МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКА «ПІДБІР ТА ЗАМОВЛЕННЯ КВІТІВ»»**

Виконав: студент 4 курсу, групи 6.1261
спеціальності 126 Інформаційні системи та технології
(шифр і назва спеціальності)
освітньої програми Інформаційні системи та технології
(назва освітньої програми)
К.Г. Леоновець
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, PhD, Калюжняк А. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент декан математичного факультету, професор, д.т.н. С.І. Гоменюк
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології
(шифр і назва)

Освітня програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г. М..
(підпис)

« ____ » _____ 2024 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Леоновцю Кирилу Геннадійовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка мобільного застосунку «Підбір та замовлення квітів»

керівник роботи Калюжняк Анастасія Вікторівна, PhD
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 20 » грудня 2024 року № 2001-с

2. Строк подання студентом роботи 26.05.25

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Технологіні основи мобільних інструментів та електронної комерції.
2. Проєктування мобільного застосунку.
3. Програмна реалізація та тестування.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 23.12.2024

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	02.01.2025	
2.	Збір вихідних даних.	19.01.2025	
3.	Обробка методичних та теоретичних	11.02.2025	
	джерел.		
4.	Розробка першого та другого розділу.	28.02.2025	
5.	Розробка третього розділу.	12.04.2025	
6.	Оформлення та нормоконтроль	26.05.2025	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	20.06.2025	

Студент _____ К.Г. Леоновець
(підпис) (ініціали та прізвище)

Керівник роботи _____ А. В. Калюжняк
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка мобільного застосунку «Підбір та замовлення квітів»»: 46 с., 11 рис., 11 джерел.

ANDROID, KOTLIN, MATERIAL DESIGN, UI/UX, ЗАМОВЛЕННЯ КВІТІВ, КВІТИ, МОБІЛЬНИЙ ЗАСТОСУНОК, ПІДБІР БУКЕТІВ, ПРОГРАМУВАННЯ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ФІЛЬТРАЦІЯ ДАНИХ.

Об’єкт дослідження – процес проектування та розробки мобільних застосунків для платформи Android у сфері електронної комерції.

Предмет дослідження – моделі, методи та засоби розробки мобільного застосунку на платформі Android з використанням мови програмування Kotlin.

Мета роботи – розробити мобільний застосунок для платформи Android, який надає користувачам інструмент для пошуку та вибору квіткових букетів за допомогою системи фільтрів.

Методи дослідження – об'єктно-орієнтоване проектування, ітеративна розробка з використанням мови Kotlin та середовища Android Studio, моделювання користувацьких сценаріїв, тестування функціональності.

У роботі обґрунтовано вибір нативної розробки для платформи Android з використанням мови Kotlin. Спроектовано архітектуру застосунку на основі компонентів Activity та деталізовано користувацькі інтерфейси для основних екранів. Програмно реалізовано ключовий функціонал: модуль фільтрації за типами квітів із двома режимами (звичайний та строгий), динамічне відображення списку результатів, перегляд зображень та зміну теми оформлення. Продемонстровано роботу функціонального прототипу, що вирішує проблему вибору букета для користувача.

Отримані результати підтверджують практичну придатність запропонованого рішення для нішевих проектів у сфері мобільної комерції та створюють основу для подальшого розширення функціоналу системи.

SUMMARY

Bachelor's Qualifying Theses «Development of a Mobile Application Selection and Ordering of Flowers»: 46 pages, 11 figures, 11 references.

ANDROID, KOTLIN, MATERIAL DESIGN, UI/UX, FLOWER ORDERING, FLOWERS, MOBILE APPLICATION, BOUQUET SELECTION, PROGRAMMING, SOFTWARE DEVELOPMENT, DATA FILTERING.

Object of research – the process of designing and developing mobile applications for the Android platform in the e-commerce domain.

Subject of research – models, methods, and tools for developing a mobile application on the Android platform using the Kotlin programming language.

Aim of the work – to develop a mobile application for the Android platform that provides users with a tool for searching and selecting flower bouquets using a filter system.

Research methods – object-oriented design, iterative development using the Kotlin programming language and Android Studio environment, modeling user scenarios, functionality testing.

The thesis substantiates the choice of native development for the Android platform using the Kotlin language. The application's architecture was designed based on Activity components, and the user interfaces for the main screens were detailed. Key functionality was implemented programmatically: a filtering module by flower type with two modes (normal and strict), dynamic display of the results list, image viewing, and theme changing. A functional prototype that solves the user's problem of bouquet selection was demonstrated.

The obtained results confirm the practical applicability of the proposed solution for niche projects in the mobile commerce sphere and create a basis for the further expansion of the system's functionality.

ЗМІСТ

Завдання на кваліфікаційну роботу 2
Реферат 4
Summary 5
Вступ 7
1 Технологіні основи мобільних інструментів та електронної комерції 9
1.1 Огляд мобільних технологій 9
1.2 Електронна комерція та мобільна комерція 13
1.2.1 Особливості ніші продажу квіткової продукції через мобільні застосунки 16
1.2.2 Аналіз конкурентного середовища 16
1.3 Обґрунтування розробки та визначення ринкової ніші 20
2 Проєктування мобільного застосунку 22
2.1 Використані методи та засоби розробки 22
2.2 Опис функціонування та сценарії використання застосунку «BouqetApp» 23
3 Програмна реалізація та тестування 28
3.1 Реалізація ключових модулів та функціоналу 28
3.1.1 Модуль фільтрації та вибору режиму 28
3.1.2 Модуль відображення списку букетів та логіка фільтрації 30
3.1.3 Механізм перемикання тем 32
3.2 Особливості розробки користувацького інтерфейсу 34
3.3 Тестування та відлагодження 36
3.4 Перспективи розвитку проекту 39
Висновки 44
Перелік посилань 46

ВСТУП

У сучасному світі, де смартфон є частиною нашого життя, відбуваються зміни у процесах купівлі та продажу. Мобільні додатки стають зручним інструментом для зручного вибору та економії часу, що особливо відчутно у щоденному житті. Флористика, як сфера, де дарування квітів асоціюється з емоціями, увагою та турботою, не є винятком. Але, як показує практика, багато людей відчувають труднощі при виборі квітів, не знаючи смаків адресатки або адресата, що може зіпсувати ефект від подарунка

Актуальність теми «Розробка мобільного додатку для підбору та купівлі квітів» виникає з потреби усунення цієї невідповідності між бажаним та реальним результатом. Сучасний ринок мобільних застосунків активно впроваджує персоналізовані рішення адаптовані до індивідуальних потреб. Запропонований мобільний додаток допомагає вирішити проблему складного вибору, пропонуючи набір фільтрів звуження запропонованих варіантів. Цей підхід враховує як естетичні, так і психологічні аспекти вибору, що допомагає створити ідеальний букет чи композицію, які відповідають особистості одержувача.

Дослідження та розробка мобільного застосунка передбачає вирішення складного завдання, що включає аналіз сучасних технологій в UX/UI дизайні, інтеграцію з платіжними та логістичними системами, та забезпечення зручного і адаптивного інтерфейсу. Особлива увага приділяється розробці алгоритмів підбору квітів, які базуються на заданих фільтрах, що дозволить не тільки прискорити вибір, але й зробити його інтуїтивно зрозумілим та приємним.

Для реалізації поставленої мети, необхідно розв’язати наступні завдання:

- провести аналіз літератури;
- провести порівняльну характеристику подібних мобільних додатків;
- реалізувати застосунок у Android Studio;
- провести тестування застосунку;

У роботі буде розглянуто теоретичну основу мобільних додатків як інструменту електронної комерції, проаналізовано потреби користувачів у персоналізованому підборі товарів, а також проведено детальний огляд сучасних технологій та методології розробки. Значна увага приділяється практичній частині дослідження, що включає розробку прототипу додатку, його тестування та оцінку ефективності запропонованого рішення.

Отже, дана робота має на меті вирішення актуальної проблеми вибору квітів через створення мобільного додатку, який дозволить користувачам швидко та ефективно знаходити відповідні варіанти, враховуючи індивідуальні вподобання. Впровадження цього рішення має на меті підвищити якість обслуговування у сфері флористики, а також розширити можливості електронної комерції шляхом інтеграції інноваційних технологій у повсякденне життя.

1 ТЕХНОЛОГІЧНІ ОСНОВИ МОБІЛЬНИХ ІНСТРУМЕНТІВ ТА ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1 Огляд мобільних технологій

Мобільні технології стрімко розвиваються і стають однією з провідних рушійних сил сучасного інформаційного суспільства. Згідно з даними дослідницької компанії Statista, у 2022 р. кількість активних користувачів смартфонів у світі перевищила 6,5 млрд, що свідчить про високу проникність мобільних пристроїв у повсякденне життя населення[1]. В Україні також спостерігається динамічне зростання використання мобільних технологій, зокрема завдяки розширенню покриття 4G та запровадженню 5G у найближчій перспективі[2].

До основних технологічних компонентів мобільних рішень належать:

- операційні системи – найпоширеніші платформи для розробки мобільних застосунків. Android від Google має відкритий вихідний код та найбільшу частку ринку, тоді як iOS від Apple вирізняється більшою контрольованістю екосистеми та високою платоспроможністю користувачів[3];
- Android розроблена компанією Google, ця ОС є лідером за кількістю пристроїв у світі. Її відкритість (на базі Android Open Source Project - AOSP) дозволяє виробникам адаптувати систему під свої пристрої, що призводить до великої фрагментації ринку – існує безліч пристроїв з різними розмірами екранів, апаратними характеристиками та версіями ОС. Це створює певні виклики для розробників, які змушені забезпечувати коректну роботу застосунку на широкому спектрі пристроїв. Водночас, відкритість та гнучкість платформи надають розробникам більше свободи у реалізації функціоналу. Розповсюдження застосунків відбувається переважно через магазин

Google Play, який має менш суворі правила публікації порівняно з App Store;

- iOS операційна система компанії Apple, що використовується виключно на її власних пристроях (iPhone, iPad). Це створює закриту, але висококонтрольовану екосистему. Фрагментація тут мінімальна, що значно спрощує процес розробки та тестування. Користувачі iOS, як правило, більш схильні до здійснення покупок у застосунках, що робить цю платформу привабливою для бізнесу. Однак, процес публікації застосунків у App Store є більш тривалим та вимогливим;
- інструменти розробки (Android Studio, Xcode, React Native, Flutter тощо) – забезпечують середовище для створення мобільних застосунків різного рівня складності;
- нативна розробка. Цей підхід передбачає використання офіційних інструментів та мов програмування для кожної платформи окремо: Kotlin, Java та Android Studio для Android, Swift, Objective-C та Xcode для iOS. Головною перевагою нативної розробки є максимальна продуктивність, повний доступ до всіх апаратних та програмних можливостей пристрою, а також відповідність інтерфейсу застосунку гайдлайнам конкретної платформи. Саме цей підхід було обрано для розробки «BouqetApp», щоб забезпечити найкращий користувацький досвід та продуктивність на цільовій платформі;
- кросплатформна розробка: Такі фреймворки, як React Native, Flutter, Xamarin, дозволяють писати код один раз і запускати його на обох платформах (Android та iOS). Це може значно зменшити час та вартість розробки. Однак, такий підхід часто має певні обмеження: продуктивність може бути нижчою порівняно з нативними застосунками, доступ до деяких специфічних функцій платформи може бути ускладненим, а інтерфейс може виглядати менш «рідним» для користувача. Цей варіант є доцільним для проектів, де швидкість

виходу на ринок є ключовою, а вимоги до продуктивності та специфічного функціоналу не є критичними;

- технології хмарних сервісів (Firebase, AWS, Azure) – забезпечують надійне зберігання даних, гнучке масштабування та доступ до ресурсів у реальному часі. Це особливо важливо для застосунків, які мають обробляти велику кількість транзакцій чи запитів одночасно[5]. Для застосунку «BouqetApp» можна було б використати хмарні сервіси на наступних етапах розвитку;
- Firebase Authentication для реалізації системи реєстрації та входу користувачів;
- Firebase Firestore або Realtime Database: для зберігання та синхронізації каталогу букетів, замовлень користувачів та списків «Обране» в реальному часі;
- Firebase Cloud Storage: для зберігання зображень букетів у високій якості;
- Firebase Cloud Functions: для виконання серверної логіки, наприклад, обробки платежів або відправки сповіщень.

Розвиток мобільних технологій стимулює появу нових моделей бізнесу, пов’язаних із доставкою товарів, послуг, а також підтримує розвиток електронної комерції, про що йтиметься далі.

Варто зазначити, що саме мобільні застосунки стали ключовим каналом взаємодії бізнесу зі споживачем, пропонуючи персоналізований досвід та миттєвий доступ до продуктів. Це особливо помітно у сферах, де швидкість та зручність є критичними, наприклад, у замовленні їжі, таксі, або у квіткової продукції. Можливість зробити замовлення «на ходу», відстежити його статус та отримати підтримку в режимі реального часу значно підвищує лояльність клієнтів.

Окрім перерахованих компонентів, не можна оминути увагою й інші важливі аспекти сучасних мобільних технологій. Наприклад, значну роль відіграють технології бездротового зв’язку, такі як Wi-Fi, Bluetooth та NFC, які

забезпечують не тільки доступ до Інтернету, але й взаємодію між пристроями та з різноманітним обладнанням, наприклад, для безконтактних платежів або швидкого обміну даними. Також все більшого значення набувають технології геолокації, які використовуються у навігаційних сервісах, логістиці, маркетингу на основі місцезнаходження та багатьох інших сферах.

Не менш важливим є питання дизайну користувацького інтерфейсу та досвіду користувача. Успіх мобільного застосунку значною мірою залежить від того, наскільки він зручний, інтуїтивно зрозумілий та приємний у використанні. Розробники приділяють значну увагу створенню адаптивних макетів, плавності анімацій та загальній естетиці, адже саме ці фактори формують перше враження та впливають на бажання користувача продовжувати взаємодію з продуктом. У контексті електронної комерції, якісний дизайн може безпосередньо впливати на конверсію та середній чек.

Зі зростанням обсягів даних, що обробляються мобільними пристроями, та розширенням їх функціоналу, актуалізуються питання безпеки та конфіденційності. Технології шифрування даних, біометрична автентифікація (сканери відбитків пальців, розпізнавання обличчя), а також суворі політики доступу до особистої інформації користувача стають невід`ємною частиною розробки надійних мобільних рішень. Розробники змушені враховувати вимоги законодавства щодо захисту персональних даних, таких як GDPR (загальний регламент про захист даних) в Європі, що додатково стимулює впровадження передових практик безпеки.

На горизонті також з’являються нові тенденції, що обіцяють подальшу трансформацію мобільного досвіду. Серед них можна виділити інтеграцію штучного інтелекту та машинного навчання безпосередньо в мобільні застосунки для персоналізації контенту, розумних рекомендацій та автоматизації рутинних завдань. Також набирають популярності прогресивні веб-застосунки, які поєднують переваги веб-сайтів та нативних мобільних застосунків, пропонуючи швидкий доступ, можливість роботи офлайн та встановлення на головний екран без необхідності завантаження з магазину застосунків. Ці

напрямки, безперечно, впливатимуть на майбутнє мобільних технологій та способи взаємодії користувачів із цифровим світом.

1.2 Електронна комерція та мобільна комерція

Електронна комерція – це сукупність процесів купівлі-продажу товарів і послуг через електронні ресурси (вебсайти, електронні торговельні майданчики, соціальні мережі тощо). Вона відкриває можливості для бізнесу завдяки:

- скороченню витрат на утримання фізичних торговельних точок;
- доступу до ширшої аудиторії потенційних клієнтів;
- спрощенню процесів логістики та оплати;

У межах електронної комерції виділяють мобільну комерцію (m-commerce), що охоплює всі торговельні операції, які здійснюються через мобільні пристрої (смартфони, планшети, та інші) [6]. Завдяки зручності користування мобільними застосунками, m-commerce показує вищі темпи зростання порівняно з традиційною e-commerce. За даними Digital 2022, у деяких країнах частка мобільних транзакцій сягає 70–80 % від загального обсягу онлайн-покупок [7].

Ключові фактори, що сприяють зростанню мобільної комерції:

- повсюдність смартфонів: Смартфон став невід'ємним атрибутом сучасної людини, що забезпечує постійний доступ до торговельних майданчиків;
- оптимізований користувацький досвід: Сучасні мобільні застосунки пропонують значно кращий досвід взаємодії порівняно з мобільними версіями веб-сайтів. Вони швидші, мають інтуїтивну навігацію та можуть використовувати нативні функції пристрою;
- спрощення оплати: Технології мобільних платежів, такі як Google Pay та Apple Pay, дозволяють здійснювати покупки в один дотик, усуваючи

необхідність ручного введення даних платіжної картки, що значно знижує кількість покинутих кошиків;

- персоналізація та маркетинг: Мобільні застосунки є потужним інструментом для персоналізованого маркетингу. Використовуючи дані про попередні покупки, перегляди, місцезнаходження та поведінку користувача, бізнес може формувати індивідуальні пропозиції, знижки та рекомендації. Push-сповіщення дозволяють оперативно інформувати клієнтів про акції або статус замовлення, підтримуючи постійний контакт;
- особливості мобільної комерції;
- персоналізація – мобільні застосунки можуть відстежувати поведінку користувачів, їхню геолокацію та вподобання, надаючи індивідуальні пропозиції;
- зручність оплати – інтеграція з мобільними гаманцями (Apple Pay, Google Pay) спрощує та пришвидшує процес оплати;
- оперативна комунікація – пуш-сповіщення, чат-боти та інтеграція з месенджерами забезпечують швидкий зворотний зв’язок та інформування користувачів[8].

Таким чином, мобільна комерція стає ключовим напрямом розвитку електронної комерції завдяки постійному зростанню кількості мобільних пристроїв та поліпшенню якості мобільного інтернету.

Важливим фактором успіху в мобільній комерції є створення безшовного користувацького досвіду, де кожен етап взаємодії - від першого знайомства з товаром до оформлення замовлення та отримання підтримки - є максимально простим та інтуїтивно зрозумілим. Мобільні застосунки, на відміну від веб-сайтів, адаптованих для мобільних пристроїв, часто пропонують кращу продуктивність, можливість використання нативних функцій пристрою (камера, геолокація, сповіщення) та більш глибокий рівень персоналізації.

Для нішевих ринків, таких як продаж квіткової продукції, мобільні застосунки відкривають унікальні можливості. Вони дозволяють не просто

представити каталог товарів, а й створити емоційний зв’язок з клієнтом через якісні зображення, зручні фільтри для підбору букетів за приводом чи типом квітів, а також надати додаткові сервіси. Можливість швидкого замовлення «в один клік» та відстеження доставки безпосередньо в застосунку значно підвищує цінність такого сервісу для зайнятих користувачів.

Окрім прямої взаємодії з клієнтом, мобільні застосунки у мобільній комерції слугують потужним інструментом для збору аналітичних даних. Інформація про популярні товари, ефективність маркетингових кампаній, шляхи користувача в застосунку та точки «відвалу» дозволяє бізнесу оптимізувати свої пропозиції, покращувати сервіс та приймати обґрунтовані рішення щодо подальшого розвитку. Це, в свою чергу, сприяє підвищенню конкурентоспроможності на динамічному ринку електронної торгівлі.

Динамічний розвиток мобільної комерції також значною мірою зумовлений інтеграцією з соціальними платформами. Такі гіганти, як Instagram та Facebook, активно розвивають власні інструменти для торгівлі, дозволяючи бізнесам створювати вітрини та продавати товари безпосередньо через соціальні мережі, що стирає межі між соціальною взаємодією та шопінгом. Це явище стає потужним каналом залучення, особливо для товарів, що мають сильну візуальну складову. Можливість миттєвого переходу від перегляду контенту до покупки, не покидаючи звичного середовища соціальної мережі, значно спрощує шлях клієнта та підвищує ймовірність імпульсивних покупок.

Крім того, на горизонті мобільної комерції з'являються технології, що обіцяють ще глибше занурення користувача у процес вибору. Наприклад, доповнена реальність може дозволити покупцям віртуально «приміряти» букет до інтер'єру або оцінити його розмір та вигляд у реальному просторі перед покупкою. Хоча масове впровадження таких технологій ще попереду, їх потенціал для підвищення залученості та зменшення кількості повернень є значним, особливо для товарів, де візуальне сприйняття відіграє ключову роль. Аналогічно, голосові помічники та технології голосової комерції поступово інтегруються в мобільний досвід, дозволяючи користувачам здійснювати

покупки за допомогою голосових команд, що може бути особливо зручно в певних ситуаціях, наприклад, під час керування автомобілем або приготування їжі.

1.2.1 Особливості ніші продажу квіткової продукції через мобільні застосунки

Квіти традиційно вважаються одним із найпопулярніших подарунків на різноманітні свята, ювілеї, весілля, дні народження, а також є способом виразити вдячність чи співчуття [11]. Проте процес вибору букета іноді викликає труднощі у користувачів, які:

- мають обмежений час і хочуть оперативно обрати потрібний букет.
- не впевнені, які квіти пасують до певної події чи кому подарувати.
- бажають отримати консультацію від флориста чи експерта, але не мають можливості відвідати фізичний магазин.

Важливо враховувати й сезонні коливання попиту (святкові періоди, День Валентина, 8 Березня, День матері тощо), коли продажі квітів суттєво зростають [12]. Мобільний застосунок може допомогти бізнесу ефективніше керувати цими піковими навантаженнями, пропонуючи можливість попереднього замовлення та оптимізуючи логістику.

Особливістю даного ринку є висока важливість візуальної складової та емоційного фактору. Якість фотографій, зручність перегляду каталогу та можливість легко знайти «саме той» букет безпосередньо впливають на рішення про покупку.

1.2.2 Аналіз конкурентного середовища

Для визначення позиціонування та функціональних вимог до розроблюваного застосунку «BouqetApp» було проведено аналіз існуючих рішень на українському ринку.

«Flowers.ua»

Загальний опис: Провідний гравець на ринку доставки квітів в Україні, що має як веб-сайт, так і розвинений мобільний застосунок. Пропонує широкий асортимент не тільки квітів, але й супутніх товарів (подарунки, солодощі, іграшки) (рис. 1.1).

Сильні сторони:

- впізнаваність бренду та довіра: довгий час на ринку, широка мережа партнерів по всій Україні;
- широкий асортимент: великий вибір готових букетів, а також подарунків, що дозволяє робити комплексні замовлення;
- розвинена логістика: забезпечує доставку у більшості міст України;
- функціональний застосунок: підтримує всі основні етапи замовлення, від вибору до оплати та відстеження.

Слабкі сторони:

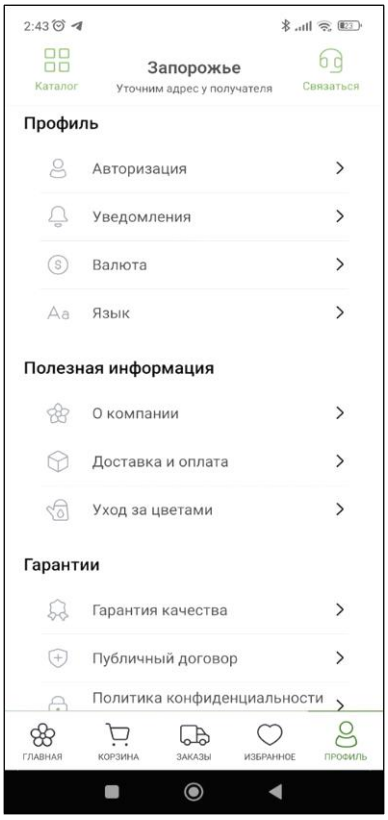
- перевантаженість інтерфейсу: через великий асортимент товарів, що не є квітами, інтерфейс може здаватися перевантаженим для користувача, який шукає виключно букет;
- обмеженість фільтрів: система фільтрації переважно базується на приводах («День народження», «Весілля») або загальних категоріях («Троянди», «Тюльпани»). Відсутня можливість гнучкого підбору букета за декількома типами квітів одночасно, особливо у «строгому» режимі;
- відсутність фокусу на «конструкторі» букетів: Застосунок орієнтований на продаж готових композицій, а не на допомогу у виборі букета за складом.

Великі маркетплейси (напр., категорія «Квіти» на Prom.ua, Rozetka).

Вони не є спеціалізованими квітковими сервісами, але мають відповідні категорії, де представлені товари від різних продавців.



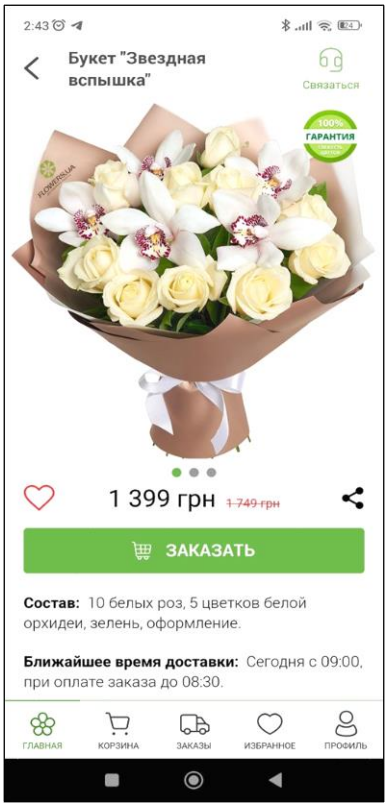
а) екран завантаження



б) сторінка профіля



в) сторінка категорій



г) сторінка букета

Рисунок 1.1 – Демонстрація застосунка

Сильні сторони:

- величезна аудиторія: користувачі, що вже звикли купувати на цих платформах, можуть шукати квіти там же;
- система відгуків: наявність відгуків про конкретних продавців та товари.

Слабкі сторони:

- відсутність спеціалізації: інтерфейс та функціонал не адаптовані під специфіку продажу квітів. відсутні спеціалізовані фільтри, консультації, емоційна складова;
- неконсистентна якість: якість продукції та сервісу може сильно відрізнятися у різних продавців, представлених на платформі;
- складність вибору: користувачеві доводиться переглядати пропозиції від багатьох різних магазинів, що ускладнює процес вибору.

Instagram-магазини та локальні флористичні студіїю. Багато флористів та невеликих магазинів використовують соціальні мережі, зокрема Instagram, як основний канал продажів.

Сильні сторони:

- висока естетика та візуальна привабливість: фокус на якісних фотографіях та унікальному стилі;
- прямий контакт з флористом: можливість отримати персональну консультацію та замовити індивідуальний букет;
- сильна спільнота: формування лояльної аудиторії навколо особистого бренду флориста.

Слабкі сторони:

- відсутність автоматизації: процес замовлення часто відбувається вручну через приватні повідомлення, що незручно та повільно;
- обмеженість каталогу: відсутність структурованого каталогу з фільтрами;

- складність оплати: зазвичай оплата відбувається шляхом переказу на картку, що менш зручно та безпечно порівняно з інтегрованими платіжними системами.
- локальне покриття: зазвичай працюють лише в межах одного міста.

1.3 Обґрунтування розробки та визначення ринкової ніші

Проведений аналіз конкурентного середовища дозволяє зробити висновок, що, незважаючи на наявність сильних гравців, існує вільна ринкова ніша для спеціалізованого мобільного застосунку, орієнтованого саме на полегшення процесу вибору букета.

На відміну від великих гравців типу «Flowers.ua», які фокусуються на широкому асортименті готових пропозицій, «BouqetApp» має на меті вирішити проблему користувача, який не знає, що саме обрати, але має певні уявлення про бажані квіти. Основна конкурентна перевага розроблюваного застосунку полягає в інструменті гнучкої фільтрації за складом букета. Можливість обрати декілька типів квітів та перемикатися між «звичайним» та «строгим» режимами є унікальною функціональністю, яка відсутня у конкурентів.

Таким чином, «BouqetApp» позиціонується не просто як черговий магазин квітів, а як помічник та порадник у виборі, що спрощує процес прийняття рішення для користувача. Це створює додаткову цінність і може залучити аудиторію, яка відчуває «параліч вибору» у великих каталогах і шукає більш простий та цілеспрямований інструмент.

За підсумками розділу, розробка мобільного застосунку «BouqetApp» є актуальною з декількох причин:

Зростання кількості мобільних покупок. Як зазначено вище, частка покупок через смартфони та планшети стрімко збільшується. Створення спеціалізованого застосунку дозволить зайняти перспективну нішу на ринку m-commerce.

Унікальна пропозиція для користувачів. Інтегрований алгоритм підбору букетів за складовими вирішує проблему складного вибору й допомагає клієнтам зробити обґрунтований вибір.

Покращення сервісу та зручності. Можливість швидко знайти потрібний букет, а в перспективі – замовити доставку за кілька кліків, підвищує лояльність клієнтів.

Інтеграція з сучасними технологіями. Застосунок може використовувати push-сповіщення, аналітику поведінки користувачів, геолокацію, що відкриває додаткові можливості для маркетингу та персоналізації пропозицій.

Відсутність прямого аналога. Попри існування сервісів доставки квітів, спеціалізованих застосунків з фокусом на гнучкому підборі за складом на українському ринку відносно небагато. Це створює конкурентну перевагу для нового проєкту.

Таким чином, розробка мобільного застосунку відповідає сучасним тенденціям розвитку електронної та мобільної комерції, а також задовольняє потреби широкого кола користувачів, що робить цей проєкт економічно та соціально доцільним.

2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1 Використані методи та засоби розробки

Розробка мобільного застосунку «BouqetApp» велася з використанням ітеративного підходу. Цей метод передбачає розбивку всього процесу створення програмного продукту на послідовні цикли, або ітерації. Кожна ітерація включає етапи планування, аналізу вимог, проектування, реалізації, тестування та оцінки. Такий підхід дозволив поступово нарощувати функціонал, починаючи з базових екранів та логіки фільтрації, і закінчуючи налаштуванням користувацького інтерфейсу та обробкою даних. Перевагою ітераційної моделі в контексті даного проекту стала можливість раннього виявлення проблем та гнучкого внесення коректив на основі проміжних результатів та тестування. Це особливо важливо при розробці мобільних застосунків, де користувацький досвід відіграє ключову роль.

Наприклад, на першій ітерації могла бути реалізована лише базова структура екрану фільтрів (FilterActivity) з жорстко закодованим списком квітів та перехід на порожній екран списку букетів. На другій ітерації – додавання логіки вибору фільтрів та їх передачі. На третій – реалізація екрану BouquetListActivity з відображенням статичного списку букетів. На четвертій – впровадження логіки фільтрації та динамічного оновлення списку. Таке покрокове додавання функціоналу значно спрощує відлагодження та дозволяє контролювати якість на кожному етапі.

Основні інструменти та технології:

- середовище розробки (IDE): Android Studio Giraffe 2023.3.2;
- мова програмування: Kotlin;
- система збирання проекту: Gradle з використанням Kotlin.

Архітектурні компоненти Android Jetpack:

- AppCompat: для забезпечення зворотної сумісності та використання сучасних елементів UI;
- Material Components for Android: для реалізації дизайну (чіпи, перемикачі, AppBar, теми);
- RecyclerView: для ефективного відображення списків (букетів);
- ConstraintLayout: для створення гнучких та адаптивних користувацьких інтерфейсів.

Формат даних для каталогу: для зберігання даних використовуються файли FilterActivity.kt та BouquetListActivity.kt (хоча для зручності краще використовувати JSON для зберігання інформації про квіти та букети).

2.2 Опис функціонування та сценарії використання застосунку «BouqetApp»

Мобільний застосунок «BouqetApp» розроблено з метою надання користувачам простого та інтуїтивно зрозумілого інструменту для пошуку квіткових букетів за певними критеріями.

Основний функціонал (рис. 2.1):

а) вибір фільтрів: на головному екрані користувач може обрати одну або декілька квіток зі списку доступних. Ці квітки виступають критеріями для фільтрації;

б) режим фільтрації: передбачено два режими пошуку:
   - звичайний режим (за замовчуванням): показуються букети, які містять хоча б одну з обраних користувачем квіток;
   - строгий режим: показуються лише ті букети, які містять усі обрані користувачем квітки;
   - перемикання між режимами здійснюється за допомогою відповідного перемикача на екрані фільтрів;

в) перегляд списку букетів: після вибору фільтрів та натискання кнопки «Підібрати букет» користувач переходить на екран зі списком букетів, що відповідають заданим критеріям. Кожен елемент списку містить зображення букета та його назву;

г) повноекранний перегляд зображення: натискання на зображення букета у списку відкриває його на весь екран для детальнішого перегляду;

д) зміна теми: у меню застосунку доступна опція для перемикання між світлою та темною темами оформлення.

е) додаткові функції меню: перегляд короткого опису застосунку та можливість закрити програму.

Сценарії використання.

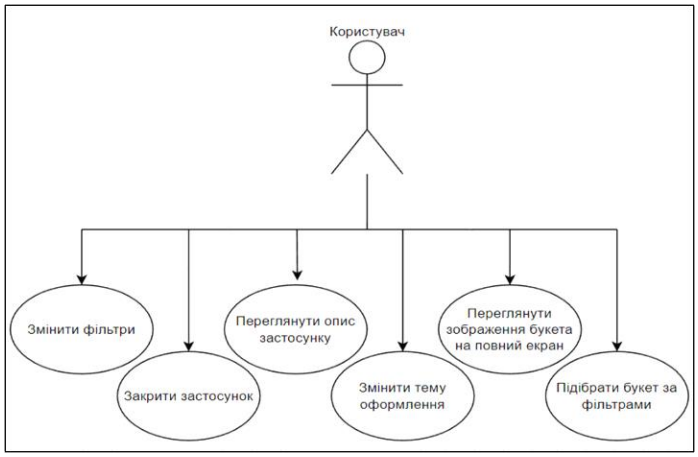


Рисунок 2.1 – Діаграма сценаріїв вкористання

Основні класи (рис. 2.2):

– AppCompatActivity: базовий клас Android, від якого успадковуються екрани;

– FilterActivity, BouquetListActivity, ImageViewerActivity: класи, що представляють кожен екран застосунку. Для них вказані деякі ключові методи та поля, наприклад, populateFlowerChips() для FilterActivity або

loadAndFilterBouquets() для BouquetListActivity. Також вони успадковуються від AppCompatActivity;

– RecyclerView.Adapter та RecyclerView.ViewHolder: базові класи Android для створення списків;
– BouquetAdapter та BouquetViewHolder: класи для відображення списку букетів. BouquetAdapter успадковується від RecyclerView.Adapter, а BouquetViewHolder від RecyclerView.ViewHolder. Також BouquetAdapter створює та використовує BouquetViewHolder;
– Flower та Bouquet: data-класи, що представляють моделі даних (квітку та букет). Для них вказані основні атрибути (id, name, imageResId).

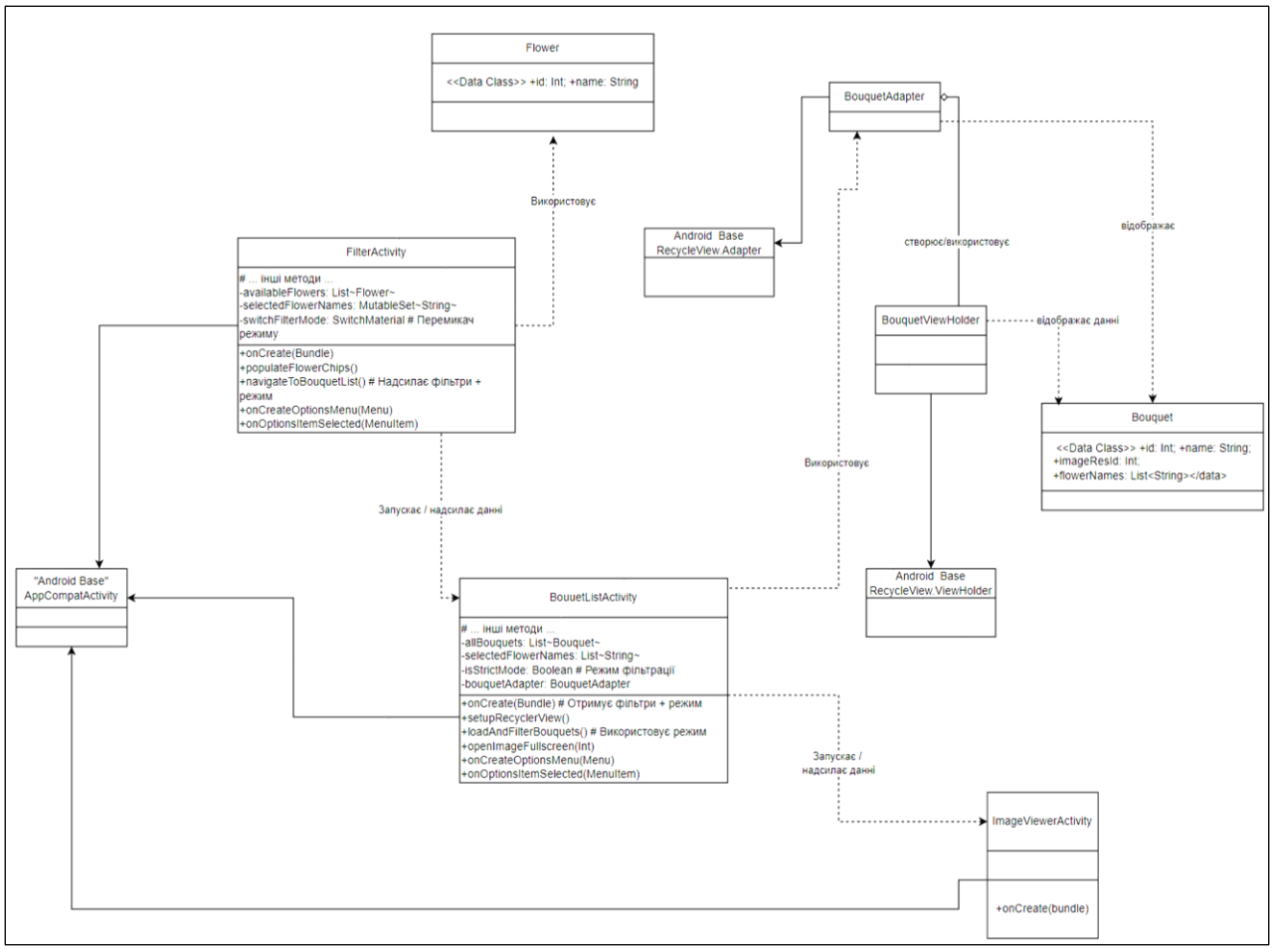


Рисунок 2.2 – Діаграма класів

Зв’язки між класами (рис. 2.2):

– успадкування (<|--): показує, що один клас є нащадком іншого (наприклад, FilterActivity є нащадком AppCompatActivity);
– асоціація/Використання (..>): пунктирна лінія зі стрілкою показує, що один клас використовує інший. Наприклад, FilterActivity використовує Flower (для відображення фільтрів) та CatalogLoader (для отримання списку квітів). BouquetListActivity використовує BouquetAdapter та CatalogLoader;
– агрегація/Композиція (o--): Лінія з порожнім ромбиком показує, що один клас «містить» або «складається з» іншого (наприклад, BouquetAdapter створює та використовує об’єкти BouquetViewHolder).

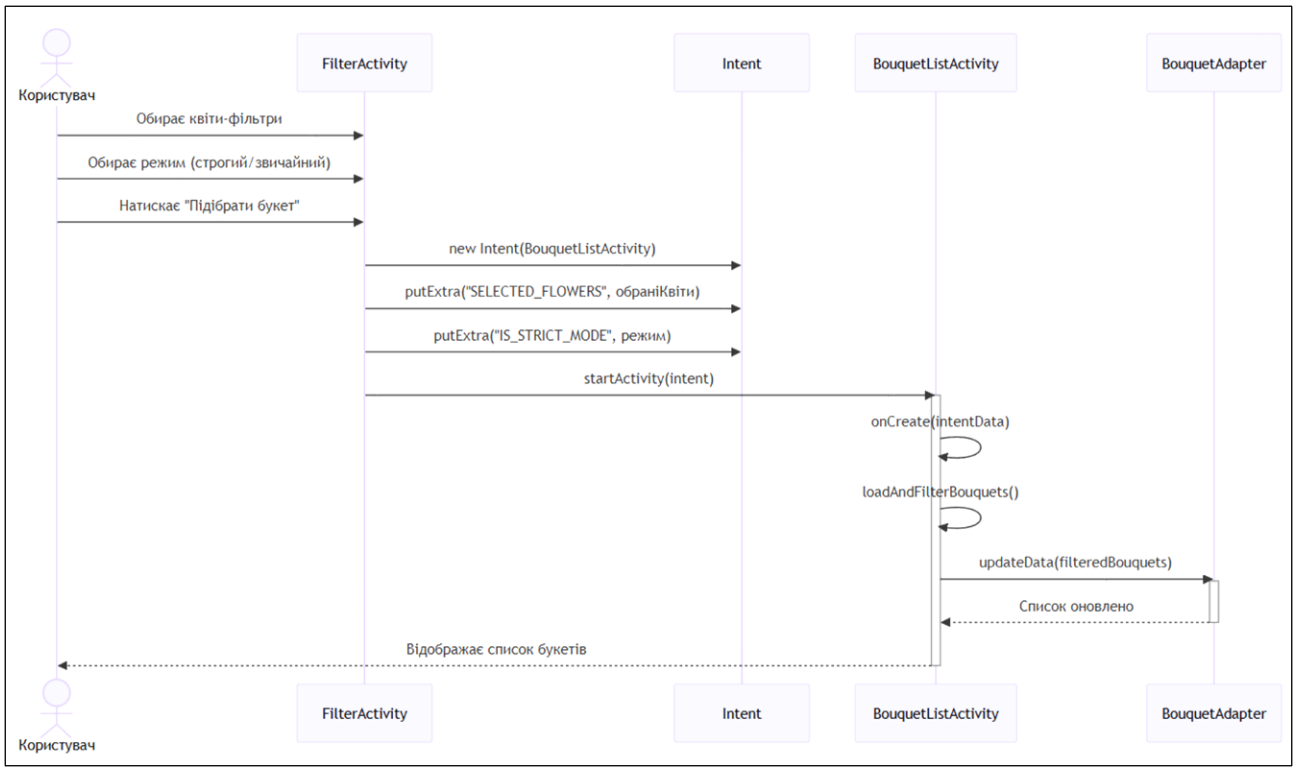


Рисунок 2.3 – Діагарма послідовності

Сценарій 1 Пошук букета (звичайний режим):

– користувач відкриває застосунок. З’являється екран «Фільтри»;
– користувач обирає одну або декілька квіток (наприклад, «Троянди», «Лілії»);
– користувач залишає перемикач «Строгий режим» у вимкненому стані;

– користувач натискає кнопку «Підібрати букет»;
– система відображає екран «Підходящі букети», де показані всі букети, що містять або троянди, або лілії, або обидві квітки.

Сценарій 2 Пошук букета (строгий режим):

– користувач знаходиться на екрані «Фільтри»;
– користувач обирає квітки (наприклад, «Еустома», «Зелень»);
– користувач вмикає перемикач «Строгий режим»;
– користувач натискає кнопку «Підібрати букет»;
– система відображає екран «Підходящі букети», де показані лише ті букети, які одночасно містять і еустому, і зелень.

Сценарій 3 Перегляд та повернення

– користувач знаходиться на екрані «Підходящі букети»;
– користувач натискає на зображення одного з букетів;
– зображення відкривається на весь екран;
– користувач натискає на зображення (або системну кнопку «Назад»);
– система повертає користувача до списку «Підходящі букети»;
– користувач натискає кнопку «Змінити фільтр»;
– система повертає користувача на екран «Фільтри».

Процес розробки програмного забезпечення рідко буває абсолютно гладким. Весь процес відлагодження супроводжувався активним використанням інструментів Android Studio, таких як Logcat для аналізу помилок під час розробки для діагностики проблем збирання. Ітеративний підхід дозволив послідовно виявляти та виправляти помилки.

У ході виконання було розроблено мобільний застосунок «BouqetApp». Було обрано сучасні інструменти та підходи до розробки під платформу Android. Незважаючи на низку технічних труднощів, пов’язаних з конфігурацією проекту, обробкою ресурсів та налаштуванням користувацького інтерфейсу, додаток працює як було заплановано. Проте, застосунок потребує подальшого вдосконалення для оптимізації, додавання нового контенту та платіжних систем.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Реалізація ключових модулів та функціоналу

Процес реалізації мобільного застосунку «BouqetApp» передбачав розробку декількох взаємопов'язаних модулів, кожен з яких відповідає за певну частину функціоналу. Нижче детально розглянуто особливості програмної реалізації цих модулів.

3.1.1 Модуль фільтрації та вибору режиму

Цей модуль є вхідною точкою для користувача та забезпечує основний функціонал для визначення критеріїв пошуку букетів.

Однією з ключових особливостей є динамічне створення елементів Chip для кожного доступного типу квітів. Дані про доступні квіти (availableFlowers: List<Flower>) визначені як приватна властивість безпосередньо у класі FilterActivity. Для кожного об'єкта Flower з цього списку створюється екземпляр com.google.android.material.chip.Chip. Важливою деталлю реалізації є програмне застосування кастомного стилю R.style.MyLargerFilterChip до кожного чіпа за допомогою ContextThemeWrapper (рис. 3.1). ContextThemeWrapper дозволяє застосувати певну тему або стиль до конкретного View або ієрархії View, не впливаючи на глобальну тему застосунку. Це особливо корисно при динамічному створенні UI-елементів, коли потрібно забезпечити їх відповідність певному дизайну. У даному випадку, це дозволило централізовано керувати виглядом (розміром, відступами, шрифтом) усіх фільтрів, не дублюючи атрибути в XML та не перевантажуючи код Activity зайвими налаштуваннями вигляду.

Стан обраних фільтрів зберігається у множині selectedFlowerNames: MutableSet<String>. Використання MutableSet є доцільним, оскільки множина автоматично забезпечує унікальність елементів (користувач не може обрати

одну й ту саму квітку двічі як окремий фільтр) та надає ефективні операції додавання (add) та видалення (remove).

```
всередині FilterActivity
private fun populateFlowerChips() {
    chipGroupFlowers.removeAllViews()

    // Створюємо контекст з нашим кастомним стилем для Chip
    val chipContext = ContextThemeWrapper(this, R.style.MyLargerFilterChip)

    availableFlowers.forEach { flower ->
        // Створюємо Chip з новим контекстом (стилем)
        val chip = Chip(chipContext).apply {
            text = flower.name
            isCheckable = true
            // Слухач для відстеження обраних фільтрів
            setOnCheckedChangeListener { _, isChecked ->
                if (isChecked) {
                    selectedFlowerNames.add(flower.name)
                } else {
                    selectedFlowerNames.remove(flower.name)
                }
            }
            isChecked = selectedFlowerNames.contains(flower.name)
// Відновлення стану
        }
        chipGroupFlowers.addView(chip)
    }
}
```

Рисунок 3.1 – Фрагмент коду з FilterActivity.kt – динамічне створення Chip

Стан перемикача «строгого режиму» (switchFilterMode: SwitchMaterial) зчитується безпосередньо перед переходом на екран списку букетів за допомогою властивості isChecked. Ці дані (список назв обраних квітів, конвертований в ArrayList для передачі через Intent, та логічне значення режиму) упаковуються в Intent та передаються до BouquetListActivity. Використання putStringArrayListExtra() для передачі списку рядків та putBooleanExtra() для режиму є стандартними та ефективними практиками для передачі невеликих обсягів даних між компонентами Android.

Навігація: Перехід до наступного екрану (BouquetListActivity) ініціюється натисканням на buttonFindBouquet. Метод navigateToBouquetList() відповідає за створення об'єкта Intent, вказуючи клас цільової Activity, додавання до нього необхідних даних (фільтри та режим) та запуск BouquetListActivity за допомогою методу startActivity().

3.1.2 Модуль відображення списку букетів та логіка фільтрації

Цей модуль відповідає за отримання критеріїв пошуку, фільтрацію основного каталогу букетів та відображення результатів користувачеві у зручному для сприйняття вигляді.

В методі onCreate() BouquetListActivity отримує дані, передані з FilterActivity, за допомогою intent.getStringArrayListExtra (FilterActivity.EXTRA_SELECTED_FLOWERS) та intent.getBooleanExtra (FilterActivity.EXTRA_IS_STRICT_MODE, false). Використання констант (EXTRA_SELECTED_FLOWERS, EXTRA_IS_STRICT_MODE), оголошених у FilterActivity (в companion object), є хорошою практикою для уникнення помилок при роботі з ключами Intent. Повний список букетів (allBouquets: List<Bouquet>) також визначений як приватна властивість безпосередньо у класі BouquetListActivity.

Реалізація логіки фільтрації ключова частина функціоналу, реалізована в методі loadAndFilterBouquets() (рис. 3.2). Даний метод демонструє ефективне використання функціональних можливостей Kotlin для роботи з колекціями:

- якщо список selectedFlowerNames порожній (користувач не обрав жодного фільтра), то для відображення береться повний список allBouquets;
- для строгого режиму (isStrictMode == true) використовується функція розширення filter для колекції allBouquets, а всередині неї – функція all { ... } для selectedFlowerNames. Умова bouquet.flowerNames.contains(selectedFlower) перевіряється для

кожного обраного фільтра. Лише ті букети, для яких ця умова виконується для всіх обраних фільтрів, потрапляють у результат;

– для звичайного режиму (isStrictMode == false) також використовується filter, але всередині неї – функція any { ... } для selectedFlowerNames. У цьому випадку достатньо, щоб букет містив хоча б один з обраних фільтрів, щоб він потрапив у результат.

Такий підхід є декларативним та читабельним, дозволяючи чітко виразити логіку фільтрації.

```
// всередині BouquetListActivity
private fun loadAndFilterBouquets() {
    val filteredBouquets = if (selectedFlowerNames.isEmpty()) {
        allBouquets // Якщо фільтри не обрані, показуємо все
    } else {
        if (isStrictMode) {
            // Строгий режим: всі обрані квіти мають бути в букеті
            allBouquets.filter { bouquet ->
                selectedFlowerNames.all { selectedFlower ->
                    bouquet.flowerNames.contains(selectedFlower)
                }
            }
        } else {
            // Звичайний режим: хоча б одна обрана квітка має бути в букеті
            allBouquets.filter { bouquet ->
                selectedFlowerNames.any { selectedFlower ->
                    bouquet.flowerNames.contains(selectedFlower)
                }
            }
        }
    }
    bouquetAdapter.updateData(filteredBouquets) // Оновлення адаптера
}
```

Рисунок 3.2 – Фрагмент коду з BouquetListActivity.kt – логіка фільтрації

Для відображення списку букетів використовується RecyclerView з LinearLayoutManager (для вертикального списку). Адаптер BouquetAdapter (class BouquetAdapter(...):RecyclerView.Adapter<BouquetAdapter.BouquetViewHolder> відповідає за створення представлень для кожного елемента списку (onCreateViewHolder) та за прив'язку даних з об'єкта Bouquet до UI-елементів у відповідному BouquetViewHolder (onBindViewHolder). Метод

bouquetAdapter.updateData(filteredBouquets) викликається після фільтрації для оновлення відображуваних даних, при цьому всередині адаптера викликається notifyDataSetChanged(), що сповіщає RecyclerView про необхідність переробити список. BouquetViewHolder реалізує патерн ViewHolder, зберігаючи посилання на View (наприклад, ImageView для зображення букета та TextView для його назви), що дозволяє уникнути повторних викликів findViewById() під час прокрутки та значно підвищує продуктивність. Обробка натискання на елемент списку реалізована через передачу лямбда-функції onItemClick в конструктор адаптера, яка потім викликається у ViewHolder при натисканні на itemView.Цей підхід дозволяє гнучко керувати тим, які букети будуть показані користувачеві.

3.1.3 Механізм перемикання тем

Застосунок «BouqetApp» підтримує динамічне перемикання між світлою та темною темами оформлення, що є стандартною вимогою до сучасних мобільних додатків для покращення користувацького досвіду та зменшення навантаження на очі в умовах низької освітленості.

Основний механізм перемикання тем базується на використанні класу androidx.appcompat.app.AppCompatDelegate та його статичного методу setDefaultNightMode(). При виборі користувачем відповідного пункту меню викликається метод toggleTheme(), який визначає поточний активний режим нічної теми (resources.configuration.uiMode and Configuration. UI_MODE_NIGHT_MASK) і встановлює протилежний:

Якщо поточна тема світла (Configuration.UI_MODE_NIGHT_NO), встановлюється темна (AppCompatDelegate.MODE_NIGHT_YES).

Якщо поточна тема темна (Configuration.UI_MODE_NIGHT_YES), встановлюється світла (AppCompatDelegate.MODE_NIGHT_NO).

Для того, щоб зміни теми застосувалися негайно, після виклику setDefaultNightMode() в BouquetListActivity викликається метод recreate(), який повністю перезапускає поточну Activity. У FilterActivity recreate() не

викликається для збереження стану обраних фільтрів при зміні теми (що є компромісним рішенням для збереження стану UI без використання ViewModel чи onSaveInstanceState для всіх елементів).

Основою для тем слугує Theme.Material3.DayNight.NoActionBar (або аналогічна з Material Components, залежно від версії бібліотеки), яка автоматично підтримує перемикання. Було створено базову тему Base.Theme.BouqetApp у res/values/themes.xml та її нічну версію у res/values-night/themes.xml.

Кольори: Рекомендується використовувати атрибути теми для визначення кольорів в XML-макетах замість жорстко заданих значень кольорів. Це дозволяє кольорам автоматично адаптуватися до поточної (світлої чи темної) теми. У проекті це частково реалізовано через стандартні атрибути Material Design.

Стиль спливаючого меню Toolbar: Для забезпечення коректного відображення тексту та фону спливаючого меню Toolbar в обох темах було створено кастомний атрибут теми myPopupOverlayTheme в res/values/attrs.xml:

Потім цей атрибут використовується в XML-макеті Toolbar за допомогою app:popupTheme="?attr/myPopupOverlayTheme". Це гарантує, що меню завжди матиме відповідний контрастний вигляд.

Фрагмент з themes.xml (світла тема) (рис. 3.3).

```
<style name="Base.Theme.BouqetApp"
parent="Theme.Material3.DayNight.NoActionBar">
   <item
name="myPopupOverlayTheme">@style/ThemeOverlay.Material3.Light</item>
</style>
```

Рисунок 3.3 – Фрагмент з themes.xml (світла тема)

Фрагмент з themes.xml (темна тема) (рис. 3.4).

```
<style name="Base.Theme.BouqetApp"
parent="Theme.Material3.DayNight.NoActionBar">
   <item
name="myPopupOverlayTheme">@style/ThemeOverlay.Material3.Dark</item>
</style>
```

Рисунок 3.4 – Фрагмент з themes.xml (темна тема)

Використання в Toolbar (XML макет Activity) (рис. 3.5).

```
<androidx.appcompat.widget.Toolbar
   android:id="@+id/toolbar"
   ...
   app:popupTheme="?attr/myPopupOverlayTheme" />
```

Рисунок 3.5 – Використання в Toolbar (XML макет Activity)

3.2 Особливості розробки користувацького інтерфейсу

Розробка інтуїтивно зрозумілого та естетично привабливого користувацького інтерфейсу є критично важливою для успіху будь-якого мобільного застосунку. У «BouqetApp» було докладено зусиль для створення чистого та функціонального дизайну, дотримуючись принципів Material Design.

Використання ConstraintLayout: Як зазначалося раніше, ConstraintLayout було обрано основним менеджером розмітки для всіх екранів. Це дозволило створювати складні відносні позиціонування елементів без необхідності використання глибоко вкладених макетів, що позитивно впливає на продуктивність. Наприклад, на екрані FilterActivity кнопка «Підібрати букет» легко прикріплюється до нижнього краю екрану, а ChipGroup – до заголовка та

перемикача, незалежно від їх висоти. Аналогічно, в BouquetListActivity RecyclerView займає весь доступний простір між Toolbar та нижньою кнопкою.

Стилізація Chip: Для покращення візуального сприйняття фільтрів (квіток) та забезпечення їх кращої інтерактивності було створено кастомний стиль MyLargerFilterChip у файлі res/values/styles.xml. Цей стиль успадковується від Widget.MaterialComponents.Chip.Filter та дозволяє централізовано налаштувати такі параметри, як мінімальна висота (chipMinHeight), внутрішні відступи (chipStartPadding, chipEndPadding, textStartPadding, textEndPadding), розмір тексту (android:textSize) та колір тексту (android:textColor), який береться з поточної теми застосунку (?attr/colorOnSurface) для коректного відображення у світлому та темному режимах. Програмне застосування цього стилю через ContextThemeWrapper є гнучким способом кастомізації динамічно створюваних елементів.

Масштабування SwitchMaterial: Для візуального збільшення перемикача режимів фільтрації та покращення його помітності на екрані FilterActivity було використано атрибути android:scaleX="1.2" та android:scaleY="1.2" безпосередньо в XML-макеті. Це простий спосіб збільшити візуальний розмір елемента без зміни його внутрішніх розмірів або відступів.

Адаптивне відображення зображень: У списку букетів (макет item_bouquet.xml) для ImageView використовується android:scaleType="fitCenter". Цей атрибут гарантує, що все зображення буде видимим всередині ImageView, зберігаючи його пропорції. Зображення буде масштабоване так, щоб воно повністю вмістилося, що може призвести до появи порожніх областей, якщо пропорції зображення та ImageView не збігаються. Альтернатива centerCrop обрізала б зображення, заповнюючи весь ImageView, що було визнано менш бажаним для перегляду букетів. Аналогічний підхід (fitCenter) використовується і в ImageViewerActivity для повноекранного перегляду.

Обробка відсутності результатів: Якщо після застосування фільтрів список букетів порожній, користувачеві відображається коротке повідомлення (Toast)

«Підходящих букетів не знайдено». Це є важливим елементом зворотного зв'язку, який інформує користувача про результат його дій.

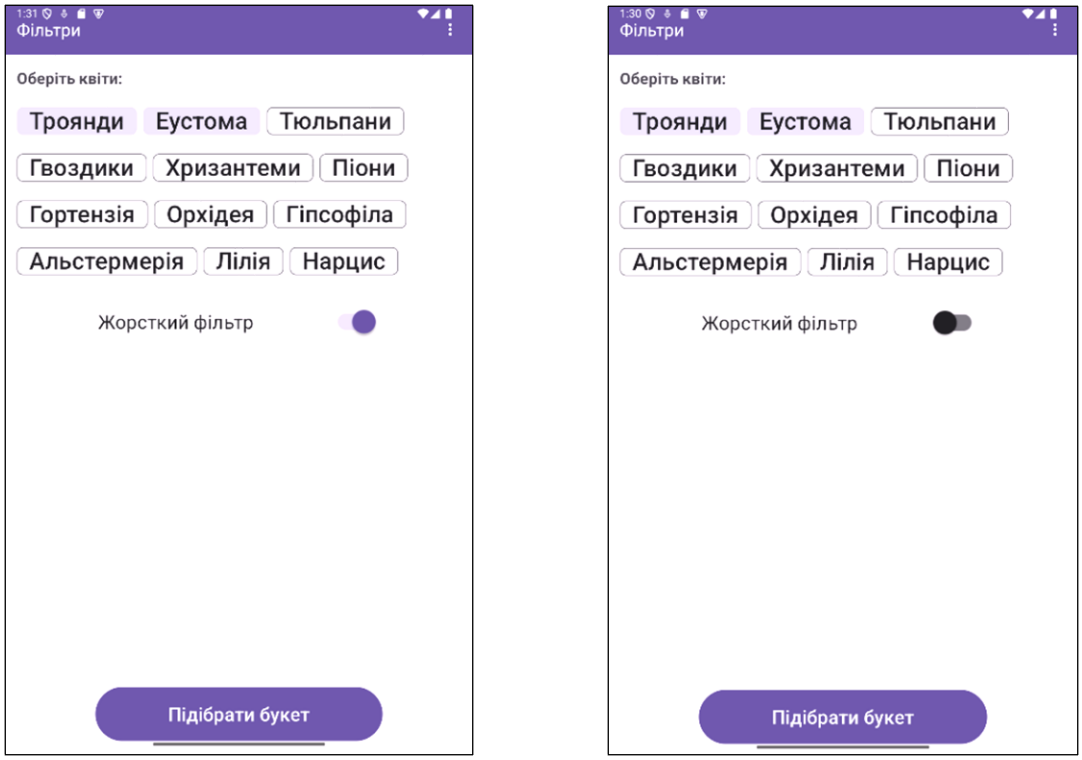
Використання векторної графіки (де можливо): Хоча в даному проекті зображення букетів є растровими, для іконок меню (якщо б вони були кастомними) та інших невеликих графічних елементів перевага надавалася б векторним форматам (SVG, Vector Drawable). Це забезпечує чітке відображення на екранах з будь-якою щільністю пікселів та зменшує розмір застосунку. Для стандартних іконок меню використовуються системні ресурси або ресурси з Material Components.

3.3 Тестування та відлагодження

Процес тестування та відлагодження є невід'ємною частиною життєвого циклу розробки програмного забезпечення і для проекту «BouqetApp» йому приділялася належна увага на кожному етапі.

Основним методом перевірки функціональності застосунку було ручне тестування. Воно проводилося на емуляторі Android Nexus 7 API 35 та на фізичному пристрої Redmi Note 10 Pro для перевірки поведінки в реальних умовах. Ручне тестування включало перевірку всіх основних користувацьких сценаріїв:

- вибір одного або декількох фільтрів (рис. 3.6); робота звичайного режиму (перевірка, що відображаються букети, що містять хоча б одну обрану квітку); робота строгого режиму (перевірка, що відображаються лише букети, що містять усі обрані квітки); перевірка поведінки при відсутності обраних фільтрів; перевірка відображення повідомлення при відсутності результатів;

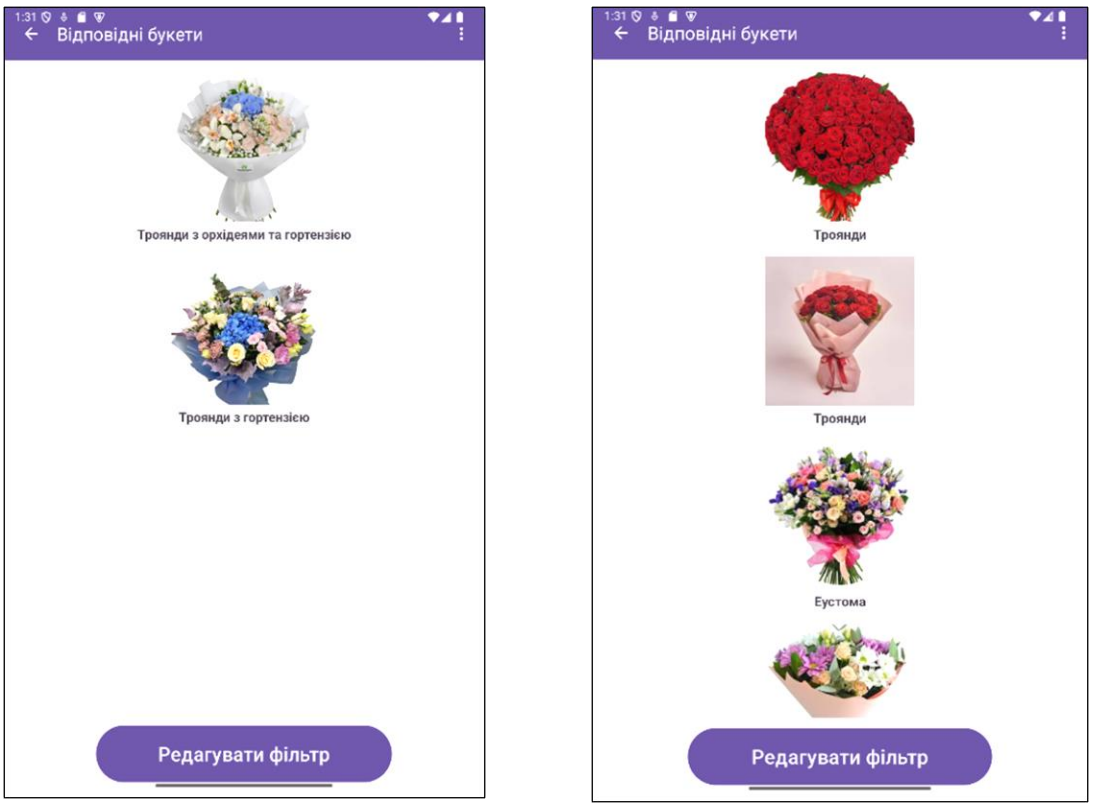


а) тестування жорсткої фільтрації

б) тестування звичайної фільтрації

Рисунок 3.6 – Демонстрація роботи фільтрів

– переходи між екранами FilterActivity, BouquetListActivity та ImageViewerActivity; коректна робота кнопки «Назад» на системному рівні та програмно реалізованої кнопки «Змінити фільтр»;
– коректне обрання/зняття вибору з Chip, робота SwitchMaterial, клікабельність елементів списку RecyclerView, повноекранний перегляд та закриття зображення (рис. 3.7);
– відкриття меню, перемикання теми (перевірка візуальних змін та коректності відображення меню в обох темах), відображення діалогового вікна з описом, закриття застосунку;
– перевірка збереження стану при зміні орієнтації екрану;
– інструмент Logcat в Android Studio був незамінним для діагностики помилок під час виконання. Він дозволяв відстежувати винятки, такі як



а) результат жорсткої фільтрації

б) результат звичайної фільтрації

Рисунок 3.7 – Результат роботи фільтрації

ClassNotFoundException (що виникала на ранніх етапах через проблеми з конфігурацією AndroidManifest.xml), NullPointerException (наприклад, при спробі доступу до неініціалізованого UI-елемента перед викликом setContentView або findViewById), Resources.NotFoundException (якщо ім'я ресурсу для зображення букета було вказано неправильно), або помилки, пов'язані з некоректною роботою з колекціями. Логи також використовувалися для виведення проміжних значень змінних для розуміння потоку виконання програми;

– на початкових етапах розробки, а також при внесенні змін до конфігураційних файлів Gradle (build.gradle.kts) або маніфесту, виникали помилки збирання. Повідомлення у вкладці «Build Output» в Android Studio надавали детальну інформацію про причину помилки, що дозволяло їх виправляти. Типовими проблемами були: невідповідність версій SDK, помилки в синтаксисі XML-макетів,

неправильні імена пакетів або ресурсів, що призводили до проблем з генерацією класу R, конфлікти залежностей;

- для більш глибокого аналізу поведінки коду використовувався вбудований відлагоджувач. Встановлення точок зупину (breakpoints) у ключових місцях коду (наприклад, у методах фільтрації, обробниках натискань, при ініціалізації Activity) дозволяло покроково виконувати програму, перевіряти значення змінних у реальному часі та розуміти логіку роботи складних алгоритмів;
- ітеративний підхід до розробки значно полегшив процес тестування та відлагодження, оскільки кожна нова порція функціоналу тестувалася окремо перед інтеграцією з рештою системи. Це дозволяло локалізувати та виправляти помилки на ранніх стадіях, не допускаючи їх накопичення.

3.4 Перспективи розвитку проекту

Мобільний застосунок «BouqetApp» у його поточній реалізації є функціональним прототипом, що демонструє основні можливості підбору квіткових букетів. Однак існує значний потенціал для подальшого розвитку та розширення його функціоналу, що може перетворити його на повноцінний комерційний продукт:

- перехід на зовнішнє сховище даних: Як обговорювалося, першочерговим кроком для підвищення гнучкості є перенесення каталогу квітів та букетів з жорстко закодованих списків у зовнішнє джерело. Це може бути JSON-файл в assets (як проміжний крок), або, що краще, інтеграція з серверною частиною через API. Це дозволить динамічно оновлювати асортимент, ціни, наявність без необхідності оновлення самого застосунку. Розглянути використання хмарних баз даних, таких як Firebase Firestore або Realtime Database, для зберігання та синхронізації даних;

- розширення інформації про букет: Додавання екрану з детальною інформацією про кожен букет, де б відображалися не тільки назва та зображення, а й повний склад (перелік квітів та декоративних елементів), опис від флориста, можливі розміри (малий, середній, великий), точна ціна, відгуки покупців, можливо, інформація про походження квітів або особливості догляду;
- функціонал «Обране» та історія переглядів: Можливість для користувача додавати букети, що сподобалися, до списку «Обране» для швидкого доступу в майбутньому. Також корисною була б історія нещодавно переглянутих букетів, щоб користувач міг легко повернутися до варіантів, що його зацікавили;
- система пошуку та сортування: Додавання текстового поля для повнотекстового пошуку букетів за назвою або ключовими словами (наприклад, «весняний букет», «букет для мами»). Реалізація можливості сортування списку букетів за різними критеріями: ціною (від дешевих до дорогих і навпаки), популярністю (на основі рейтингів або кількості замовлень), датою додавання (новіші спочатку);
- інтеграція функціоналу замовлення та оплати: Це ключовий крок для перетворення застосунку на інструмент мобільної комерції;
- кошик для додавання обраних букетів та перегляду вмісту;
- форму оформлення замовлення (введення адреси доставки, імені та телефону отримувача, контактних даних замовника, вибір дати та часу доставки, можливість додати листівку з текстом);
- інтеграцію з популярними українськими платіжними системами (наприклад, LiqPay, Portmone, Fondy) або через Google Pay для спрощення процесу оплати;
- систему відстеження статусу замовлення (наприклад, «Прийнято», «Збирається флористом», «Передано кур'єру», «Доставлено»);
- система реєстрації та авторизації користувачів: Для збереження історії замовлень, списку обраного, адрес доставки та персоналізованих

пропозицій необхідна буде система облікових записів користувачів. Це може бути реалізовано через Firebase Authentication (з підтримкою входу через email/пароль, Google, Facebook);

- рекомендаційна система. На основі історії переглядів, покупок та вподобань користувача, а також загальних тенденцій, можна реалізувати систему персоналізованих рекомендацій букетів («Вам також може сподобатися...», «З цим букетом часто купують...»);
- онлайн-консультації з флористом. Інтеграція чату або можливості відеодзвінка з флористом для отримання професійної консультації щодо вибору букета, особливо для складних або індивідуальних замовлень;
- геолокаційні сервіси. Використання геолокації для автоматичного визначення міста користувача (для показу релевантних пропозицій), визначення найближчих пунктів самовивозу (якщо така опція буде) або оптимізації маршрутів доставки;
- Push-сповіщення. Інформування користувачів про нові надходження в каталозі, спеціальні пропозиції, акції, знижки, нагадування про важливі дати (наприклад, дні народження близьких, збережені користувачем), а також про зміну статусу замовлення;
- подальше вдосконалення дизайну на основі відгуків користувачів та сучасних тенденцій, додавання плавних анімацій переходів між екранами, покращення загальної навігації та юзабіліті. Можливе впровадження елементів доповненої реальності (AR) для візуалізації букетів в інтер'єрі користувача, що дозволить краще оцінити їх вигляд та розмір;
- переклад інтерфейсу та контенту на інші мови (наприклад, англійську) для потенційного виходу на ширший ринок;
- впровадження системи бонусів, накопичувальних знижок або спеціальних пропозицій для постійних клієнтів.

Кожен з цих напрямків вимагатиме додаткового аналізу, проектування та реалізації, але вони значно розширять можливості застосунку та його цінність для користувачів, перетворюючи його з простого каталогу на повноцінну платформу для вибору та замовлення квітів.

У ході виконання третього розділу кваліфікаційної роботи було здійснено безпосередню програмну реалізацію мобільного застосунку «BouqetApp» на основі попередньо розроблених проектних рішень та проведеного аналізу. Ключову увагу було приділено втіленню основного функціоналу, зокрема розробці модулів фільтрації букетів за типами квітів та обраним користувачем режимом (звичайний або строгий), відображення динамічного списку результатів за допомогою RecyclerView. Дані каталогу квітів та букетів на даному етапі були інтегровані безпосередньо у вихідний код класів FilterActivity та BouquetListActivity для спрощення прототипування.

Було детально розглянуто та обґрунтовано особливості програмної реалізації користувацького інтерфейсу з використанням компонентів Material Design, таких як Chip для інтерактивних фільтрів та SwitchMaterial для перемикання режимів фільтрації. Програмне створення UI-елементів та застосування кастомних стилів через ContextThemeWrapper дозволило досягти необхідної гнучкості та візуальної консистентності. Важливим аспектом стала реалізація підтримки світлої та темної тем оформлення, що покращує користувацький досвід та відповідає сучасним стандартам мобільної розробки.

У розділі наведено приклади ключових фрагментів коду мовою Kotlin, що ілюструють логіку роботи основних модулів, зокрема динамічного створення фільтрів та ефективних алгоритмів фільтрації списку букетів з використанням функціональних можливостей Kotlin для роботи з колекціями. Додатково, представлено UML діаграми (діаграму класів та діаграму послідовності для ключового сценарію), що візуалізують архітектуру та динаміку взаємодії компонентів системи.

Також було описано підходи до тестування та відлагодження застосунку, які включали ручне тестування основних користувацьких сценаріїв та

використання інструментів Android Studio, таких як Logcat та вбудований відлагоджувач, для виявлення та усунення помилок. На завершення, визначено та деталізовано потенційні напрямки для подальшого розвитку та вдосконалення проекту, що підкреслює масштабованість розробленого рішення.

Таким чином, реалізація застосунку «BouqetApp» продемонструвала успішне застосування отриманих теоретичних знань та практичних навичок у сфері розробки мобільних застосунків під платформу Android з використанням мови Kotlin та сучасних технологій і підходів. Створено функціональний прототип, що відповідає поставленим завданням та готовий до подальшого вдосконалення.

Було описано процес програмної реалізації мобільного застосунку «BouqetApp». Розглянуто вибір технологій, архітектурні підходи, реалізацію ключових функціональних блоків, а також проблеми, що виникали, та шляхи їх вирішення. Розробка велася з використанням сучасних інструментів та практик, що дозволило створити функціональний прототип застосунку, готовий до подальшого розвитку.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно вирішено поставлене завдання – розроблено прототип мобільного застосунку «Підбір та замовлення квітів» для платформи Android. Було досягнуто головної мети, яка полягала у створенні інтуїтивно зрозумілого інструменту для зручного пошуку та вибору квіткових букетів за допомогою гнучкої системи фільтрів.

На основі проведеної роботи можна зробити наступні висновки:

- у першому розділі було проведено комплексний аналіз предметної області. Досліджено сучасний стан та тенденції розвитку мобільних технологій, а також ринку електронної та мобільної комерції. Аналіз показав стійке зростання популярності мобільних покупок та високий попит на спеціалізовані нішеві застосунки. Оцінка конкурентного середовища на українському ринку сервісів доставки квітів виявила відсутність рішень з фокусом на гнучкому підборі букетів за їх складом, що підтвердило актуальність та ринкову доцільність розроблюваного проекту;
- у другому розділі було детально описано процес проектування застосунку. Обґрунтовано вибір технологічного стеку, зокрема мови програмування Kotlin та середовища Android Studio, як сучасних та ефективних інструментів для нативної розробки під Android. Визначено архітектуру застосунку, що базується на взаємодії декількох Activity, та описано структуру моделей даних. Спроектовано користувацькі інтерфейси для всіх основних екранів та розроблено ключові сценарії взаємодії користувача з застосунком. На цьому етапі було закладено міцний концептуальний фундамент для подальшої програмної реалізації;
- у третьому розділі було представлено безпосередню програмну реалізацію мобільного застосунку «BouqetApp». Детально описано

розробку ключових функціональних модулів: модуля фільтрації з двома режимами (звичайним та строгим), модуля відображення списку букетів за допомогою RecyclerView, а також механізму перемикання світлої та темної тем. Особливу увагу приділено практичним аспектам реалізації, зокрема динамічному створенню UI-елементів, застосуванню кастомних стилів та ефективному використанню функціональних можливостей мови Kotlin для реалізації логіки фільтрації. Також описано процес тестування та відлагодження, що дозволило забезпечити стабільну роботу основного функціоналу.

Результатом кваліфікаційної роботи є функціональний програмний продукт – прототип мобільного застосунку, який успішно реалізує поставлені на початку завдання. Розроблений застосунок демонструє унікальну торгову пропозицію завдяки гнучкій системі фільтрації та може слугувати основою для створення повноцінного комерційного продукту. Було також визначено перспективи подальшого розвитку проекту, що включають інтеграцію з серверною частиною, розширення функціоналу (кошик, оплата, система авторизації) та покращення користувацького досвіду.

Таким чином, мету кваліфікаційної роботи досягнуто, та отримано досвід проектування та розробки мобільних інформаційних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. Statista. Number of smartphone users worldwide from 2016 to 2022. Statista. URL : https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide (дата звернення : 10.05.2025).

2. Інтернет Асоціація України. Звіт про розвиток інтернет-ринку в Україні. Інтернет Асоціація України. URL : https://inau.ua (дата звернення : 10.05.2025).

3. Android Open Source Project (AOSP). About the Android Open Source Project. Android Open Source Project. URL : https://source.android.com (дата звернення : 11.05.2025).

4. Flutter. Flutter documentation. Flutter. URL : https://flutter.dev/docs (дата звернення : 11.05.2025).

5. Google Cloud. Firebase Documentation. Firebase. URL : https://firebase.google.com/docs (дата звернення : 11.05.2025).

6. We Are Social & Hootsuite. Digital 2022: Global Overview Report. We Are Social. URL : https://wearesocial.com/uk/digital-2022 (дата звернення : 12.05.2025).

7. Batra D. Mobile Commerce: Technologies, Services, and Business Models. IGI Global, 2018. 366 с.

8. eMarketer. Global Ecommerce Update 2021. eMarketer. URL : https://www.emarketer.com/content/global-ecommerce-update-2021 (дата звернення : 12.05.2025).

9. Huang L. V., Benyoucef M. User preferences of social features on social commerce websites: An empirical study. *Technological Forecasting and Social Change*. 2015. Vol. 95. P. 57–72.

10. Euromonitor International. Global Market Analysis: Flowers and Ornamental Plants. Euromonitor International. URL : https://www.euromonitor.com (дата звернення : 13.05.2025).

11. GfK Ukraine. Дослідження споживчих настроїв на ринку квітів в Україні. GfK Ukraine. URL : https://www.gfk.com (дата звернення : 13.05.2025).