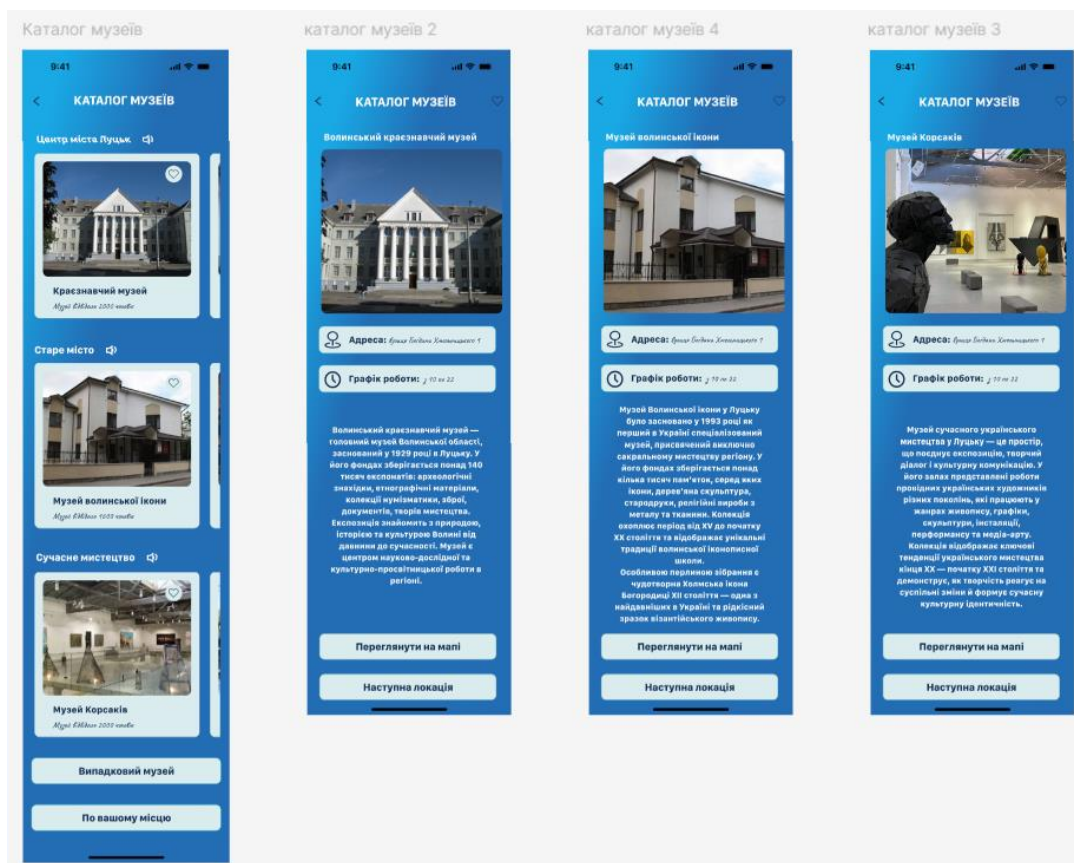
Додаток Б.2. Каталог маршрутів «MY LUTSK»



Додаток Б.3. Фотозони «MY LUTSK»



Додаток Б.4. Сувеніри «MY LUTSK»



Додаток Б.5. Музеї «MY LUTSK»