

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ ВАДИМА ГЕТЬМАНА**  
**Навчально-науковий інститут**  
**«Інститут інформаційних технологій в економіці»**  
**Кафедра інформаційних систем в економіці**

**ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»**  
галузь знань 12 «Інформаційні технології»  
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

**КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА**

на тему: «Проектування інформаційної системи для автоматизації процесів  
квіткового магазину»

здобувача Шевченка Олександра Олександровича  
(ПІБ)

\_\_\_\_\_ (Підпис)

**Науковий керівник:**

\_\_\_\_\_ Артемчук В.О.

**Робота допущена до захисту перед  
екзаменаційною комісією з атестації  
здобувачів вищої освіти**

завідувач кафедри:  
к.е.н., доцент.

\_\_\_\_\_ Тішков Б.О.

**Київ 2025**

**Міністерство освіти і науки України**  
**Київський національний економічний університет імені Вадима Гетьмана**  
**Навчально-науковий інститут «Інститут інформаційних технологій в економіці»**

**Кафедра інформаційних систем в економіці**

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»**

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)

освітньо-професійної програми

\_\_\_\_\_ Помазун О.М.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

\_\_\_\_\_ Тішков Б.О.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

здобувача вищої освіти **Шевченку Олександр Олександровичу**

*очної (денної) форми навчання*

**на підготовку кваліфікаційної бакалаврської роботи**

**на тему:** «Проектування інформаційної системи для автоматизації процесів квіткового магазину»

Тему затверджено наказом ректора Університету від «07» березня 2025 р. № 466-ст.

**Кваліфікаційна бакалаврська робота виконується на матеріалах** отриманих під час навчання, аналізу практичних бізнес-процесів інтернет-квіткового магазину і реалізації власного програмного рішення для автоматизації цих процесів.

**План кваліфікаційної бакалаврської роботи**

**Розділ I** Характеристика та аналіз існуючих інформаційних систем для онлайн-торгівлі квітами.

**Розділ II** Розробка вимог до інформаційної системи автоматизації квіткового магазину.

**Розділ III** Проектування та реалізація компонентів інформаційної системи для автоматизації процесів квіткового магазину.

**Об'єкт дослідження:** автоматизації процесів квіткового інтернет магазину, що забезпечує взаємодію клієнтів і адміністраторів із ключовими бізнес-процесами через зручний вебінтерфейс.

**Предмет дослідження:** підходи та технології розробки та проектування інформаційної системи для автоматизації квіткового магазину з використанням фреймворку Django, HTML/CSS, Python і бази даних SQLite.

**Мета кваліфікаційної бакалаврської роботи** – розробка інформаційної системи, яка підвищує ефективність управління товарами й замовленнями, автоматизує обробку замовлень і спрощує взаємодію користувачів із сервісом.

**Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:**

У розділі I (п. 1.1) необхідно окреслити та проаналізувати галузь застосування розроблюваної інформаційної системи (підсистеми, вебзастосунку), виділити в ній основні бізнес-процеси, завдання та проблеми, що потребують автоматизації, а також визначити групи користувачів і зацікавлених осіб, їхні очікування і вимоги до функціональності та характеристик рішення. У пункті 1.2 слід провести систематичний огляд наукових досліджень, публікацій і галузевих матеріалів, присвячених аналогічним системам і технологіям, і на основі цього виділити їх сильні та слабкі сторони.

У розділі II (п. 2.1) потрібно сформулювати набір функціональних та нефункціональних вимог, яким має відповідати розроблювана інформаційна система (підсистема, вебзастосунок) або застосована технологія. У пункті 2.2 слід виокремити й описати ключові задачі, що вирішуються цією системою (підсистемою, застосунком) або технологією, а також розробити алгоритмічні схеми для їх реалізації. У пункті 2.3 необхідно створити відповідні діаграми—структурні, процесні та діаграми взаємодії—для графічного відображення архітектури системи, основних бізнес-процесів і потоків повідомлень між її компонентами.

У розділі III (п. 3.1) слід уточнити та описати види даних, що використовуються в інформаційній системі або технології, вказати їх джерела, методи збору та механізми зберігання й обробки. У пункті 3.2 необхідно окреслити мінімальні та рекомендовані апаратні вимоги для розгортання і стабільної роботи системи. У пункті 3.3 потрібно надати детальний опис обраного технологічного стека. Нарешті, у пункті 3.4 необхідно продемонструвати реальні приклади функціональних можливостей розробленої системи.

---

**Завдання підготував**  
**науковий керівник** \_\_\_\_\_

**Артемчук Володимир Олександрович**

“10” березня 2025 р.

**Завдання одержав**  
**студент** \_\_\_\_\_

**Шевченко Олександр Олександрович**

“10” березня 2025 р.

**Відгук**  
про кваліфікаційну бакалаврську роботу  
здобувача навчально-наукового інституту  
«Інститут інформаційних технологій в економіці»  
освітньо-професійної програми  
«Комп'ютерні науки»

Шевченка Олександра Олександровича

на тему:

**«Проектування інформаційної системи для автоматизації процесів  
квіткового магазину»**

- 1. Актуальність теми:** у сучасних умовах цифровізації торгівлі інформаційні системи у вигляді вебзастосунку для квіткового магазину набувають особливого значення, оскільки дозволяють забезпечити зручний вибір композицій, оперативне оформлення замовлень і своєчасну обробку запитів.
- 2. Позитивні риси кваліфікаційної бакалаврської роботи:** робота структурована відповідно до методичних вимог та складається з трьох логічно взаємопов'язаних розділів. У ній продемонстровано глибоке розуміння предметної області інтернет-торгівлі квітами та ключових бізнес-процесів. Практична частина реалізована за допомогою сучасного стеку технологій-Python, Django, SQLite, Bootstrap.
- 3. Наявність самостійних розробок автора:** представлений інформаційна система створена автором повністю самостійно, включно з розробкою моделі даних, бекенду, шаблонів і механізмів обробки замовлень
- 4. Цінність теоретичних висновків та практичних рекомендацій:** робота містить ґрунтовний аналіз ринку онлайн-квіркової торгівлі, а зроблені висновки підкреслюють переваги автоматизації процесів та покращення досвіду користувачів. Практичні рекомендації щодо подальшого (розвитку інтеграція платіжних шлюзів, аналітичних модулів та адаптивного інтерфейсу) можуть бути безпосередньо реалізовані для підвищення конкурентоспроможності магазину.
- 5. Наявність недоліків:** суттєвих недоліків у роботі не виявлено. Водночас доцільно було б розширити адаптивність дизайну та реалізувати окремий мобільний додаток або адаптивний вебдодаток для ще зручнішого доступу з мобільних пристроїв.
- 6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК:** кваліфікаційна бакалаврська робота відповідає всім вимогам щодо змісту, структури та обсягу. Загалом, розроблена інформаційна система автоматизації квіркового інтернет магазину Шевченка О.О. рекомендується до захисту з високою оцінкою.

Науковий керівник

Артемчук В.О.

\_\_\_\_\_  
(підпис)

“\_\_” червня 2025 р.

## Рецензія

на кваліфікаційну дипломну роботу здобувача вищої освіти  
Шевченка Олександра Олександровича  
(прізвище, ім'я, по батькові)

Тема «Проектування інформаційної системи для автоматизації процесів квіткового магазину»

Актуальність теми кваліфікаційної дипломної роботи і доцільність її розроблення тема дипломної роботи є актуальною в умовах стрімкого розвитку електронної комерції та зростання популярності онлайн-продажів квіткової продукції. Автоматизація бізнес-процесів, таких як управління каталогом, оформлення та відстеження замовлень, дозволяє покращити сервіс, оптимізувати логістику та зменшити витрати. Реалізована система підвищить ефективність роботи квіткового магазину та поліпшить користувацький досвід.

Якість проведеного дослідження дослідження має чітку структуру та логіку викладу. Автор детально проаналізував предметну область, вивчив існуючі рішення для квіткової інтернет-торгівлі та визначив їхні переваги й недоліки. Обґрунтовано вибір технологій (Django, SQLite, Bootstrap), а теоретичні положення вдало поєднані з практичною реалізацією.

Позитивні риси кваліфікаційної дипломної роботи добре спроектована система, яку можна застосувати у будь-якій квітковій мережі, що дозволить значно покращити та спростити взаємодію з покупцями.

Зауваження слід деталізувати опис захисту від можливих помилок при оплаті та обробці виняткових ситуацій (наприклад, нестача товару або проблеми з доставкою).

Практична значимість висновків і рекомендацій Результати роботи мають безпосередню практичну цінність для інтернет-квіткових магазинів, оскільки реалізована інформаційна система сприяє підвищенню оперативності обробки замовлень, покращенню якості обслуговування клієнтів та оптимізації внутрішніх процесів.

### Рецензент

Провідний науковий співробітник  
відділу математичного і економічного моделювання  
Інституту проблем моделювання в енергетиці  
ім. Г.Є. Пухова НАН України  
доктор технічних наук  
старший науковий співробітник



Підпис *Андрій Яцишин*  
Засвідчую  
Начальник відділу кадрів ІПМЕ ім. Г.Є. Пухова  
Національної академії наук України

## АНОТАЦІЯ

кваліфікаційної бакалаврської роботи  
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,  
виконаної на тему: «Проектування інформаційної системи для автоматизації  
процесів квіткового магазину»  
Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена актуальній проблемі автоматизації процесів квіткових магазинів, що пропонується розв'язати шляхом розроблення сучасної інформаційної системи.

Кваліфікаційна робота складається з трьох розділів, логічно взаємопов'язаних між собою.

**У першому розділі** подано характеристику предметної галузі та об'єкта дослідження, здійснено аналіз існуючих рішень автоматизації для квіткових магазинів та визначено проблеми, які потребують вирішення. Проведено огляд літературних джерел щодо сучасного стану розвитку інтернет-торгівлі квітами, тенденцій ринку та вимог до ефективних, безпечних і зручних систем для автоматизації продажів та управління замовленнями.

**Другий розділ** присвячено опису функціональних і технічних вимог до системи, постановці конкретних задач, а також створенню моделей інформаційної системи.

**Третій розділ** – конструктивний. Тут наведено детальний опис інформаційного, технічного і програмного забезпечення реалізованої інформаційної системи. Унікальною особливістю розробленого рішення є зручна система керування каталогом товарів та автоматизація процесів обробки замовлень.

**У висновках** наведено основні результати роботи, практичну цінність створеної інформаційної системи та рекомендації щодо її подальшого впровадження в діяльність квітового магазину.

## РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 63 сторінки, 22 рисунків, 7 таблиць, перелік джерел посилань з 28 найменувань, 3 додатків.

### «Проектування інформаційної системи для автоматизації процесів квіткового магазину»

Об'єкт дослідження: автоматизації процесів квіткового інтернет магазину, що забезпечує взаємодію клієнтів і адміністраторів із ключовими бізнес-процесами через зручний вебінтерфейс.

Предмет дослідження: підходи та технології розробки та проектування інформаційної системи для автоматизації квіткового магазину з використанням фреймворку Django, HTML/CSS, Python і бази даних SQLite.

Мета дослідження: – розробка інформаційної системи для автоматизації ключових бізнес-процесів квіткового магазину, що забезпечить підвищення ефективності роботи магазину, зниження операційних витрат, автоматизацію обробки замовлень і покращення якості обслуговування клієнтів незалежно від їхнього місця перебування.

Відповідно до поставленої мети визначені такі завдання:

1. Дослідити сферу автоматизації квіткових магазинів, виявити переваги і недоліки існуючих рішень.
2. Розглянути коло потенційних користувачів і визначити базові очікування до майбутньої інформаційної системи.
3. Визначити вимоги до функціоналу, безпеки, зручності та масштабованості системи, що розробляється.
4. Побудувати інформаційну модель, описати алгоритми роботи з каталогом, замовленнями та логістикою, розробити базу даних.
5. Реалізувати вебзастосунок із використанням сучасних технологій та фреймворків, протестувати систему на відповідність визначеним вимогам.
6. Оцінити ефективність роботи системи, зокрема швидкість обробки замовлень, зручність користування, стабільність роботи та можливість подальшого розширення.

Теоретична, методична та практична значущість отриманих результатів полягає в удосконаленні підходів до автоматизації процесів квіткових магазинів через реалізацію зручної та гнучкої інформаційної системи з можливістю автоматичної обробки замовлень і адаптації до різних умов роботи. Система дозволяє автоматизувати процеси оформлення, обліку та доставки квіткової продукції, підвищуючи ефективність діяльності магазину. Отримані результати можуть бути використані у квіткових магазинах будь-якого масштабу, мережах магазинів, а також приватними підприємцями для покращення обслуговування клієнтів. Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: автоматизація, квітковий магазин, інформаційна система, вебдодаток, бази даних, Python, Django, розробка програмного забезпечення.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	3
ВСТУП.....	4
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	2
1.1 Характеристика предметної області та об'єкта дослідження.....	2
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	7
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	13
2.1 Аналіз і специфікація вимог до інформаційної системи .....	13
2.2 Постановка та алгоритм розв'язання задачі.....	20
2.2.1 Постановка задачі .....	20
2.2.2. Алгоритм розв'язання задачі .....	24
2.3 Моделювання інформаційної системи.....	28
2.3.2 Моделювання структури системи .....	32
2.3.3 Розподіл вимог за компонентами системи.....	35
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ ...	38
3.1 Інформаційне забезпечення .....	38
3.2 Технічне забезпечення.....	40
3.3 Реалізація основних функцій інформаційної системи .....	42
3.4 Результати реалізації інформаційної системи.....	47
ВИСНОВКИ.....	56
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	61

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface

CRUD – Create, Read, Update, Delete

CSRF – Cross-Site Request Forgery

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IT – Інформаційні технології

MVC – Model-View-Controller

ORM – Object-Relational Mapping

REST – Representational State Transfer

SQL – Structured Query Language

UI – User Interface

UX – User Experience

БД – База даних

ІЗ – Інформаційне забезпечення

ІС – Інформаційна система

ПЗ – Програмне забезпечення

СУБД – Система управління базами даних

## ВСТУП

У сучасному світі, де цифрові технології стрімко інтегруються в усі сфери бізнесу, онлайн-торгівля квітами та подарунками перетворилася на невід’ємну частину роздрібного ринку. Традиційні квіткові салони дедалі частіше доповнюють або навіть замінюють інформаційними системами, що забезпечують зручний віддалений доступ до каталогу букетів і швидке оформлення замовлень. Завдяки вебдодаткам клієнти можуть переглянути асортимент у будь-яку годину, порівняти ціни, обрати спосіб доставки та оплатити покупку онлайн — усе це мінімізує час на відвідування фізичних магазинів і розширює аудиторію за межі локального району.

Особливу актуальність розвиток таких систем набуває для малого та середнього бізнесу, який не завжди має ресурси на складні ERP-рішення. Локальні флористичні майстерні в Україні потребують гнучких, недорогих, але водночас надійних інструментів автоматизації — від створення та впорядкування товарного каталогу до обробки замовлень і контролю залишків на складі. У цьому контексті розробка вебзастосунку для квітового магазину не лише відповідає світовим трендам e-commerce, а й вирішує конкретні виклики вітчизняного ринку: нерівномірність клієнтського потоку, обмеженість технічних ресурсів та необхідність швидкого масштабування в періоди святкових підйомів попиту.

Результатом дипломної роботи став вебдодаток на Django з каталогом букетів і базовою адмін-панеллю для завантаження фото та редагування описів. Проте практичні випробування показали потребу в суттєвому розширенні функціоналу:

- введення авторизації та розмежування прав доступу між адміністраторами та зареєстрованими клієнтами;
- налаштування кошика й обробки замовлень, що автоматично підраховує загальну вартість із урахуванням кількості товарів;
- додавання форми оформлення доставки з фіксацією контактних даних і генерацією унікального номеру замовлення;

- покращення інтерфейсу та захисту даних, щоб клієнти могли безпечно здійснювати оплату і відстежувати статус своїх заявок.

Аналіз наукової та галузевої літератури свідчить, що питання проєктування подібних систем охоплюють такі ключові аспекти:

1. **Архітектура вебдодатків.** MVC-підхід, REST-API, клієнт-серверна взаємодія та мультисервісні рішення.
2. **Моделювання баз даних.** Нормалізація, індексація, оптимізація запитів, зберігання мультимедійних файлів.
3. **Безпека та аутентифікація.** Стандарти JWT, OAuth2, захист від CSRF/SQL-ін'єкцій, шифрування даних.
4. **UX/UI-дизайн.** Адаптивність під різні пристрої, інтуїтивність навігації, мінімізація кроків на шляху користувача до платного вікна.

У вітчизняному експертному середовищі (Сидоренко О., Петренко І.) відзначають необхідність створювати рішення, що поєднують простоту впровадження з високим рівнем безпеки, а зарубіжні дослідження (Smith J., Lee A.) підкреслюють важливість масштабованості при пікових навантаженнях, особливо під час святкових кампаній. Окрім того, модель «користувач — кошик — замовлення» широко описана в роботах з електронної комерції, але реалізація для нішевого ринку квітів потребує урахування специфіки сезонності товарів та логістичних особливостей доставки живих рослин.

**Мета дослідження** полягає в комплексному створенні інформаційної системи для квіткового магазину, яка забезпечить:

- формування товарного каталогу з можливістю фільтрації за категоріями та ціною;
- безпечну реєстрацію й авторизацію користувачів;
- функціонал «кошика» з автоматичним обчисленням вартості та зберіганням кількості товарів;
- інтерфейс оформлення замовлення з введенням контактних даних клієнта та вибором способу доставки;

- адмін-панель для керування асортиментом та перегляду статистики замовлень.

**Для досягнення цієї мети** необхідно розв'язати такі завдання:

1. Дослідити сферу автоматизації квіткових магазинів, виявити переваги і недоліки існуючих рішень.
2. Розглянути коло потенційних користувачів і визначити базові очікування до майбутньої інформаційної системи.
3. Визначити вимоги до функціоналу, безпеки, зручності та масштабованості системи, що розробляється.
4. Побудувати інформаційну модель, описати алгоритми роботи з каталогом, замовленнями та логістикою, розробити базу даних.
5. Реалізувати вебзастосунок із використанням сучасних технологій та фреймворків, протестувати систему на відповідність визначеним вимогам.
6. Оцінити ефективність роботи системи, зокрема швидкість обробки замовлень, зручність користування, стабільність роботи та можливість подальшого розширення.

**Об'єкт дослідження** – автоматизації процесів квітового інтернет магазину, що забезпечує взаємодію клієнтів і адміністраторів із ключовими бізнес-процесами через зручний вебінтерфейс.

**Предмет дослідження** – підходи та технології розробки та проектування інформаційної системи для автоматизації квітового магазину з використанням фреймворку Django, HTML/CSS, Python і бази даних SQLite

**Методи дослідження** включають аналіз літератури, моделювання даних, проектування архітектури, розробку та тестування коду, а також опитування користувачів для оцінки зручності інтерфейсу.

**Теоретична та практична значущість** полягає в тому, що результати дипломної роботи можуть бути застосовані для швидкого запуску та розвитку невеликих флористичних бізнесів, покращення клієнтського сервісу та оптимізації внутрішніх процесів обробки замовлень. Запропонована система є готовим

«коробковим» рішенням, яке легко масштабувати і доповнювати новими функціями — від інтеграції електронних платежів до аналітики продажів.

**Інформаційна база** включає документацію Django та Bootstrap, наукові статті з e-commerce, методичні рекомендації щодо створення вебзастосунків, а також актуальні онлайн-джерела (офіційні блоги розробників, форуми та фахові портали).

Таким чином, дипломна робота продовжує і поглиблює попередні напрацювання, спрямовуючи їх на створення повноцінного сервісу для онлайн-торгівлі квітами, що відповідає сучасним вимогам ринку та потребам кінцевих користувачів.

**Структура роботи.** Загальний обсяг роботи становить 63 сторінки друкованого тексту, 22 рисунків, 7 таблиць, 3 додатків. Список використаних джерел налічує 28 найменування.

## РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Характеристика предметної області та об'єкта дослідження

Предметна область даного дослідження охоплює весь спектр бізнес-процесів інтернет-квіткового магазину, які можуть бути автоматизовані за допомогою вебінформаційної системи. У межах цього дослідження виділено чотири центральних напрями:

#### 1. Управління каталогом букетів

- **Перегляд** — користувачі (гості та зареєстровані) можуть ознайомитися зі структурованим каталогом, що містить зображення, назви, опис та ціну кожного букета.
- **Фільтрація та сортування** — реалізовано фільтри за кольором квітів, типом (тюльпани, троянди, мікси), ціною та сезонними акціями.
- **Перевірка наявності** — перед оформленням замовлення система у реальному часі консультується з базою даних про залишки на складі і відображає статус “в наявності” або “немає в наявності”.

#### 2. Механізм “кошика” та багатокрокове оформлення замовлення

- **Додавання в кошик** — інтуїтивна кнопка на сторінці товару дозволяє швидко додати обраний букет до віртуального кошика.
- **Перегляд та редагування кошика** — користувачі можуть змінювати кількість товарів, видаляти позиції або переходити до оформлення.
- **Багатокрокова форма** — процес оформлення розбитий на кроки: введення контактних даних, вибір способу доставки та оплати, підтвердження замовлення. На кожному кроці система проводить валідацію вводу та підраховує остаточну суму з урахуванням можливих знижок і тарифів доставки.

#### 3. Реєстрація та аутентифікація клієнтів і адміністраторів

- **Гостьовий режим** — незареєстровані відвідувачі можуть переглядати каталог і додавати товари в кошик, але для завершення покупки їм пропонується зареєструватися або увійти.
- **Аутентифікація** — захищений механізм входу на основі Django-Auth, із підтримкою відновлення пароля, підтвердження електронної пошти та авторизації через токени.
- **Ролі користувачів** — у системі передбачено дві основні ролі: **Клієнт** (може оформлювати замовлення, переглядати історію покупок) та **Адміністратор** (має доступ до панелі управління товарами, замовленнями та користувачами).

#### 4. Адміністрування облікових записів та записів у базі даних

- **CRUD-операції з товарами** — адміністратор може додавати нові букети, редагувати їхні характеристики (склад, ціну, зображення) та видаляти застарілі позиції.
- **Управління акціями** — створення тимчасових знижок, промокодів та спецпропозицій із зазначенням дати початку та завершення дії.
- **Моніторинг та звітність** — генерація звітів по продажах, перегляд аналітики відвідуваності каталогу та конверсії в замовлення.

**Об'єктом дослідження** є Інформаційна система автоматизації процесів квіткового магазину — комплекс програмних та апаратних засобів, що реалізують перелічені вище функції через вебінтерфейс. Ця система є багатокористувацькою та передбачає гнучке розмежування прав доступу, інтеграцію з платіжними шлюзами і службами доставки, а також забезпечення надійності та захищеності даних.

#### **Структурна схема предметної області**

Загальну архітектуру ІС (системна діаграма верхнього рівня) подано на рис. 1.1, вона поділена на три ключові підсистеми:

## 1. Авторизація користувача

- Незареєстровані відвідувачі можуть перейти на сторінку авторизації або реєстрації.
- Після успішного входу — доступ до особистого кабінету (клієнта або адміністратора).

## 2. Оформлення замовлення квітів

- Маршрут користувача: Головна сторінка → Каталог букетів → Сторінка товару → Додавання в кошик → Оформлення замовлення → Сторінка оплати → Сторінка підтвердження успішного замовлення.

## 3. Адміністраторська панель

- Перевірка наявності товарів → Керування асортиментом → Адміністрування акаунтів користувачів та системних налаштувань.

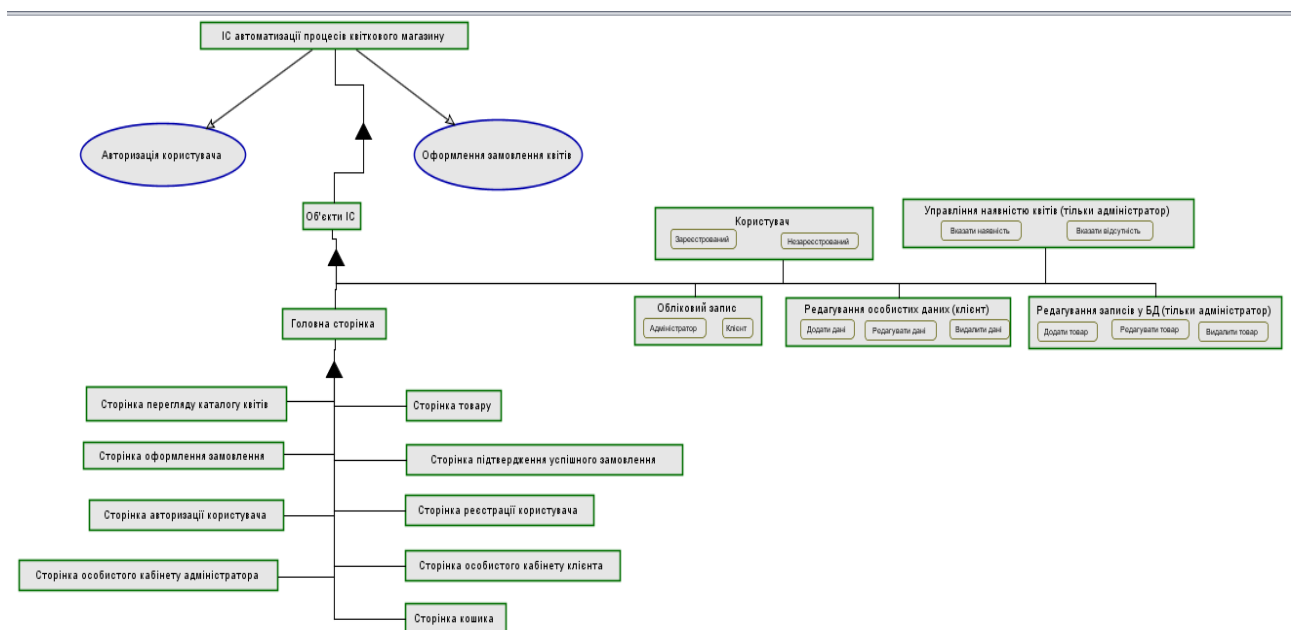


Рисунок 1.1 – Системна діаграма верхнього рівня

Джерело: сформовано автором самостіно

Подальша деталізація показана на рис. 1.2–1.4:

Кроки підсистеми “Оформлення замовлення”: від вибору букета до оплати та підтвердження зображено на рис. 1.2.

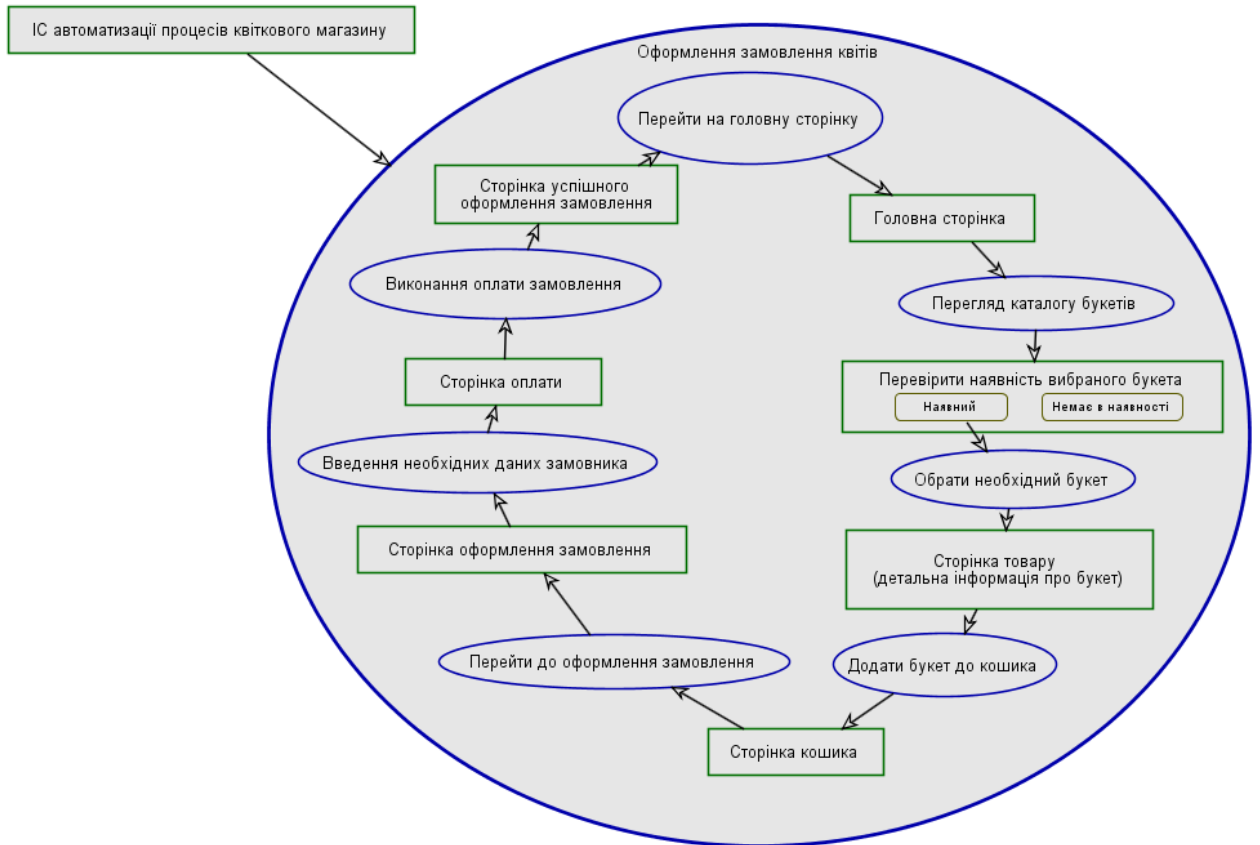


Рисунок 1.2 – Діаграма Оформлення замовлення

Джерело: сформовано автором самостіно

Ілюстрація послідовності дій у підсистемі “Авторизація”: від переходу на сторінку входу чи реєстрації до завантаження персонального кабінету з відповідними правами зображено на рис. 1.3.

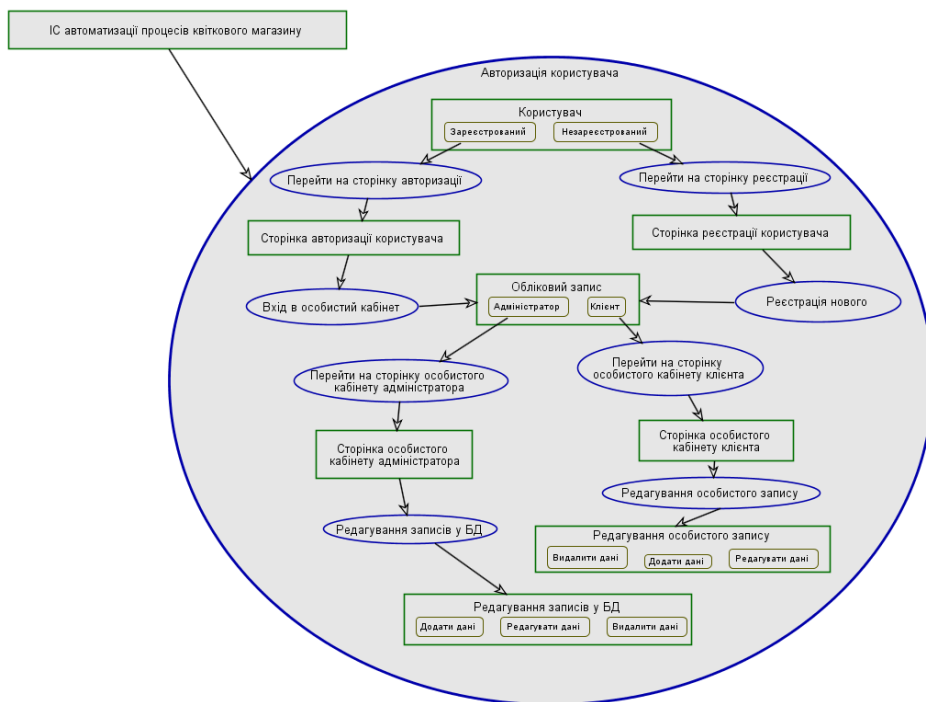


Рисунок 1.3 – Діаграма авторизація користувача

Джерело: сформовано автором самостіно

Узагальнююча ієрархія всіх об'єктів ІС, будучи своєрідним "мапою" функціональних модулів і сторінок зображена на рис. 1.4.



Рисунок 1.4 – Узагальнююча ієрархія всіх об'єктів

Джерело: сформовано автором самостіно

## 1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

У сучасних умовах стрімкої діджиталізації торгівлі та глобальних економічних викликів інформаційні системи електронної комерції, зокрема для онлайн-продажу квітів, набувають критичного значення. Впровадження таких платформ дозволяє автоматизувати оформлення замовлень, знизити навантаження на персонал магазину та забезпечити оперативний зворотний зв'язок із клієнтом [3, 4].

Вітчизняні дослідження наголошують на необхідності створення адаптивних, гнучких і безпечних вебсервісів, що враховують особливості ринку квітів та рекомендації профільних органів. У науково-практичній літературі приділяють увагу питанням структурування каталогу букетів, оптимізації інтерфейсу користувача, інтеграції з платіжними шлюзами та системами управління складом [5]. Зокрема, Іващенко О. і Ковальчук М. підкреслюють, що інтуїтивна навігація і спрощена процедура оформлення замовлення значно підвищують конверсію та рівень задоволеності покупців, а також зменшують кількість «кинутих» кошиків [3].

Зарубіжні джерела виділяють ключові технічні виклики: забезпечення масштабованості платформи під час пікових навантажень (святкові періоди, акції), стабільності роботи у мобільних та десктопних середовищах, а також захисту від кібератак. У дослідженні Сміта та співавт. (2023) аналізуються проблеми обробки тисяч паралельних замовлень та оптимізації кешування для прискорення завантаження сторінок [7]. Аналітичні звіти Gartner (2025) виділяють основні загрози для e-commerce-систем — DDoS-атаки, збої платіжних шлюзів, вразливості в API — і пропонують комплексні заходи: резервне копіювання, шифрування даних і багаторівневу систему аутентифікації користувачів [6].

Таким чином, аналіз літератури й практичного досвіду показує, що успішна інформаційна система для онлайн-квіткового магазину має поєднувати:

- зручний користувацький інтерфейс з адаптивним дизайном;

- інтеграцію з платіжними й кур'єрськими сервісами;
- масштабовану архітектуру, що витримує пікові навантаження;
- безпеку даних і захист від зовнішніх атак;
- аналітичні інструменти для відстеження поведінки покупців і оптимізації бізнес-процесів.

Саме такі вимоги лягли в основу розробленої в цій роботі платформи для автоматизації електронної торгівлі квітами.

**Дослідження існуючих інформаційних систем для обраної предметної області.** У сучасному сегменті української інтернет-торгівлі квітами можна виділити три яскраві приклади платформ, що суттєво відрізняються за підходом до продажу, набором функцій і користувацьким досвідом:

### **UkraFlora**

Тип рішення: корпоративний сайт-каталог із B2B/B2C-орієнтацією

Каталог і фільтрація: великий асортимент зрізаних квітів і кімнатних рослин, але без AJAX-фільтрів — товари відображаються списком, фільтрація лише базова (по категоріях).

Кошик і оформлення: відсутній власний «кошик» — оформлення через форму замовлення або контакт-центр (телефон, e-mail).

Оплата: переважно передоплата на реквізити або оплата при отриманні готівкою/терміналом кур'єра.

Доставка: внутрішня служба доставки компанії, гнучкі умови; розрахунок строків і вартості по регіонах.

Переваги:

- Надійний постачальник із 1993 р. — довіра великого числа корпоративних клієнтів.
- Широкий асортимент (понад 100 зрізаних і 200+ кімнатних рослин).

Недоліки:

- Відсутність швидкого онлайн-чеку-аута і інтегрованого кошика.

- Мінімально інтерактивний UX — користувачу доводиться дзвонити або писати для замовлення.

### **Dicentra**

Тип рішення: повнофункціональний B2C-інтернет-магазин на базі власної платформи

Каталог і фільтрація: AJAX-фільтри за кольорами, видами квітів, ціною, тематикою (весілля, День вчителя тощо).

Кошик і оформлення: багатокроковий checkout; можна змінювати кількість товарів, вибирати дату/час доставки.

Оплата: карткою онлайн, Apple Pay/Google Pay, терміналом кур'єра.

Доставка: кур'єрською службою по місту й області; трекінг-повідомлення SMS/Viber.

Переваги:

- Інтуїтивно зрозумілий інтерфейс, адаптивний дизайн для мобільних.
- Повний цикл онлайн-замовлення без виходу з сайту.

Недоліки:

- Щомісячні витрати на підтримку та оновлення (хоча ціни конкурентні).
- Обмежена кастомізація UX для особливих акцій чи промо.

### **Tea-Rose**

Тип рішення: «вітрина» — статичний онлайн-каталог із формою зворотного зв'язку

Каталог і фільтрація: товари розбиті на категорії, без динамічних фільтрів; пошук обмежений.

Кошик і оформлення: власного кошика немає — замовлення формується через кнопку «Замовити» → форма заявки або чат у Telegram.

Оплата: уточнюється під час дзвінка/в листуванні (готівка, переказ на карту).

Доставка: кур'єром за домовленістю; інформація про терміни й тарифи виводиться у вигляді текстового блоку.

Переваги:

- Мінімальні витрати на розробку та підтримку.
- Швидкий запуск «вітрини» під будь-які акції.

Недоліки:

- Користувачі відволікаються на зовнішні канали (чат, телефон).
- Відсутність аналітики та автоматизації процесу продажу.

Таблиця 1.1 – Поорівняння платформ для інтернет-квіткового магазину

Характеристика	UkraFlora	Dicentra	Tea-Rose
Тип рішення	Корпоративний каталог	Повнофункціональний магазин	Статична вітрина
Каталог/фільтрація	Базові категорії	AJAX-фільтри (колір, тематика)	Категорії без фільтрів
Кошик/checkout	Ні	Так, багато-кроковий	Ні, лише форма заявки
Оплата	Передоплата/готівка	Онлайн-картка, Apple/Google Pay	За домовленістю
Доставка	Власна служба	Кур'єр + трекінг SMS/Viber	За домовленістю
Аналітика/UX	Мінімальна	Повна інтеграція аналітики	Відсутня
Швидкість запуску	Середня (налагодження B2B)	Швидкий (готова платформа)	Дуже швидкий (вітрина)
Вартість підтримки	Середня–висока (B2B продукція)	Середня (хостинг + підтримка)	Низька

*Джерело: сформовано автором самостіно*

Після детального аналізу трьох українських платформ для онлайн-торгівлі квітами — **UkraFlora**, **Dicentra** та **Tea-Rose** — можна виокремити їхні ключові переваги та недоліки у контексті потреб інтернет-квіткового магазину.

**UkraFlora** — корпоративний сайт-каталог із широким асортиментом зрізаних квітів і кімнатних рослин, але без інтегрованого кошика чи швидкого чеку-аута.

Основні плюси:

- Надійність постачання (працює з 1993 року), довіра B2B-клієнтів.
- Дуже великий асортимент.

Недоліки:

- Відвідувачам доводиться оформляти замовлення через форму чи дзвінок, а не онлайн-кошик.
- Мінімальна інтерактивність і відсутність аналітики користувацької активності.

**Dicentra** — повнофункціональний B2C-інтернет-магазин із гнучким AJAX-фільтром, багатокроковим оформленням, оплатою картою та трекінгом доставки.

Переваги:

- Інтуїтивний UX і responsive-дизайн.
- Повністю онлайн-чеку-аут із вибором дати/години доставки.

Недоліки:

- Щомісячна плата за підтримку та обмежена кастомізація бекенду.
- Використання готових шаблонів може не повністю відповідати унікальному стилю магазину.

**Tea-Rose** — статичний «вітринний» каталог із формою заявки або чатом замість кошика.

Плюси:

- Швидке «виведення вітрини» і мінімальні витрати на розробку.
- Легкість у підтримці — немає складної логіки продажів.

Недоліки:

- Відсутність повноцінного онлайн-замовлення, аналітики продажів і автоматизації.
- Клієнт змушений переходити в зовнішні канали (чат, телефон), що знижує конверсію.

Усі три системи мають вагомні переваги, але жодна не поєднує:

1. Миттєвого односторінкового чекуаута (як у Dicentra).
2. Низьких витрат на впровадження (як у Tea-Rose).
3. Поглибленої аналітики й B2B-функцій (як у UkraFlora).

Отже, існує потреба в розробці власної інформаційної системи автоматизації онлайн-квіткового магазину, яка б об'єднала:

- Односторінковий checkout із можливістю швидкого замовлення «в один клік»;
- Мінімальні витрати на старт і підтримку за рахунок використання open-source компонентів;
- Вбудовану аналітику та персоналізовані рекомендації, щоб підвищити середній чек і повторні продажі;
- Гнучкий каталог з AJAX-фільтрами, що легко налаштовується для різних акцій;
- Інтегровані платіжні шлюзи (Visa/Mastercard, Google/Apple Pay) та кур'єрські сервіси з трекінгом.

Розробка такої системи дозволить поєднати швидкість запуску, економічність і потужність аналітичного модуля, забезпечивши конкурентні переваги на ринку інтернет-квіткової торгівлі.

## РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Аналіз і специфікація вимог до інформаційної системи

#### Бізнес-вимоги

Інтернет-квітковий магазин — це комплексна комерційна платформа, успіх якої зумовлений узгодженістю інтересів усіх зацікавлених сторін (стейкхолдерів).

#### Внутрішні стейкхолдери

##### Розробники

- Відповідають за створення масштабованої, надійної та безпечної системи, здатної обробляти велику кількість одночасних запитів (перегляд каталогу, створення замовлень).
- Потребують чіткого технічного завдання, стабільного фінансування та ресурсів для довготривалого супроводу, оновлень і усунення вразливостей.

##### Керівництво компанії

- Розглядає ІС як інструмент росту доходів: розширення клієнтської бази, оптимізація витрат на обробку замовлень та логістику, зменшення участі оператора в рутинних процесах.
- Очікує мінімізації витрат на підтримку системи й швидкого повернення інвестицій.

##### Маркетинг і PR

- Формують привабливий образ магазину в мережі, контролюють комунікацію в соціальних мережах і розсилках.
- Відповідають за лояльність клієнтів, впровадження акційних пропозицій і збір зворотного зв'язку.

## Зовнішні стейкхолдери

### Клієнти (покупці)

- Хочуть швидко знайти потрібний букет, зручно відфільтрувати асортимент, переконатися у наявності та оперативно оформити доставку.
- Очікують захисту персональних даних, прозорих умов оплати й можливості відстежувати статус замовлення в режимі онлайн.

### Партнери з доставки

- Потребують інтеграції з API магазину для автоматичного отримання даних про адреси та часові слоти доставки.

### Регуляторні органи та платіжні шлюзи

- Вимагають дотримання стандартів електронної комерції, захисту даних клієнтів (GDPR-подібні норми), безпечних платежів (PCI DSS) та конфіденційності.

Щоб забезпечити баланс інтересів і досягти поставлених бізнес-цілей, система автоматизації інтернет-квіткового магазину повинна виконувати такі ключові вимоги:

1. Надійна реєстрація та аутентифікація користувачів із захистом від несанкціонованого доступу.
2. Зручний каталог із фільтрами та швидким пошуком, з підтримкою високої навантаженості.
3. Інтуїтивно зрозумілий процес оформлення замовлення з вибором способу оплати та доставки.
4. Прозоре відстеження статусу замовлення клієнтом і адміністративний екран для моніторингу та редагування замовлень.
5. Інтеграція з партнерами з логістики та платіжними провайдерами через надійні API.
6. Інструменти маркетингу (акції, купони, email-розсилки) для підвищення конверсії та утримання клієнтів.

Діаграма бізнес-вимог і бізнес-цілей внутрішніх та зовнішніх стейкхолдерів інформаційної системи автоматизації інтернет-квіткового магазину показана на рис. 2.1.

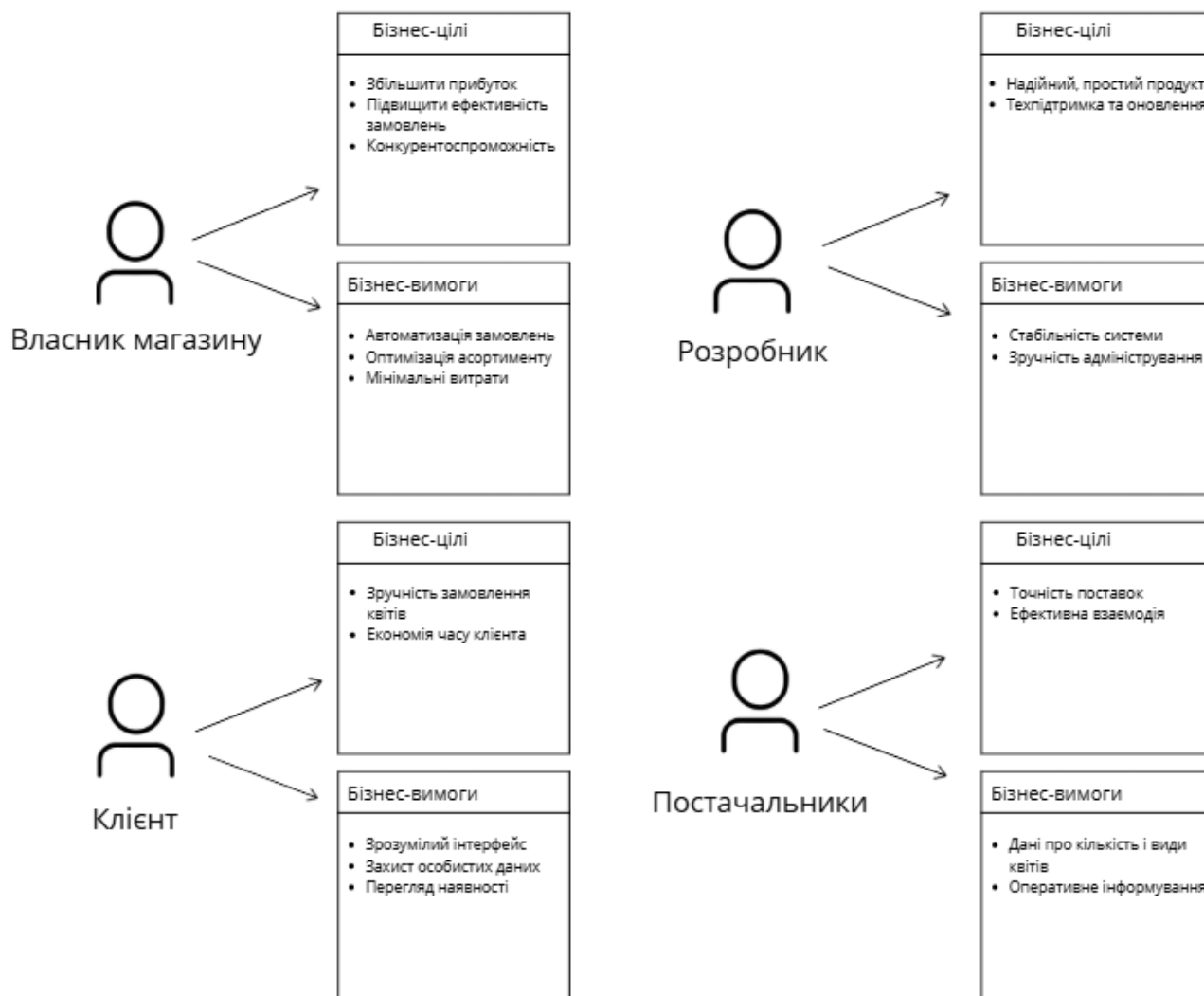


Рисунок 2.1 – Діаграма бізнес-вимог

Джерело: сформовано автором самостіно

**Функціональні вимоги.** Функціональні вимоги описують внутрішню логіку роботи інформаційної системи, зокрема обробку та маніпуляцію даними, обчислення і дії, які вона має виконувати для задоволення потреб кінцевих користувачів і адміністраторів [11].

Функціональні вимоги до вебзастосунку інтернет-квіткового магазину визначають ключові операції та сервіси, які система повинна надавати:

Головними підсистемами є:

#### Автентифікація та профіль користувача

- забезпечує реєстрацію нових клієнтів і вхід у систему;
- дозволяє редагувати персональні дані (ПІБ, адреса, телефон, email);
- розмежовує права доступу: клієнт — лише до оформлення та перегляду замовлень, адміністратор — до службової частини (додавання/редагування товарів, перегляд усіх замовлень).

#### Каталог товарів і пошук/фільтрація

- відображає асортимент букетів із ціною, описом і фото;
- підтримує багатовимірну фільтрацію (за типом квітів, кольором, ціновим діапазоном тощо) і пошук за назвою;
- забезпечує швидке завантаження та пагінацію при великій кількості товарів.

#### Кошик та оформлення замовлення

- дає змогу клієнту додавати/видаляти позиції, змінювати кількість товарів;
- обчислює загальну вартість із урахуванням доставки;
- реалізовує багатокрокове оформлення: введення контактних даних, вибір способу доставки/оплати, підтвердження замовлення;
- інтегрується з платіжним шлюзом для безпечної оплати карткою або через онлайн-банкінг.

#### Адміністрування

- CRUD-операції з каталогом товарів (назва, опис, фото, ціна, кількість на складі);
- перегляд, фільтрація та зміна статусів замовлень (нове, обробляється, доставлено, скасовано);
- управління користувацькими акаунтами (призначення/зміна ролей, блокування).

#### Моніторинг та звітність

- генерація звітів про продажі за період, популярні позиції, середній чек;
- відображення статистики відвідувань і конверсії в адмін-панелі;
- експорт даних у CSV/XLS для подальшого аналізу.

### Маркетингові інструменти

- створення акцій та купонів із відсотковою або фіксованою знижкою;
- відправка email-розсилок зареєстрованим користувачам про новинки й акції;
- рекомендаційний модуль «Супутні товари» на сторінці картки букета.

Ці підсистеми забезпечують повноцінний функціонал, необхідний для комфортного користування магазином, ефективного адміністрування асортименту та аналітики продажів.

Діаграма функціональних вимог інформаційної системи автоматизації інтернет-квіткового магазину показана на рис. 2.2 а специфікації вимог, виконані у середовищі Enterprise Architect, наведено в «Додатку Б» (рис. Б.1–Б.4).

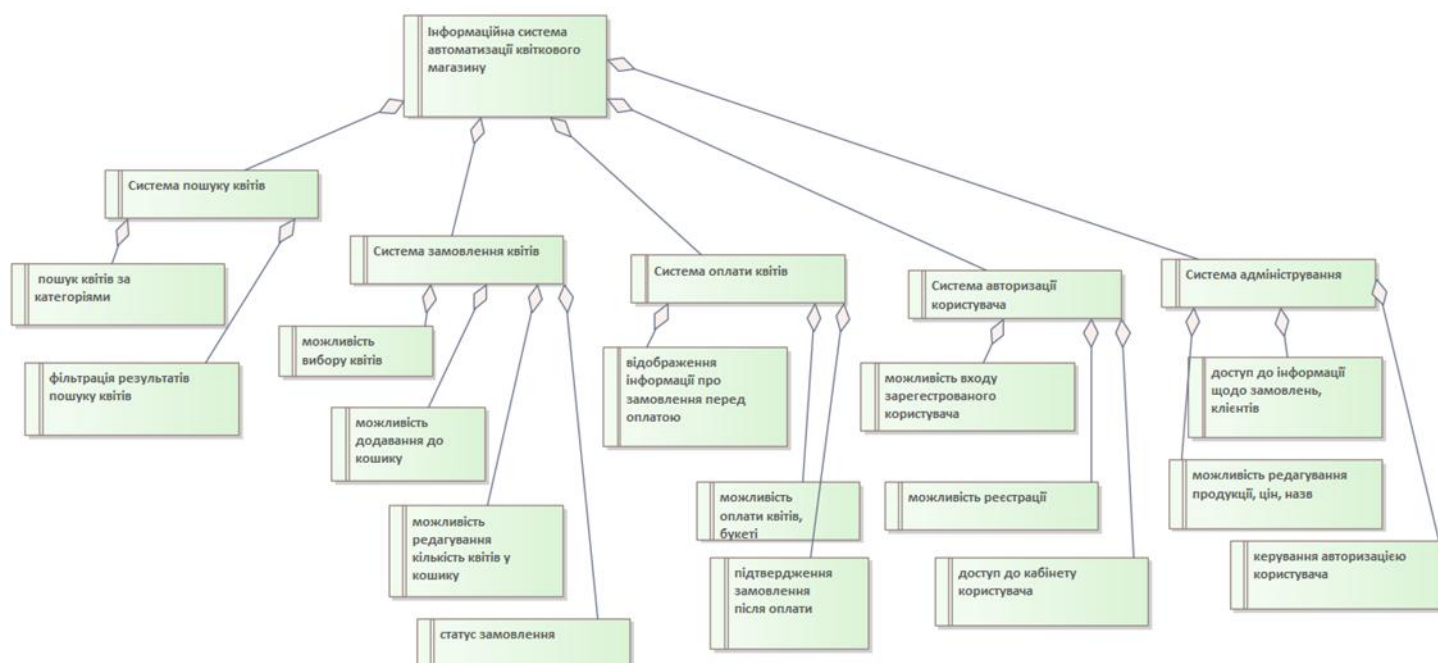


Рисунок 2.2 – Діаграма функціональних вимог

Джерело: сформовано автором самостіно

**Нефункціональні вимоги.** Нефункціональні вимоги визначають загальні критерії якості роботи інформаційної системи онлайн-квіткового магазину — як саме система повинна працювати, а не що вона робить [12].

#### Зручність і доступність інтерфейсу

- Адаптивний дизайн: коректне відображення на десктопах, планшетах і смартфонах;
- Інтуїтивна навігація: зрозумілий головний меню, чітка організація каталогу і кошика;
- Швидке завантаження сторінок: час відгуку інтерфейсу не більше 200 мс;
- Підтримка мов локалізації: українська та англійська мови інтерфейсу.

#### Продуктивність і масштабованість

- Час відповіді сервера: не більше 300 мс для запитів до каталогу та кошика;
- Швидкість обробки замовлень: оформлення і збереження замовлення — не більше 1 секунди;
- Паралельність: підтримка мінімум 500 одночасних активних сесій без деградації продуктивності;
- Горизонтальне масштабування: можливість додавання нових вебсерверів і балансувальників навантаження.

#### Надійність та стійкість до відмов

- Доступність: 99,9 % часу без простоїв;
- Резервне копіювання: щоденне бекапування БД і файлових медіа;
- Відновлення після збою: RTO (Recovery Time Objective) не більше 2 годин;
- Моніторинг і сповіщення: автоматичне відстеження помилок і відправка повідомлень адміністратору.

#### Безпека

- Захищені з'єднання: усі запити через HTTPS/TLS;

- Шифрування конфіденційних даних: паролі зберігаються з використанням bcrypt;
- Розмежування прав доступу: ролі «клієнт» і «адміністратор» із суворим контролем ресурсів;
- Захист від атак: CSRF-токени, валідація вводів, захист від SQL-ін'єкцій та XSS.

#### Сумісність і переносимість

- Браузери: підтримка сучасних версій Chrome, Firefox, Edge, Safari;
- ОС: коректна робота на Windows, macOS, Linux, iOS і Android;
- Інтеграція: відкритий REST API для зв'язку з платіжними шлюзами, CRM та службами доставки.

#### Супровідність та розширюваність

- Модульність коду: чітка архітектура з розмежуванням відповідальностей (MVC/CBV, сервіси);
- Документованість: автогенерована документація API (Swagger/OpenAPI);
- Тестування: покриття юніт- і інтеграційними тестами не менше 80 %;
- Легкість впровадження нових фіч: використання системи контролю версій та CI/CD.

#### Юридична та нормативна відповідність

- GDPR-сумісність: можливість експорту та видалення персональних даних користувача;
- Законодавство України: відповідність правилам електронної торгівлі та захисту прав споживачів.

Нефункціональні вимоги до інформаційної системи автоматизації інтернет-квіткового магазину зображені на рис. 2.3. Специфікації нефункціональних вимог, виконані в Enterprise Architect, наведено в «Додатку Б» (рис. Б.5–Б.9).

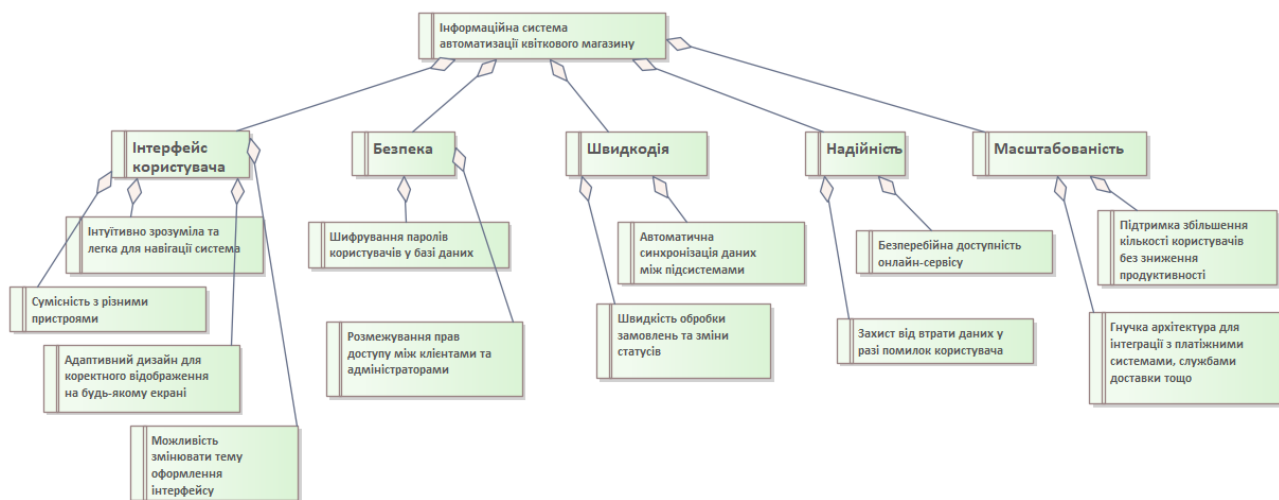


Рисунок 2.3 – Діаграма нефункціональних вимог

Джерело: сформовано автором самостіно

## 2.2 Постановка та алгоритм розв'язання задачі

### 2.2.1 Постановка задачі

**Характеристика задачі.** Метою є розробка інтуїтивно зрозумілої та простої інформаційної системи автоматизації інтернет-квіткового магазину. Система повинна забезпечувати дві групи користувачів:

Клієнти:

- реєстрація та вхід у власний кабінет;
- перегляд каталогу букетів із фільтрацією й пошуком;
- додавання товарів до кошика та керування його вмістом;
- багатокрокове оформлення замовлення (введення контактних даних, вибір способу оплати та підтвердження покупки);
- отримання інформації про статус замовлення та номер для відстеження.

Адміністратори:

- управління каталогом (додавання/редагування/видалення букетів);
- перегляд і обробка замовлень (зміна статусу, формування звітів);
- керування користувачами (перегляд реєстрацій, блокування, зміна ролей);

- моніторинг активності системи та конфігурація налаштувань доставки й оплати.

Інформаційну модель ІС автоматизації квіткового магазину, яка демонструє основні сутності (Клієнт, Адміністратор, Букет, Кошик, Замовлення) та їхні зв'язки з базою даних. Взаємодія відбувається через вебінтерфейс: клієнти формують запити до каталогу та кошика, а адміністратори — до розділу управління товарами й замовленнями наведено на рисунку 2.4.

Такий розподіл функцій гарантує зручність використання для покупців і контроль за бізнес-процесами з боку адміністрації, а чітка модель даних слугує основою для подальшої розробки архітектури системи та алгоритмів обробки запитів.

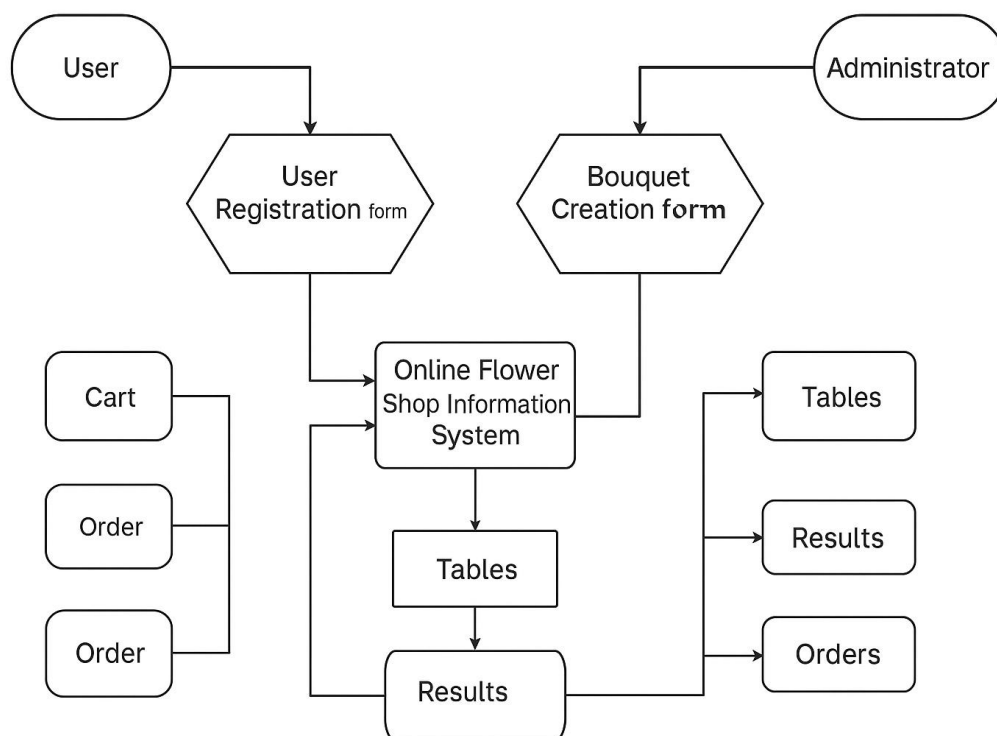


Рисунок 2.4 – Інформаційна модель ІС

*Джерело: сформовано автором самостіно*

Вихідна інформація. У контексті інформаційної системи інтернет-квіткового магазину вихідна інформація — це сукупність повідомлень і звітів, які система генерує у відповідь на дії користувачів та адміністраторів. Вона забезпечує підтвердження коректного виконання операцій, інформує про помилки або

недоступність ресурсів, а також надсилає необхідні дані для подальшої роботи з системою.

Для клієнтів:

- Підтвердження оформлення замовлення (номер, сума, очікуваний термін доставки).
- Повідомлення про помилки при оплаті (недостатньо коштів, прострочена картка тощо).
- Сповіщення про оновлення профілю (успішна зміна адреси доставки, телефону та ін.).
- Повідомлення про наявність/відсутність товару при додаванні в кошик.
- Статус відвантаження (відправлено, у дорозі, доставлено).

Для адміністраторів:

- Нове замовлення (номер замовлення, ім'я клієнта, перелік позицій).
- Стан оплати (успішна/провалена транзакція).
- Низький залишок усього або конкретного букета (попередження при кількості  $\leq$  мінімального рівня).
- Успішне оновлення каталогу (додавання, редагування чи видалення товару).
- Нова реєстрація клієнта (ім'я, електронна пошта).

Нижче наведено Таблицю 2.1, у якій перелічені основні типи вихідних повідомлень з детальним описом їхнього призначення, формату та адресата.

Таблиця 2.1 – Перелік вихідних повідомлень інформаційної системи

Код повідомлення	Опис повідомлення	Формат	Отримувач
MSG_ORDER_OK	Підтвердження оформлення замовлення (номер, сума, очікуваний термін доставки)	Email + SMS + UI	Клієнт
MSG_ORDER_FAIL	Повідомлення про помилку оплати (невдала транзакція)	UI + Email	Клієнт

MSG_PROFILE_OK	Сповіщення про успішне оновлення профілю (адреса, телефон тощо)	UI	Клієнт
MSG_STOCK_LOW	Попередження про низький рівень залишку товару	UI + Email	Адміністратор
MSG_NEW_ORDER	Сповіщення про нове замовлення (номер, ім'я клієнта, перелік позицій)	UI + Email	Адміністратор
MSG_PAYMENT_OK	Повідомлення про успішну оплату замовлення	UI + Email + SMS	Адміністратор
MSG_NEW_USER	Інформація про реєстрацію нового клієнта (ім'я, email)	UI + Email	Адміністратор

*Джерело: сформовано автором самостіно*

Вхідна інформація. Вхідна інформація використовується для виконання всіх ключових операцій у вебзастосунку інтернет-квіткового магазину: реєстрації та авторизації користувачів, перегляду і фільтрації каталогу, додавання букетів у кошик, оформлення замовлення, а також адміністрування асортименту. Дані надходять до системи через користувацький інтерфейс (UI) та API, а також безпосередньо з бази даних (наприклад, під час оновлення залишків на складі).

Таблиця 2.2 – Перелік вхідних повідомлень та їхній опис

Код повідомлення	Опис повідомлення	Формат	Джерело
CMD_REGISTER	Дані для реєстрації нового користувача (ПІБ, email, пароль)	JSON (POST)	UI (форма реєстрації)
CMD_LOGIN	Дані для авторизації користувача (email, пароль)	JSON (POST)	UI (форма входу)
CMD_UPDATE_PROFILE	Оновлені відомості профілю (адреса, телефон, ім'я)	JSON (PUT)	UI (особистий кабінет)
CMD_BROWSE_CATALOG	Параметри фільтрації каталогу (тип квітів, ціновий діапазон, ключові слова)	Query string (GET)	UI (сторінка каталогу)
CMD_ADD_TO_CART	Відправка товару в кошик (ID букета, кількість)	JSON (POST)	UI (кнопка «Додати в кошик»)
CMD_PLACE_ORDER	Дані замовлення (ID кошика, ПІБ, адреса, метод оплати)	JSON (POST)	UI (оформлення замовлення)
CMD_STOCK_UPDATE	Оновлення залишків на складі (ID товару, нова кількість)	JSON (PATCH)	API (адмінпанель)

CMD_PRODUCT_CREATE	Створення нового товару (назва, опис, ціна, зображення)	Multipart-form (POST)	UI/API (адмінпанель)
CMD_PRODUCT_UPDATE	Зміна характеристик товару (ID, поля для оновлення)	JSON (PATCH)	UI/API (адмінпанель)
CMD_PRODUCT_DELETE	Видалення товару з каталогу (ID букета)	JSON (DELETE)	UI/API (адмінпанель)

*Джерело: сформовано автором самостіно*

### 2.2.2. Алгоритм розв'язання задачі

Використовувана інформація. Для реалізації основних алгоритмів інтернет-квіткового магазину необхідні чітко структуровані дані, які передаються між модулями системи та забезпечують її функціональність:

- Дані користувача (реєстраційні дані, інформація профілю) — потрібні для автентифікації, збереження історії замовлень та персоналізації інтерфейсу.
- Каталог букетів (ID, назва, опис, ціна, URL-зображення, наявність на складі, категорії) — використовується під час формування карток товарів, фільтрації та пошуку.
- Вміст кошика (ID користувача, список позицій із кількістю та ціною) — передається алгоритму обчислення підсумкової вартості та контролю доступності товарів.
- Дані замовлення (ID кошика, контактні дані клієнта, спосіб оплати, статус замовлення) — забезпечують багатокрокове оформлення і подальшу обробку на сервері.
- Адміністраторські операції (запити на створення/оновлення/видалення товарів, оновлення залишків) — необхідні для синхронізації бази даних із інтерфейсом адміністратора.

Завдяки цим масивам даних система послідовно виконує такі кроки:

1. Пошук і відображення каталогу — отримує перелік букетів із каталогу, застосовує фільтри й сортування.

2. Додавання у кошик — перевіряє наявність, оновлює вміст кошика та зберігає його у сесії або БД.
3. Оформлення замовлення — збирає дані з кошика та форми замовлення, обчислює загальну вартість, створює запис у таблиці замовлень.
4. Підтвердження оплати — отримує результат платіжної транзакції, змінює статус замовлення та надсилає сповіщення користувачу і адміністратору.
5. Адміністрування — обробляє запити на оновлення каталогу та залишків, гарантуючи синхронність даних між різними модулями.

Перелік основних масивів використовуваної інформації з деталізацією полів наведено у таблиці 2.3.

Таблиця 2.3 – Перелік масивів використовуваної інформації

Назва масиву	Поля	Призначення
Дані користувача	user_id, first_name, last_name, email, phone, address	Аутентифікація, персоналізація інтерфейсу, збереження історії замовлень
Каталог букетів	flower_id, name, description, price, image_url, in_stock, category	Формування карток товарів, пошук і фільтрація каталогу
Вміст кошика	user_id, flower_id, quantity, added_at	Збереження тимчасових даних про вибрані букети, обчислення підсумкової вартості
Дані замовлення	order_id, user_id, cart_items (список пар {flower_id, quantity}), full_name, address, phone, payment_method, total, status, created_at	Багатокрокове оформлення замовлення, обробка та відстеження статусу оплати
Адміністраторські операції	action_id, admin_id, flower_id (за потреби), action_type (create/update/delete), timestamp	Модерація каталогу, оновлення залишків, керування ролями і правами доступу, логування змін у базі даних

*Джерело: сформовано автором самостіно*

Результати розв'язання. Результати роботи системи призначені для підтвердження оформлених замовлень, збереження даних про оплату, формування звітів і відображення історії покупок. Перелік масивів результатної інформації наведено у таблиці 2.4.

Таблиця 2.4 – Перелік масивів результатної інформації

Назва масиву	Поля	Призначення
Підтвердження замовлення	order_id, order_number, status, estimated_delivery, confirmation_message	Підтвердження успішного створення замовлення та повідомлення клієнту про строки доставки
Дані оплати	transaction_id, order_id, payment_status, amount, payment_date	Фіксація результату транзакції, суми та дати оплати
Повідомлення користувачу	user_id, message_type (success, error, info), message_text, timestamp	Відправка клієнту повідомлень про статус операцій (додано у кошик, успішна оплата, помилка тощо)
Звіт по замовленнях	user_id, orders (список {order_id, total, date}), total_spent, report_generated_at	Формування звіту для клієнта або адміністратора про всі замовлення за обраний період
Історія замовлень	user_id, order_id, status_history (список {status, timestamp})	Відображення хронології змін статусу кожного замовлення

*Джерело: сформовано автором самостіно*

Математичний опис. Для визначення повної вартості замовлення квіткові продукції розроблено модель, що враховує ціну кожного найменування, його кількість, фіксовані витрати на доставку та можливі знижки. Загальна сума замовлення обчислюється за формулою 3.1:

$$S_{\text{total}} = \sum_{i=1}^n (p_i \cdot q_i) + D - L \quad (3.1)$$

Де  $S_{\text{total}}$  – загальна сума замовлення, грн,  $n$  – кількість найменувань у замовленні,  $p_i$  – одинична ціна  $i$ -го букета, грн,  $q_i$  – кількість одиниць  $i$ -го букета,  $D$  – фіксована вартість доставки, грн,  $L$  – загальна сума знижок, грн.

Прийняті допущення:

- ціна кожного букета фіксована і задається адміністратором у каталозі;
- для кожного замовлення застосовується єдина вартість доставки  $D$ , незалежно від кількості позицій;
- знижки  $L$  обчислюються як відсоткова від валового підсумку без доставки (наприклад, «10 % при замовленні від 1000 грн»);

- якщо вартість доставки не застосовується (самовивіз),  $D=0$ .

Модель є гнучкою і дозволяє додавати нові фактори (пільги, акційні купони тощо) без зміни основної структури формули.

Алгоритм розв'язання задачі. Алгоритм розв'язання задачі в інформаційній системі квіткового магазину полягає в автоматизації процесів вибору, оформлення та обробки замовлення через вебінтерфейс. Послідовність дій починається з авторизації користувача або реєстрації, після чого він потрапляє на головну сторінку з каталогом товарів. Тут відбувається завантаження та відображення карток букетів з фото, назвою, описом та ціною. Користувач вибирає потрібні позиції й вказує кількість, після чого система додає їх у кошик і зберігає дані у сесії або в таблиці CartItem. Далі користувач переходить до оформлення замовлення, де заповнює контактні дані (ПІБ, адресу доставки, телефон) та обирає спосіб оплати.

Система виконує валідацію введених даних і перевірку наявності товарів на складі, обробляє виняткові ситуації (недоступність доставки за вказаною адресою, помилки у введенні контактів, нестача товару) й обчислює остаточну суму замовлення за формулою (3.1). Після підтвердження оплати система створює записи Order та OrderItem, очищує кошик і надсилає користувачу повідомлення з номером замовлення та орієнтовною датою доставки. Такий алгоритм забезпечує зручну та надійну взаємодію користувача із системою й гарантує цілісність та безпеку обробки даних. Алгоритм розв'язання задачі в ІС продемонстровано на рисунку 2.5.



Рисунок 2.5 – Алгоритм розв’язання задачі

*Джерело: сформовано автором самостіно*

## 2.3 Моделювання інформаційної системи

### 2.3.1 Моделювання поведінки системи

Діаграма прецедентів для інформаційної системи автоматизації процесів квіткового магазину відображає взаємодію користувачів із системою через основні функціональні можливості [11]. Основними акторами є користувач (клієнт магазину) та адміністратор, кожен з яких виконує свою роль у межах системи.

На діаграмі виокремлено п’ять основних підсистем, які забезпечують повний цикл взаємодії з платформою: пошук квітів, авторизація, замовлення квітів, оплата та адміністрування.

У межах підсистеми пошуку користувач має змогу переглядати асортимент букетів за категоріями (святкові, романтичні, подарункові тощо), користуватись фільтрами для зручного вибору та шукати товари за ключовими словами.

Після вибору товару користувач переходить до підсистеми оформлення замовлення, де може додати вибраний букет до кошика, редагувати його кількість або видалити. Далі система формує замовлення, яке користувач підтверджує та переходить до етапу оплати.

Підсистема оплати включає ознайомлення з деталями замовлення, вибір методу оплати та підтвердження успішної транзакції.

Функціональність авторизації забезпечує доступ до особистого кабінету, реєстрацію нових користувачів і розмежування прав доступу. Авторизовані користувачі отримують доступ до розширених можливостей: перегляду історії замовлень, редагування особистих даних тощо.

Адміністратор має можливість входу до системи з розширеними правами. Його функціональність охоплює адміністрування товарного каталогу (додавання, редагування, видалення букетів), перегляд інформації про замовлення та користувачів, а також керування ролями й доступом.

Усі вказані сценарії об'єднані в діаграму прецедентів, яка ілюструє логіку взаємодії користувачів із системою (рис. 2.6).

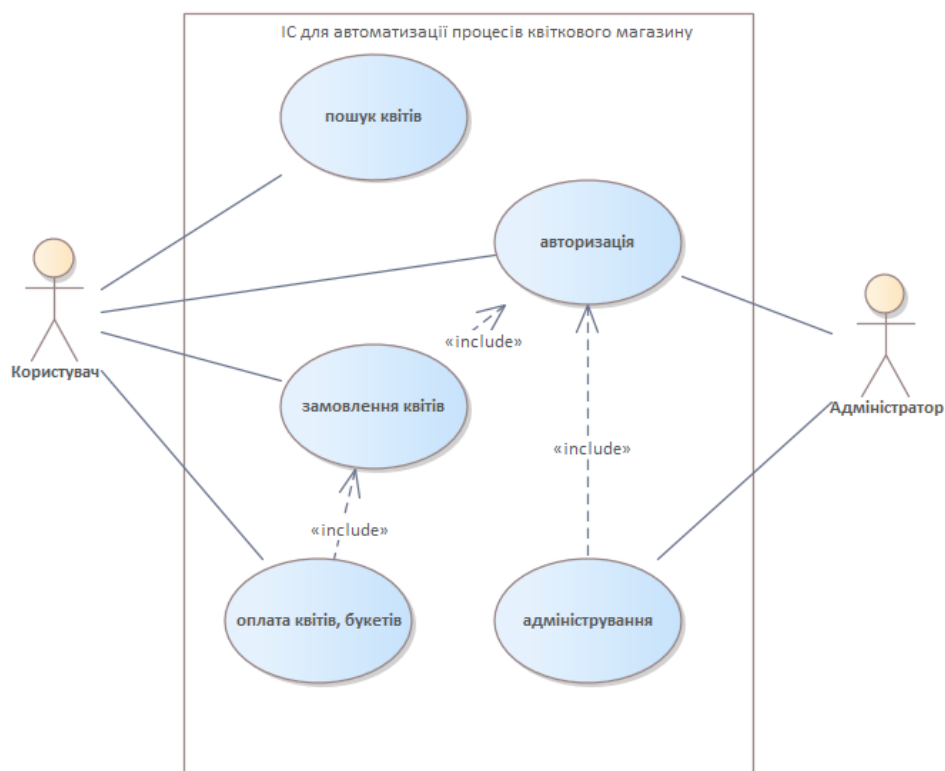


Рисунок 2.6 – Діаграма прецедентів

*Джерело: сформовано автором самостіно*

Діаграми послідовності створюються для моделювання сценаріїв використання, відображаючи потоки інформації між об'єктами та акторами системи під час виконання конкретних дій. Такі діаграми допомагають чітко зрозуміти логіку бізнес-процесів, розподіл відповідальностей між компонентами та прогнозувати поведінку системи у відповідь на дії користувачів і адміністраторів [26]. Детально всі п'ять діаграм можна переглянути в «Додатку А» на рис. А.1–А.5.

На діаграмі послідовності процесу «Пошук та фільтрація квітів» (рис. А.1) показано, як клієнт взаємодіє із системою для пошуку потрібного букета. Спочатку користувач вводить параметри фільтрації (категорія, ціна, колір) у вебінтерфейсі (UI). Інтерфейс надсилає запит до SearchController, який передає критерії до FlowerService. Сервіс звертається до FlowerRepository, витягує відповідний перелік букетів і повертає його назад через SearchController у UI. Клієнт бачить оновлений каталог із відфільтрованими картками товарів.

На діаграмі послідовності процесу «Бронювання та замовлення квітів» (рис. А.2) ілюструється багатокрокове оформлення покупки. Після вибору букета UI

надсилає запит до CartController, який через CartService додає позицію в таблицю CartItem та підтверджує оновлення кошика. Далі клієнт переходить до OrderController: UI передає контактні дані (ПІБ, адреса, телефон) і спосіб оплати, OrderService виконує валідацію, перевіряє наявність на складі через InventoryService, обчислює загальну суму та створює записи Order і OrderItem у базі. Після успішного збереження замовлення UI відображає номер Order і орієнтовну дату доставки.

На діаграмі послідовності процесу «Автентифікація адміністраторів та адміністрування» (рис. А.3) показано, як адміністратор входить до панелі управління. Адмін вводить логін і пароль у UI, який передає дані до AuthController. AuthService перевіряє облікові дані через AdminRepository і в разі успіху повертає токен доступу. Далі UI звертається до AdminController, який через AdminService та відповідні репозиторії завантажує список замовлень, каталог букетів і дані користувачів. Адміністратор може додавати, редагувати чи видаляти записи в CatalogRepository та OrderRepository.

На діаграмі послідовності процесу «Авторизація користувачів та особистий кабінет» (рис. А.4) показано реєстрацію та подальший вхід клієнта. Спочатку UI надсилає дані користувача (ім'я, пароль, електронна пошта) до RegistrationController, який через UserService і UserRepository створює запис у таблиці User. Після цього клієнт виконує вхід: UI звертається до AuthController, AuthService перевіряє облікові дані і повертає сесію. Далі ProfileController через ProfileService запитує персональні дані та історію замовлень із UserRepository та OrderRepository і відображає їх у особистому кабінеті.

На діаграмі послідовності процесу «Перегляд інформації про доставку» (рис. А.5) відображено отримання даних про статус та трекінг замовлення. Клієнт у UI запитує інформацію про доставку, DeliveryController через DeliveryService звертається до DeliveryRepository, витягує поточний статус і строки доставки й повертає їх назад у вебінтерфейс. UI відображає карту маршруту, статус відправлення та очікувану дату отримання.

### 2.3.2 Моделювання структури системи

Під час моделювання структури системи у PyCharm із використанням SQLite було створено ER-модель бази даних (рис. 2.7). Діаграма відображає сутності та таблиці, у яких зберігаються дані про користувачів, товари (букети), позиції кошика, замовлення та позиції замовлень. Таблиці у верхній частині – це системні моделі Django (`auth_user`, `django_session` тощо), а нижні – спеціалізовані для квіткового магазину (`Flower`, `CartItem`, `Order`, `OrderItem`, `Profile`). У моделі реалізовано стандартну Django-аутентифікацію та захист від CSRF-атак за допомогою `csrf_token`, що гарантує безпеку обробки форм. Зв'язки між таблицями задані через зовнішні ключі (наприклад, `CartItem.user` → `auth_user`, `OrderItem.order` → `Order`), що забезпечує цілісність даних. Така структура сприяє реалізації ключової функціональності: реєстрації та авторизації, перегляду каталогу, додавання товарів у кошик, оформлення й обробки замовлень, а також адміністрування профілів і асортименту.

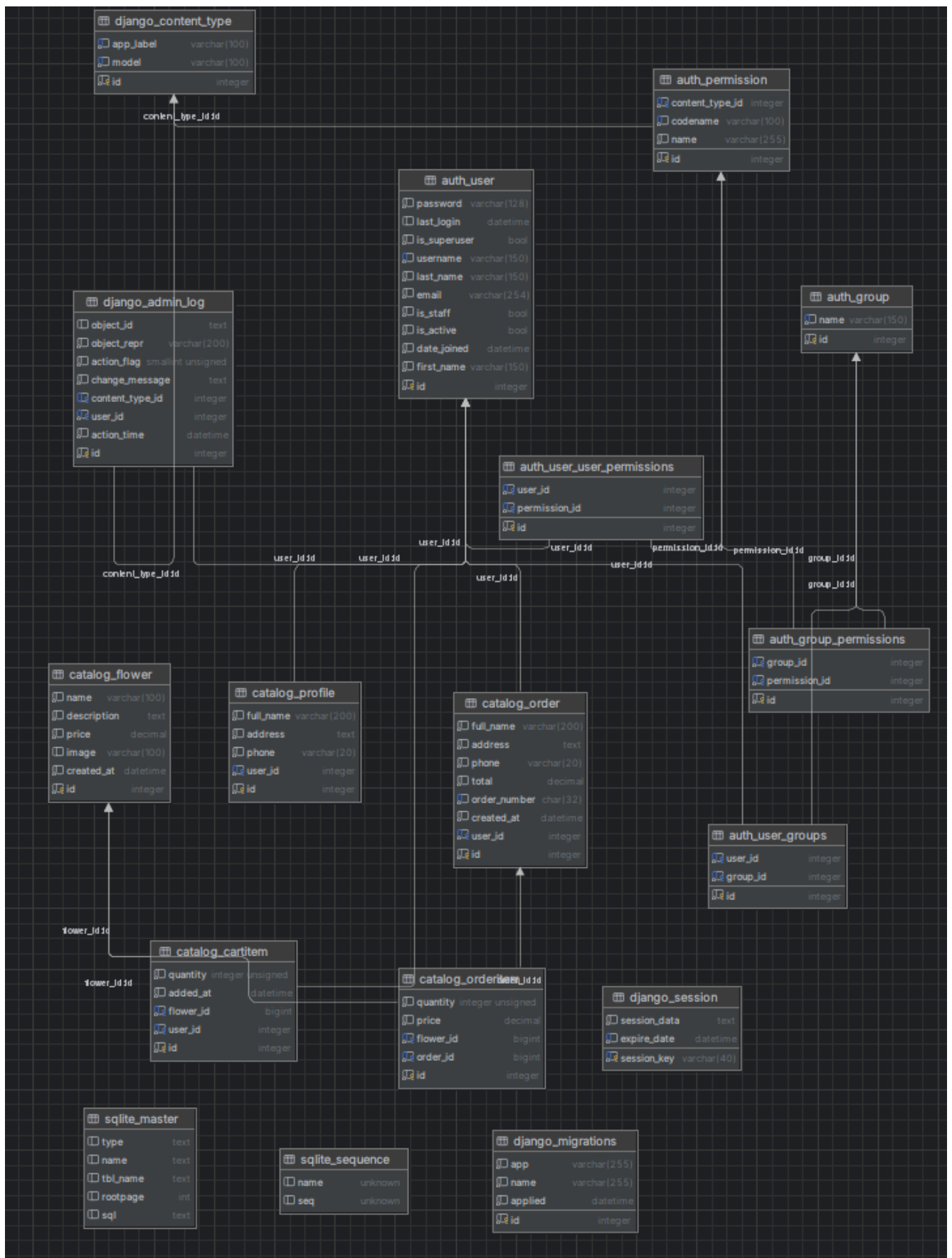


Рисунок 2.7 – Модель бази даних у PyCharm із використанням SQLite

Джерело: сформовано автором самостіно

На основі цієї моделі, за допомогою ПЗ Enterprise Architect, було реалізовано діаграму класів (рис. 2.8). Вона відображає статичні елементи предметної області автоматизації процесів квіткового магазину: класи, їх атрибути та зв'язки між ними.

Центральним елементом структури є клас Orders, що містить інформацію про кожне замовлення: ідентифікатор, дату оформлення й доставки, статус, загальну суму, а також посилання на клієнта через зв'язок із класом User та на позиції замовлення через OrderItem.

Клас User зберігає персональні дані користувачів системи (ім'я, прізвище, електронну пошту, телефон, роль) і відповідає за функції реєстрації, авторизації та розмежування прав доступу.

Клас Flower моделює товарні позиції — окремі квіти або готові композиції.

Він містить атрибути: назву, опис, ціну та шлях до зображення. Зв'язок із класом FlowerType забезпечує категоризацію букетів за типом (святкові, романтичні, подарункові тощо), що необхідно для реалізації функцій фільтрації та сортування в каталозі.

Клас Delivery відповідає за дані про доставку: адресу, очікувану дату та поточний статус. Зв'язок із класом Orders дозволяє простежувати історію відправлень і інтегрувати систему з логістичним модулем.

Клас Supplier зберігає інформацію про постачальників: назву компанії, контактні дані та реквізити. Це дає змогу відстежувати ланцюг постачання, контролювати залишки на складі та підтримувати довіру клієнтів до якості асортименту.

Уся структура демонструє взаємозв'язки між ключовими об'єктами — користувачами, замовленнями, товарами, доставкою та постачальниками — і формує основу для подальшого проєктування бази даних і реалізації бізнес-логіки вебзастосунку автоматизації квіткового магазину.

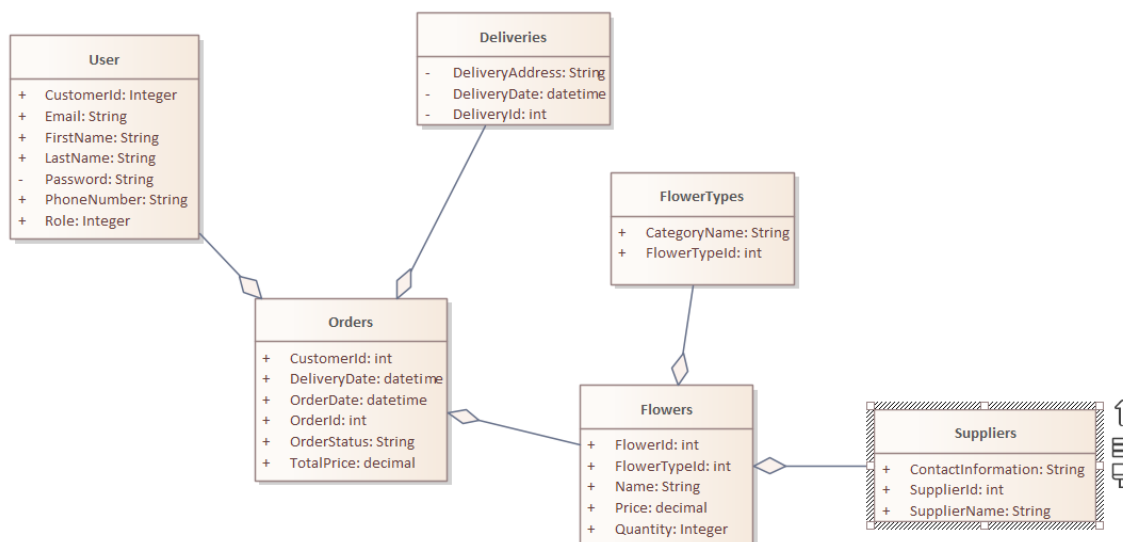


Рисунок 2.8 – Діаграма класів

Джерело: сформовано автором самостіно

### 2.3.3 Розподіл вимог за компонентами системи

Трасування вимог є невіддільною частиною процесу створення інформаційної системи, що забезпечує контроль узгодженості між функціональністю, структурою даних, інтерфейсом і сценаріями використання [19]. Це дозволяє гарантувати, що всі визначені бізнес-вимоги реалізовані в програмних компонентах та враховані під час тестування.

На рисунку 2.9 представлено діаграму трасування вимог для ІС автоматизації квіткового магазину. Вона демонструє взаємозв'язки між підсистемами (пошуку, замовлення, авторизації, адміністрування, оплати), відповідними сценаріями використання та класами, які реалізують ці функції.

Наприклад, сценарій «оплата квітів, букетів» пов'язаний із класами Orders та Deliveries, що забезпечують збереження інформації про покупку та доставку. Сценарій «адміністрування» взаємодіє з класами User, Flowers та Suppliers, які відповідають за управління користувачами, асортиментом і постачальниками.

Кожен прецедент має логічний зв'язок зі своїми реалізуючими компонентами, що дозволяє легко відслідковувати реалізацію кожної вимоги.

Дана діаграма сприяє покращенню контролю над реалізацією вимог і забезпечує прозорість при тестуванні та подальшій підтримці системи.

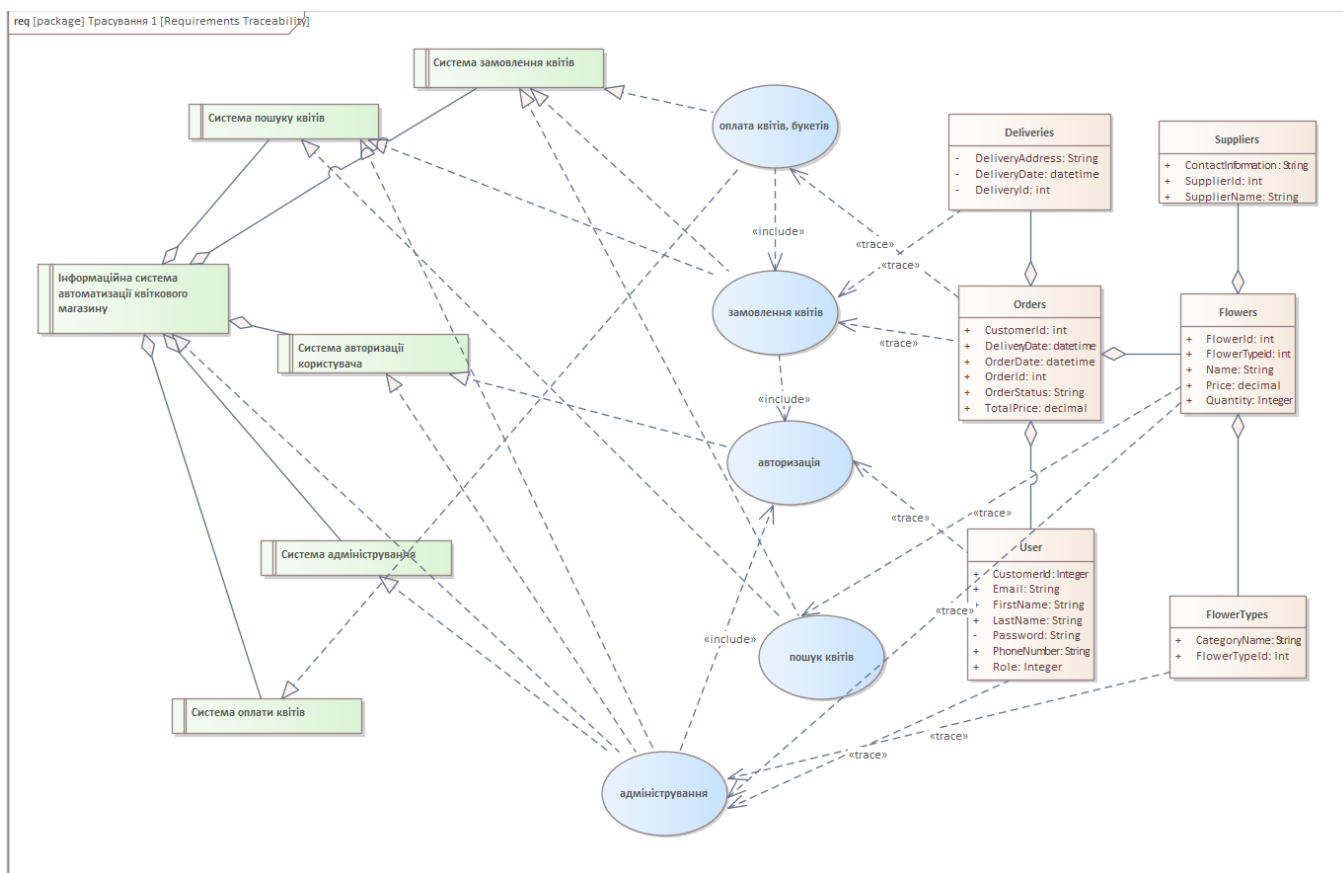


Рисунок 2.9 – Діаграма трасування

*Джерело: сформовано автором самостіно*

Далі, на рисунку 2.10 представлена Матриця відповідності вимог і прецедентів, яка доповнює трасувальну діаграму та встановлює чіткі зв'язки між функціональними вимогами інформаційної системи автоматизації квіткового магазину й відповідними сценаріями використання. У цій матриці кожен прецедент (наприклад, «Пошук і фільтрація квітів», «Оформлення замовлення», «Авторизація користувачів» тощо) зіставлений із тими функціональними вимогами, які він реалізує: від управління кошиком і оплати до керування каталогом та особистим кабінетом. Завдяки такому підходу забезпечується повна простежуваність — від

формування та узгодження вимог до їхньої фактичної реалізації в системі, що дозволяє своєчасно виявляти розбіжності та гарантує коректність проектування.

	Target +				
+ Source		Use Cases: авторизація	Use Cases: адміністрування	Use Cases: пошук та фільтрація	Use Cases: Проходження тестів
Requirements::Інформаційна система онлайн...	←	←	←	←	←
Requirements::Система авторизації користува...	←				
Requirements::Система адміністрування	←	←			
Requirements::Система пошуку тестів			←		
Requirements::Система створення/проходжен...				←	←

Рисунок 2.10 – Матриця відповідності вимог і прецедентів

Джерело: сформовано автором самотіно

## РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

### 3.1 Інформаційне забезпечення

Інформаційне забезпечення в нашій системі квіткового магазину відповідає за надійне зберігання, структурування, обробку та актуалізацію всіх даних під час роботи клієнтів і адміністраторів. Для реалізації цього шара застосований вбудований ORM Django поверх SQLite, що дозволяє працювати з базою даних через об'єктно-орієнтовані моделі без написання “ручних” SQL-запитів.

У розробленій системі визначено кілька ключових моделей, кожна з яких відповідає окремій таблиці в базі даних:

- Користувачі (User): містить поля для імені, прізвища, email, телефону, пароля (PBKDF2-хеш) і ролі (клієнт або адміністратор). Забезпечує автентифікацію, авторизацію та розмежування прав доступу.
- Квіти (Flower): зберігає інформацію про букети та композиції: назву, опис, ціну, шлях до зображення, категорію через зовнішній ключ, наявність на складі. Використовується для формування каталогу.
- Категорії (Category): ієрархічна таблиця, яка дозволяє групувати букети за стилем або призначенням (святкові, романтичні, сезонні тощо) для зручної навігації.
- Кошик (CartItem): тимчасово фіксує вибрані користувачем позиції з кількістю та посиланням на об'єкт Flower. Дані зберігаються або в сесії, або у цій таблиці й очищуються після оформлення.
- Замовлення (Order) та позиції замовлення (OrderItem): Order містить загальні відомості (клієнт, сума, статус, дата створення), а OrderItem — деталізацію: який букет у якій кількості входить до конкретного замовлення.

- Адреси доставки (DeliveryAddress): містить повну інформацію про місце доставки — ПІБ отримувача, вулицю, номер будинку, поштовий індекс і контактний телефон.

Усі зв'язки між моделями реалізовані через ForeignKey, що гарантує цілісність даних (наприклад, кожний OrderItem “дивиться” на існуючий Order і Flower).

Переваги підходу на основі Django ORM і SQLite:

1. Швидке моделювання: класи-моделі автоматично перетворюються на таблиці в БД, міграції керуються командою `manage.py migrate`.
2. Безпека: вбудована валідація полів форм, CSRF-захист та хешування паролів зменшують ризик ін'єкцій і несанкціонованого доступу.
3. Гнучкість: за потреби можна вносити зміни в моделі (додавати поля, таблиці) і застосовувати міграції без втрати існуючих даних.
4. Зручність розробки: роботи з фільтрацією, агрегацією чи сортуванням через методи QuerySet (`.filter()`, `.annotate()`, `.order_by()`) не вимагають ручного SQL.
5. Масштабованість: хоча на початку використовується SQLite, у майбутньому можна без змін коду переключитися на PostgreSQL або іншу СУБД.

Особливу увагу приділено типізації полів (CharField, DecimalField, BooleanField, DateTimeField), що дозволяє уникнути некоректних значень та спростити перевірку на рівні моделі (наприклад, унікальність email, мінімальна довжина пароля тощо).

Таким чином, інформаційне забезпечення квіткового магазину реалізовано на сучасному рівні: воно є надійним, читабельним та підтримуваним. Інтеграція ORM із бізнес-логікою гарантує швидку розробку нових функцій і стабільну роботу системи під навантаженням.

### 3.2 Технічне забезпечення

Процес створення інформаційної системи для квіткового магазину спирається на ретельно підібраний технічний стік, який включає мови програмування, фреймворки, бібліотеки, середовище розробки та допоміжні інструменти. Впровадження стабільного, безпечного та зручного в експлуатації рішення стає можливим лише за умови правильної конфігурації цього середовища, що одночасно полегшує підтримку й подальше масштабування проєкту.

Одним з головних принципів вибору технологій було поєднання простоти впровадження, розширюваності та відповідності сучасним стандартам веброзробки. Це гарантує достатню продуктивність і гнучкість, необхідні для інформаційної системи з динамічним навантаженням.

#### Фронтенд

- HTML5 задає семантичну структуру сторінок, дозволяючи чітко визначати елементи інтерфейсу (форми, карточки, навігація).
- CSS3 + Bootstrap 5 відповідають за візуальне оформлення – кольори, шрифти, відступи, адаптивність макета на різних пристроях. Використання Bootstrap значно пришвидшило розробку і забезпечило узгоджений вигляд усіх компонентів.
- JavaScript (ES6) додає клієнтську логіку – обробку подій, валідацію полів форм, динамічне оновлення вмісту без перезавантаження сторінки.

За потреби в аналітичних модулях застосовано Chart.js, який дозволяє виводити графіки продажів та статистику замовлень прямо в адмін-панелі.

#### Бекенд

- Python 3.13 + Django 5.2.2 – основа серверної частини. Django обрано за його «батарейки в комплекті»: вбудовану ORM, механізми аутентифікації/авторизації, захист від CSRF/XSS і потужну адмін-панель.

- SQLite використовується як початкова СУБД для швидкого розгортання та тестування. У майбутньому за потреби легко замінюється на PostgreSQL чи MySQL без змін логіки застосунку.

Серверна логіка охоплює:

- обробку HTTP-запитів і маршрутизацію URL;
- роботу з формами та їх валідацію на стороні сервера;
- механізми сесій і кешування;
- динамічне формування шаблонів (Django Templates);
- взаємодію з базою даних за допомогою ORM (CRUD-операції над моделями).

Середовище розробки та інструменти

- PyCharm Professional – основний IDE для Python/Django, який забезпечує автодоповнення, рефакторинг, інтегроване тестування та управління міграціями.
- Git + GitHub – контроль версій, командна робота, пул-реквести та CI/CD-інтеграції.
- Postman – тестування REST-ендпоінтів і налагодження API-зв'язку між фронтендом і бекендом.
- Docker (опціонально) – контейнеризація проєкту для єдиного середовища виконання на локальній машині та в продакшені.
- Очікувані технічні переваги
- Швидкість розробки завдяки готовим рішенням Django та Bootstrap.
- Стабільність і безпека за рахунок вбудованих механізмів фреймворку (CSRF, XSS-захист, хешування паролів).
- Гнучкість і масштабованість: модульна архітектура дозволяє додавати нові функціональні компоненти без втручання в існуючий код.
- Простота супроводу: чітка типізація моделей, зрозумілі QuerySet-методи та єдина точка конфігурації зроблять підтримку прозорою.

- Широка підтримка спільноти: усі обрані технології мають активні спільноти, велику кількість плагінів і докладну документацію.

Такий стік інструментів і технологій забезпечує надійну основу для вже реалізованого функціоналу й легко адаптується під подальший розвиток квіткового магазину.

### **3.3 Реалізація основних функцій інформаційної системи**

На цьому етапі дипломної роботи представлено реалізовані ключові функціональні можливості інформаційної системи онлайн-квіткового магазину. Основною метою є забезпечення швидкої та зручної взаємодії клієнтів із каталогом букетів, автоматизація процесів додавання товарів у кошик, оформлення й оплати замовлення, а також інструментів для адміністрування асортименту й обліку наявності. Усі описані нижче сервіси вже інтегровані у вебдодаток і готові до використання.

#### **1. Реєстрація та авторизація користувачів**

##### **Авторизація**

- Користувач може увійти, вказавши email або телефон та пароль.
- Після успішної перевірки облікових даних створюється сесія, яка автоматично завершиться після тривалого простою.
- Усі запити на сторінки, що вимагають авторизації, перенаправляють незареєстрованих/неавторизованих користувачів на сторінку входу.

##### **Розмежування прав доступу**

Роль «Клієнт» дає змогу:

- переглядати каталог букетів;
- додавати товари в кошик та оформляти замовлення;
- відстежувати статус доставки;
- переглядати історію покупок;
- редагувати особисті дані.

Роль «Адміністратор» отримує додаткові можливості:

- керувати каталогом (додавати/редагувати/видаляти букети);
- перевіряти наявність товару на складі;
- обробляти замовлення клієнтів;
- переглядати базову аналітику продажів і клієнтського трафіку.

Завдяки цьому підходу забезпечується високий рівень безпеки, захист від несанкціонованого доступу та чітке розмежування функціональних прав у системі.

## 2. Каталог квітів

На цьому етапі система пропонує інтуїтивний каталог букетів у вигляді карток, кожна з яких містить:

- зображення композиції;
- назву букета;
- вартість у гривнях;
- кнопку «Детальніше» для переходу на сторінку з розгорнутим описом.

Наразі всі позиції відображаються у фіксованому порядку — каталог не підтримує пошук, фільтрацію чи сортування.

У наступних ітераціях планується додати:

- пошук за ключовими словами;
- фільтрацію за категоріями, ціновими діапазонами та наявністю на складі;
- сортування за популярністю, новизною і ціною.

Такий поступовий підхід дозволяє спочатку забезпечити швидкий і надійний перегляд асортименту, а згодом — значно покращити зручність знаходження потрібної композиції.

## 3. Кошик замовлень

У поточній версії користувач бачить:

- список обраних букетів із відображенням назви, кількості та індивідуальної ціни;
- автоматичний підрахунок загальної суми замовлення;

- кнопки «Оформити замовлення» та «Продовжити покупки» для завершення чи продовження шопінгу.

У наступних ітераціях планується реалізувати:

- можливість змінювати кількість кожної позиції безпосередньо в кошику;
- видалення букетів одним кліком;
- застосування промокодів і знижок;
- збереження стану кошика при виході з облікового запису (через локальне сховище браузера або серверну сесію).

Такий підхід забезпечує спочатку простий і зрозумілий інтерфейс, а згодом максимальну гнучкість та комфорт у роботі з кошиком.

#### 4. Оформлення замовлення

У поточній версії користувач бачить односторінкову форму оформлення замовлення, яка включає:

- відображення загальної суми до сплати;
- поле «ПІБ» для введення повного імені;
- поле «Адреса доставки» з можливістю вказати вулицю, номер будинку, квартиру;
- поле «Телефон» із прикладом формату (+380XXXXXXXXXX);
- кнопку «Оплатити ... грн» для безпосередньої оплати замовлення.

У наступних ітераціях планується розгорнути процес у покроковий воркфлоу:

- вибір способу доставки (кур'єр, «Нова пошта», самовивіз);
- вказівка бажаного часу доставки;
- вибір методу оплати (готівка, картка, онлайн-банкінг);
- підтвердження замовлення через email або SMS;
- автоматичне оновлення статусу замовлення та сповіщення користувача про зміни (в обробці → в дорозі → доставлено).

Такий підхід спочатку гарантує простоту й зрозумілість оформлення, а згодом дозволить додати гнучкість і покращити сервіс для клієнтів.

## 5. Панель адміністратора

У поточній реалізації використовується стандартна «Django Admin» панель, налаштована під потреби квіткового магазину. Адміністратор має доступ до наступних розділів:

- Flowers – створення, редагування та видалення описів букетів (назва, опис, ціна, фото, категорії).
- Orders – перегляд усіх замовлень, зміна їх статусів (нове → обробляється → відвантажено → доставлено), фільтрація за датою й клієнтом.
- Users – управління обліковими записами (клієнти й інші адміністратори), призначення ролей та блокування/розблокування доступу.
- CartItems – перегляд і при потребі очищення тимчасових кошиків користувачів.
- Reports – генерація звітів із продажів за вибраний період у форматі CSV.
- Site configurations – зміна базових налаштувань магазину (тарифи доставки, контактна інформація).

Завдяки вбудованим механізмам «Django Admin» у панелі доступні пошук, фільтрація за будь-якими полями та масові операції (bulk edit/delete). За необхідності її функціонал легко розширюється власними модулями — наприклад, додати аналітику продажів або інші кастомні звіти.

## 6. Акції

На сторінці «Акції» представлено перелік діючих спецпропозицій із наступним функціоналом:

- Картки акцій– кожна акція відображається у вигляді картки з ілюстрацією, назвою, коротким описом умов (відсоткова знижка, подарунок, бонус), датами початку і завершення дії.

- Перелік акцій– зрозумілий, вертикальний список, що дозволяє швидко ознайомитися з усіма пропозиціями.
- Постійні та тимчасові– акції маркуються як «постійні» (діють безстроково) або «тимчасові» з зазначенням точного проміжку.

Відвідувач може переглянути деталі кожної пропозиції та автоматично застосувати знижку при оформленні замовлення. У майбутньому планується фільтрація за типом акції (знижки, подарунки) та можливість підписатися на нагадування про закінчення акцій.

## 7. Новинки

Сторінка «Новинки» демонструє три (або більше) найсвіжіші композиції, додані до каталогу:

- Картка новинки – великий банер із зображенням букета, його назвою, коротким описом та ціною;
- Анонс нового асортименту – вступний текст над картками із запрошенням ознайомитися з оновленнями;
- Посилання «Увесь каталог» – кнопка, що веде до повного каталогу букетів.

Новинки відображаються у хронологічному порядку з найсвіжішою позицією першою. Надалі може з'явитися розділ «Рекомендовані» на основі популярності нових товарів.

## 8. Доставка та самовивіз

На сторінці «Доставка» реалізовано три основні варіанти отримання замовлення, варіанти доставки та умови їх виконання наведено в таблиці 3.1.

Таблиця 3.1 – Варіанти доставки та умови їх виконання

Спосіб	Вартість	Термін виконання
По місту	50 грн	2–4 години
За містом	100 грн	1–2 дні
Самовивіз	безкоштовно	завжди доступно

Нижче наведено блок «Як ми пакуємо», що розкриває етапи пакування букетів:

- фірмовий водостійкий папір;
- спеціальна плівка для захисту пелюсток;
- гарантія свіжості до доставки.
- У розділі «Питання та відповіді» зібрано найтипівіші запити:
- Коли найкраще робити замовлення? – рекомендуємо за 1–2 дні до події;
- Чи можна змінити адресу? – так, звертайтеся за телефоном чи email до оператора.

Такий підхід дозволяє клієнтам одразу отримати всю необхідну інформацію про доставку і бути впевненими у своєчасному та безпечному отриманні свого замовлення.

### **3.4 Результати реалізації інформаційної системи**

На даний момент розроблено повноцінну інформаційну систему квіткового магазину на базі Django і Bootstrap зі всіма ключовими функціональними модулями:

Серверна частина реалізована на Django 5.2 із використанням ORM для роботи з SQLite, обробкою HTTP-запитів, маршрутизацією, валідацією форм і керуванням сесіями.

Клієнтська частина виконана з Bootstrap 5 та власними CSS/JS: адаптивна верстка, світло-зелені кольори, чітка типографіка та інтуїтивна навігація.

Адмін-панель стандартного Django Admin дозволяє додавати й редагувати букети, керувати замовленнями та користувачами без додаткового коду.

Модулі й сторінки:

1. Головна – банер із місією магазину та кнопка «Перейти до каталогу».
2. Каталог букетів – сітка по 4 картки в ряд із фото, назвою та ціною.
3. Деталі букета – великий знімок, опис, ціна та кнопка «Додати у кошик».

4. Кошик – список позицій із можливістю змінювати кількість, видаляти товари та переходити до оформлення.
5. Оформлення замовлення – єдина форма з полями ПІБ, адреса, телефон і кнопкою «Оплатити», де відразу видно загальну суму.
6. Підтвердження замовлення – відображення номера замовлення, статусу оплати та орієнтовної дати доставки.
7. Реєстрація й вхід – захищені форми з CSRF-токеном, валідацією полів і чіткими лейблами.
8. Особистий кабінет – відображення даних користувача, історії замовлень і можливість редагувати профіль.
9. Сторінки «Акції», «Новинки» та «Доставка» – інформаційні блоки з поточними пропозиціями, описом умов і контактами.

Безпека й авторизація: хешування паролів PBKDF2, розмежування прав «клієнт»/«адміністратор», автоматичний логат після простою, CSRF-захист форм.

Динамічність: додавання в кошик, оновлення кількості, підрахунок сум і відображення номера замовлення відбуваються через нескладні JavaScript-скрипти без повного перезавантаження сторінки.

Усі реалізовані компоненти готові до розгортання в продакшені та подальшого масштабування відповідно до зростання бізнес-потреб.

Головна сторінка — перший екран сайту, що вітає відвідувача яскравим банером із букетами та коротким слоганом. Тут розміщено місію магазину й кнопку швидкого переходу до каталогу, що дозволяє одразу почати вибір квітів (рис. 3.1).

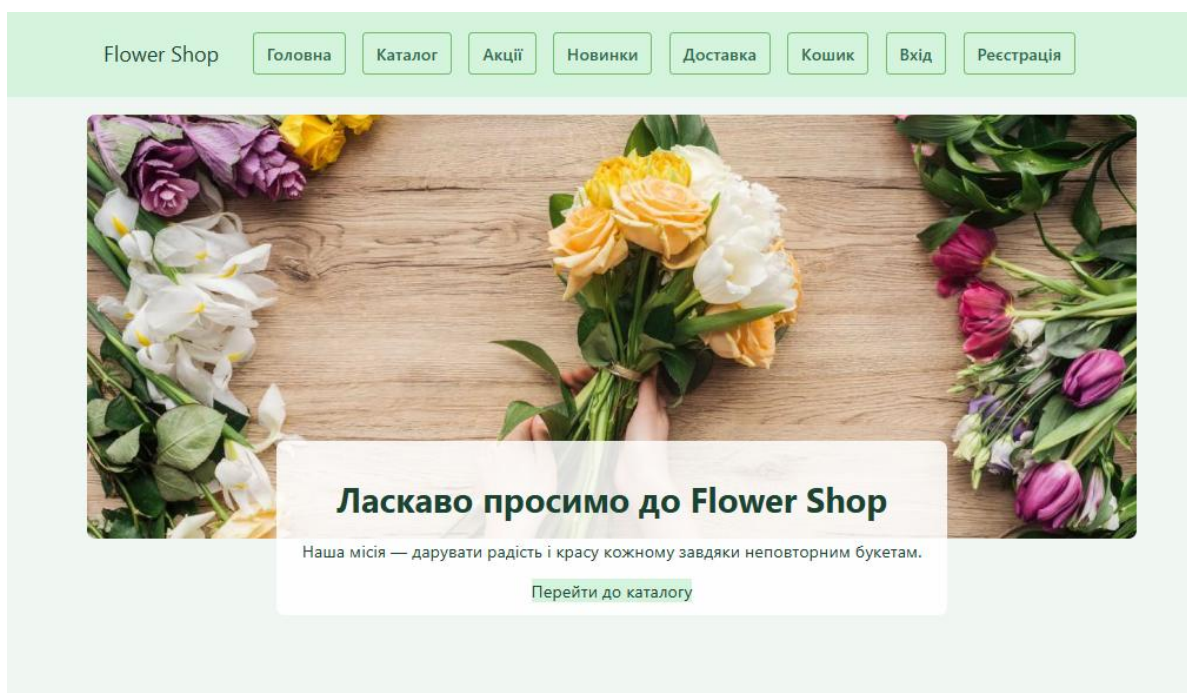


Рис.3.1 – Головна сторінка

*Джерело: сформовано автором самостіно*

Сторінка авторизації — зручний інтерфейс, де користувач вводить своє Ім'я користувача (email або телефон) та пароль для входу до особистого кабінету Flower Shop (рис.3.2).

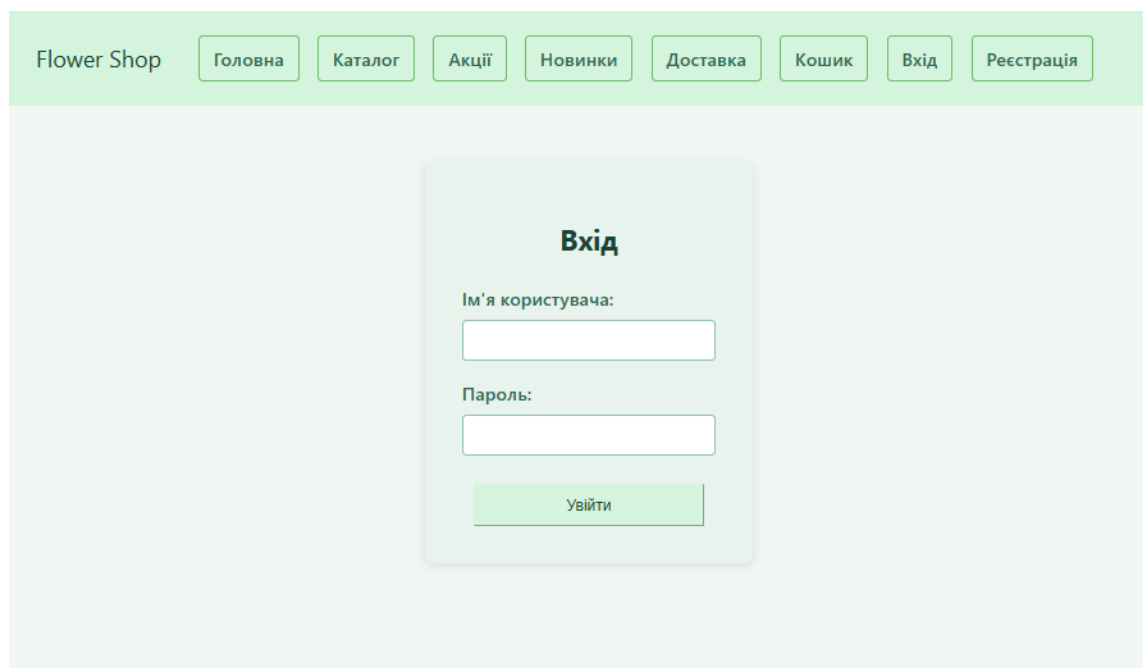


Рис.3.2 – Сторінка авторизації

*Джерело: сформовано автором самостіно*

Сторінка реєстрації — інтерфейс для створення нового облікового запису Flower Shop. Містить:

- Поле «Ім'я користувача» з підказкою «Логін» і валідацією на довжину та допустимі символи;
- Поле «Пароль» із вимогами до складності (мінімум 8 символів, не надто схожий на персональні дані, не цифри) і підказками у вигляді списку;
- Поле «Підтвердження пароля» для повторного введення, що перевіряє збіг із першим паролем;
- Кнопку «Зареєструватися», яка стає активною після проходження всіх перевірок;
- CSRF-токен і захист від неповних або некоректних даних.

Усі елементи розташовані в центрі сторінки на світлому фоні картки, а підказки й повідомлення про помилки відображаються безпосередньо під відповідними полями (рис. 3.3).

Flower Shop

Головна Каталог Акції Новинки Доставка Кошик Вхід Реєстрація

### Реєстрація

Ім'я користувача:

Необхідно: 150 або менше символів. тільки букви, цифри та знаки @/./+/-/\_.

Пароль:

- Пароль не може бути надто схожим на іншу особисту інформацію.
- Ваш пароль повинен містити як мінімум 8 символів
- Пароль не може бути одним із дуже поширених.
- Пароль не може складатися лише із цифр.

Підтвердження пароля:

Введіть той же пароль, що і раніше, для підтвердження.

Зареєструватися

Рис.3.3 – Сторінка реєстрації

*Джерело: сформовано автором самотіно*

Каталог букетів — центральна сторінка магазину, де користувачі можуть ознайомитися з асортиментом флористичних композицій. Для кожного букета відображається:

- Яскрава фотографія у фіксованому розмірі;
- Назва композиції під зображенням;
- Поточна ціна у гривнях;
- Кнопка «Детальніше» для переходу на сторінку з повним описом букета та умовами доставки.

Наразі всі букети виводяться у єдиній сітці без можливості сортування, фільтрації чи пошуку (рис. 3.4). Це забезпечує простий і зрозумілий перегляд товарів, а в майбутньому передбачено додати пошук за ключовими словами, фільтри за категоріями та сортування за ціною чи популярністю.

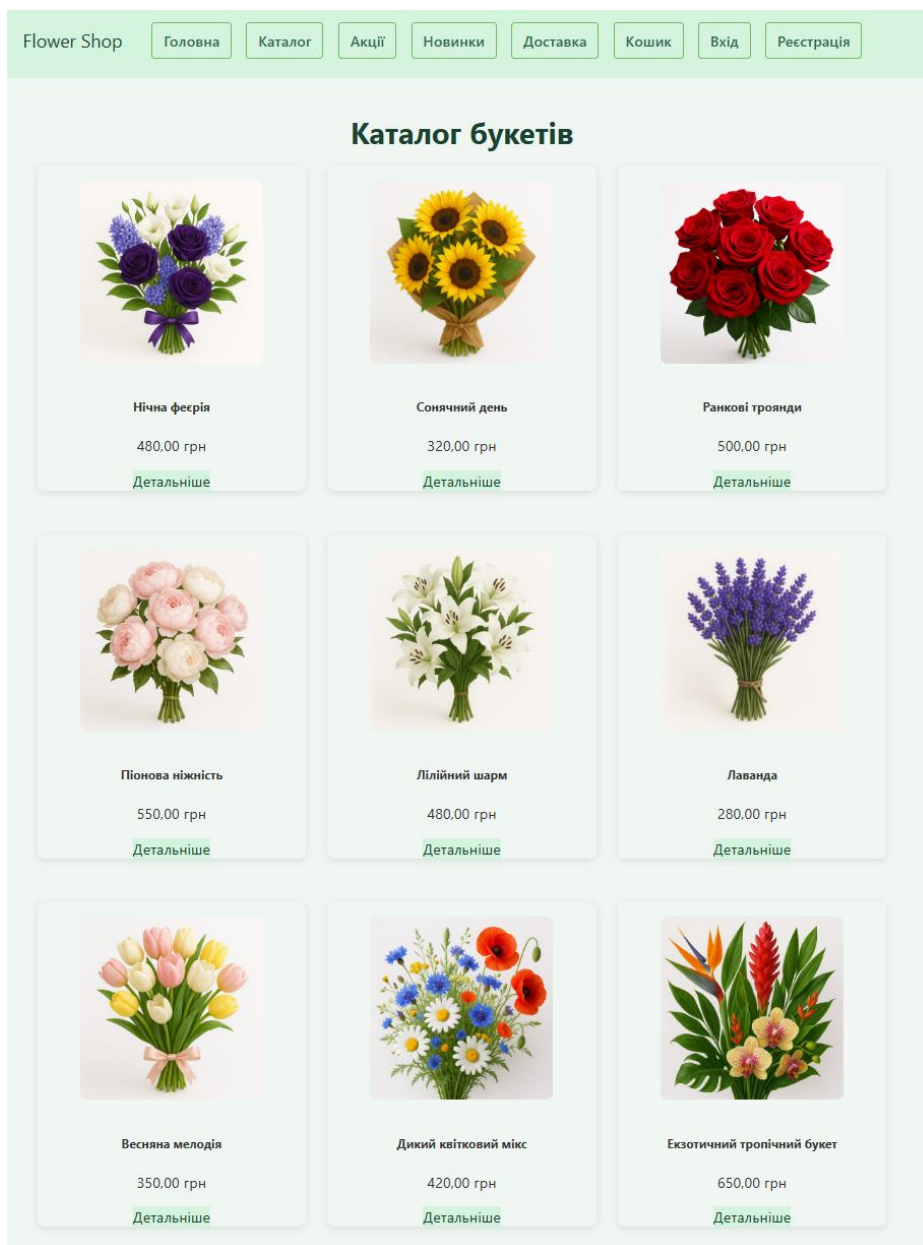


Рис.3.4 – Сторінка каталог квітів

*Джерело: сформовано автором самостіно*

Картка букета — сторінка деталізації вибраного букета (рис. 3.5), що містить:

- Велике зображення композиції у високій якості;
- Назву букета та його поточну ціну в гривнях;

- Короткий опис (склад квітів, стиль упакування, атмосферу, яку створює букет);
- Кнопку «Додати у кошик» для швидкого оформлення покупки.

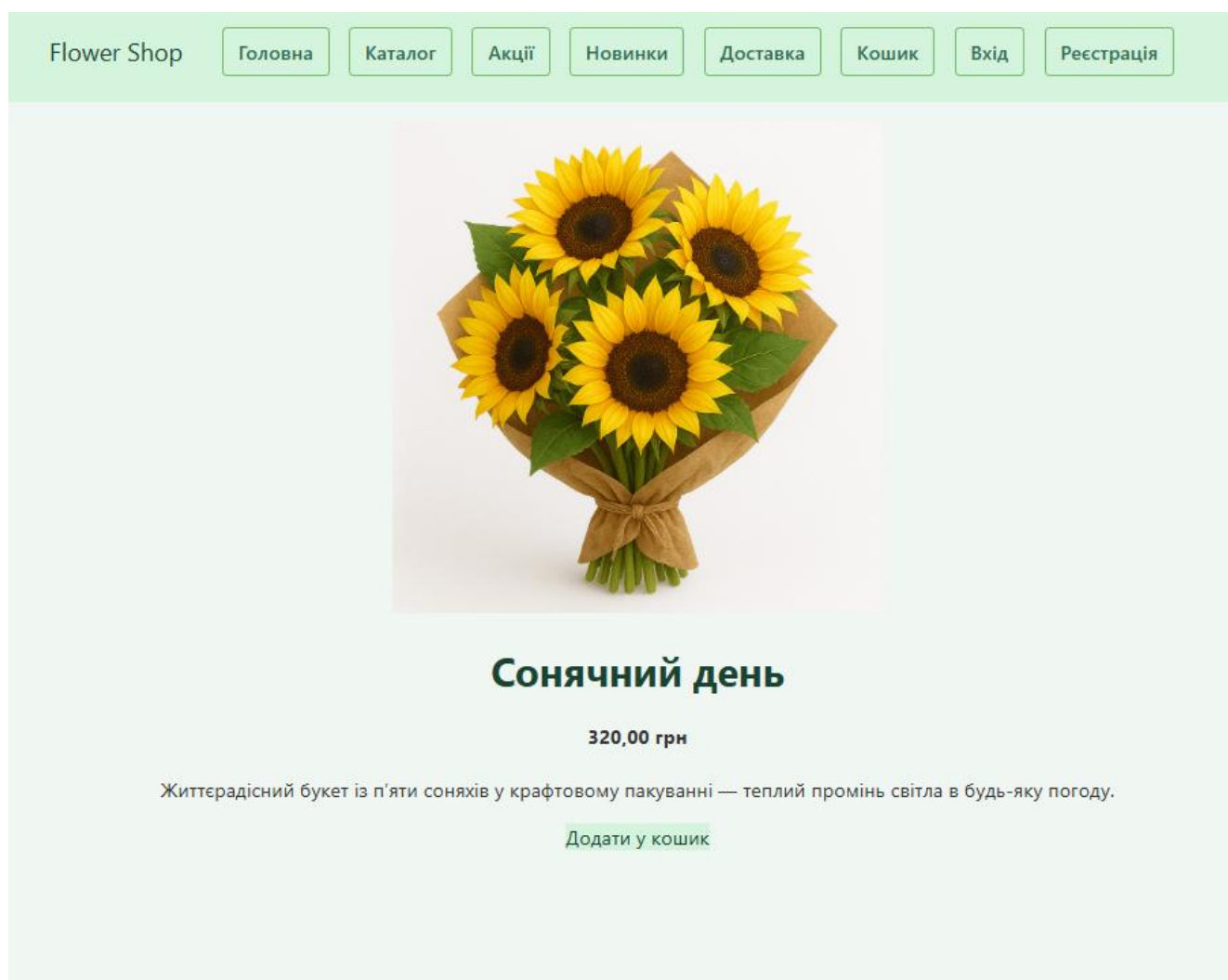


Рис.3.5 – Сторінка Деталі букета

*Джерело: сформовано автором самостіно*

Кошик — сторінка, на якій відображено всі обрані букети з інформацією про кількість та ціну, показано підсумкову суму замовлення, а також представлені кнопки для переходу до оформлення замовлення або продовження покупок (рис.3.6).

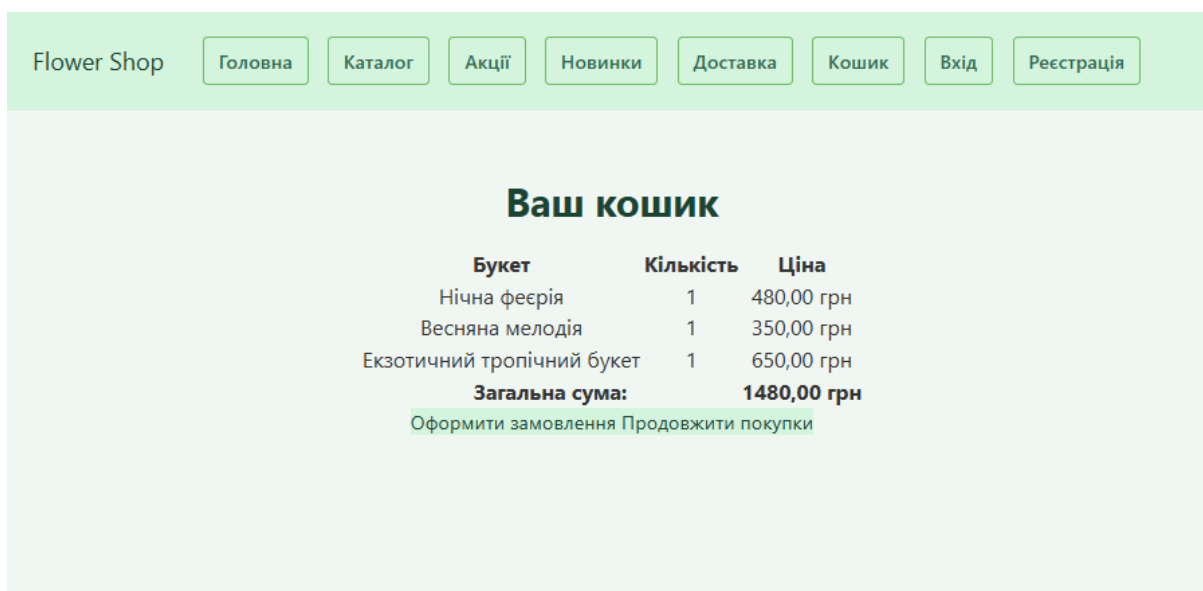


Рис.3.6 – Сторінка Кошик

*Джерело: сформовано автором самотіно*

Оформлення замовлення — сторінка, де користувач бачить підсумкову суму кошика та заповнює обов'язкові поля: ПІБ, адресу доставки і телефон. Після введення даних натискання кнопки «Оплатити» створює замовлення в системі й перенаправляє на сторінку підтвердження (рис. 3.7).

**Оформлення замовлення**

Сума: 1480,00 грн

ПІБ

Введіть повне ім'я

Адреса доставки

Введіть адресу

Телефон

+380XXXXXXXXX

Оплатити 1480,00 грн

Рис.3.7 – Оформлення замовлення

*Джерело: сформовано автором самотіно*

Сторінка підтвердження замовлення — після успішної оплати система відображає коротке повідомлення з подякою, номером замовлення та його

статусом. Користувач може повернутися на головну сторінку або перейти в особистий кабінет для відстеження подальшого оновлення статусу доставки (рис. 3.8).

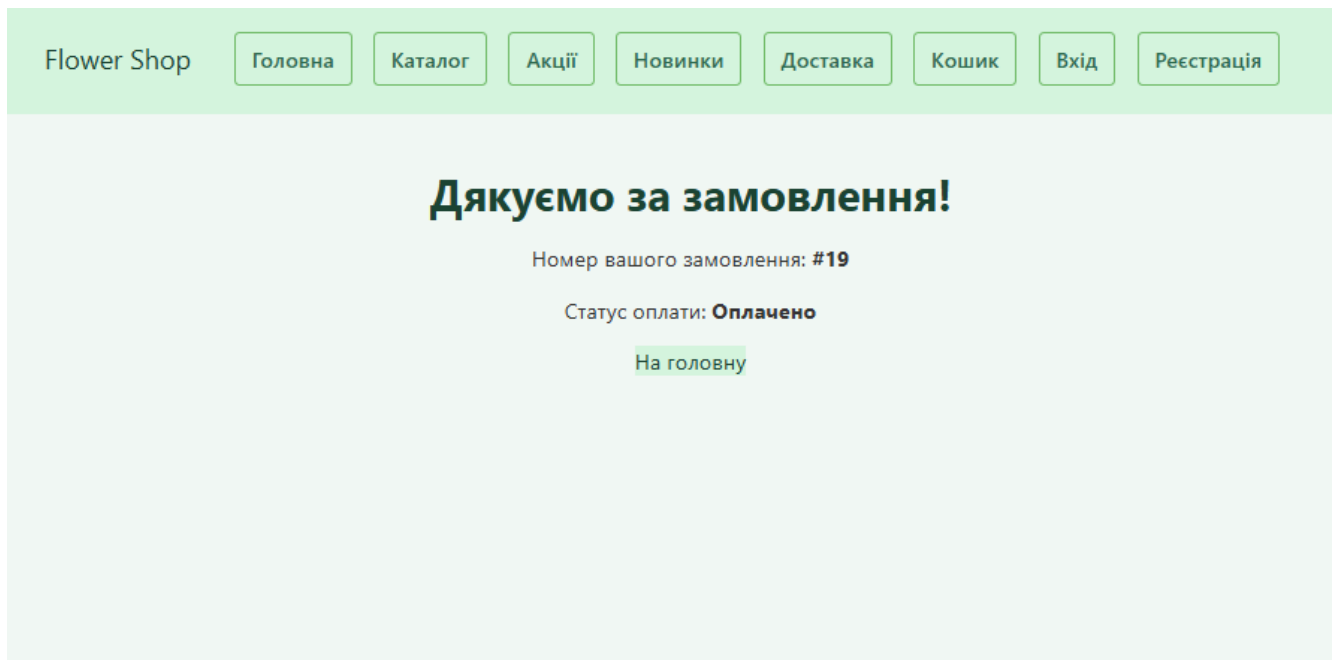


Рис.3.8 – Сторінка підтвердження замовлення

*Джерело: сформовано автором самостіно*

## ВИСНОВКИ

Упродовж роботи над дипломним проєктом було виконано весь цикл створення повноцінної вебсистеми для інтернет-квіткового магазину — від аналізу вимог до запуску готового рішення на локальному сервері. На початковому етапі було вивчено бізнес-логіку торгівлі квітами, визначено ключові користувацькі сценарії та оформлено їх у вигляді прецедентів і діаграм процесів.

Це дало змогу чітко окреслити необхідний функціонал: каталог товарів, кошик із багатокроковим оформленням замовлення, особистий кабінет користувача, адміністративну панель і систему повідомлень.

Після цього було спроектовано базу даних за допомогою ER-моделі в PyCharm з SQLite як рушієм зберігання. У схемі з'явилися таблиці Users, Flowers, Orders, OrderItem та ShippingAddress із відпрацьованими зовнішніми ключами, що гарантують цілісність зв'язків. Одразу ж реалізував Django-моделі на основі цієї структури: кожен клас моделі містить унікальні методи та властивості для зручності фільтрацій і валідації даних. Зокрема, модель Flowers підвантажує зображення, а Orders автоматично обчислює загальну суму замовлення на основі пов'язаних елементів кошика.

Для розробки бекенда, було налаштовано маршрути в файлі urls.py, створено контролери (views.py) із класовими та функціональними поданнями, де застосував міксини для обмеження доступу та реалізації авторизації. Авторизація та реєстрація побудовані на стандартній Django-системі, але доповнені хешуванням паролів за PBKDF2 і CSRF-захистом для кожної форми. Особливу увагу приділено механізму збереження стану кошика: дані успішно зберігаються або в сесії, або в таблиці CartItem, що забезпечує комфортне повернення до оформлення без втрати вибору.

Інтерфейс реалізуваний за допомогою поєднання Bootstrap-та власних стилів у CSS, адаптованих під пастельно-зелену палітру бренду Flower Shop. Тут HTML-шаблони Django тісно взаємодіють із JavaScript, який було застосовано винятково для поліпшення UX: показ підказок, динамічне відображення помилок у формах та

плавні переходи між етапами оформлення. З використанням Chart.js було додано у адмін-панель прості графіки продажів, що дають змогу миттєво оцінити поточні тренди.

У процесі розробки були написані юніт-тести для критичних методів моделі та інтеграційні тести для перевірки сценаріїв реєстрації, входу, оформлення замовлення. Завдяки цьому вдалося виявити й усунути низку дрібних багів ще на етапі розробки. Весь код зберігається у Git-репозиторії, що дозволило відстежувати історію змін і підтримувати чистоту основної гілки.

Нині система працює локально, адмін-панель повністю функціональна, каталог відображає всі композиції, а оформлення замовлення завершується коректною генерацією записів у базі й виведенням номера замовлення.

Перспективою подальшого розвитку є розгортання на хмарному сервері з використанням Docker-контейнерів, інтеграцію з платіжними шлюзами та підключенням сторонніх API доставки. Загалом реалізована платформа демонструє високу готовність до виробничого використання, забезпечує безпеку даних та відкриває двері для подальшої масштабованості під будь-які бізнес-потреби.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Education: From COVID-19 school closures to recovery. URL: <https://www.unesco.org/en/covid-19/education-response>. (дата звернення: 07.06.2025).
2. The Learning Tools Interoperability (LTI). URL: <https://www.ledtech.org/standards/lti>. (дата звернення: 07.06.2025).
3. Офіційна документація Django. URL: <https://docs.djangoproject.com/> (дата звернення: 12.06.2025).
4. Офіційна документація SQLite. URL: <https://www.sqlite.org/docs.html> (дата звернення: 12.06.2025).
5. Bootstrap Documentation. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 12.06.2025).
6. Python Documentation. URL: <https://docs.python.org/3/> (дата звернення: 12.06.2025).
7. JavaScript Documentation (MDN). URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 12.06.2025).
8. HTML5 Documentation (MDN). URL: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5> (дата звернення: 12.06.2025).
9. CSS Documentation (MDN). URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 12.06.2025).
10. Вікіпедія. AJAX. URL: <https://ru.wikipedia.org/wiki/AJAX> (дата звернення: 12.06.2025).
11. W3Schools: HTML, CSS, JavaScript Tutorials. URL: <https://www.w3schools.com/> (дата звернення: 12.06.2025).
12. Діаграма прецедентів. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_прецедентів](https://uk.wikipedia.org/wiki/Діаграма_прецедентів) (дата звернення: 12.06.2025).
13. Діаграма послідовності. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_послідовності](https://uk.wikipedia.org/wiki/Діаграма_послідовності) (дата звернення: 12.06.2025).

14. Що таке документ бізнес-вимог. URL: <https://www.ba.in.ua/2023/11/24/shho-take-dokument-biznes-vumog-business-requirements-document> (дата звернення: 12.06.2025).
15. Airbyte. SQLite vs MS SQL Server - Key Differences. URL: <https://airbyte.com/data-engineering-resources/sqlite-vs-sql-server> (дата звернення: 11.06.2025).
16. Моделі даних. Реляційна модель даних. URL: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/BD\\_2016\\_3/page5.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/BD_2016_3/page5.html) (дата звернення: 12.06.2025).
17. Лекція. Середовище баз даних. URL: [https://virt.ldubgd.edu.ua/pluginfile.php/38818/mod\\_resource/content/1/2.Середовище\\_баз\\_даних.pdf](https://virt.ldubgd.edu.ua/pluginfile.php/38818/mod_resource/content/1/2.Середовище_баз_даних.pdf) (дата звернення: 12.06.2025).
18. UX Design Principles. Nielsen Norman Group. URL: <https://www.nngroup.com/articles/ten-usability-heuristics> (дата звернення: 12.06.2025).
19. Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs> (дата звернення: 12.06.2025).
20. OpenVPN Documentation. URL: <https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4> (дата звернення: 12.06.2025).
21. Prometheus Monitoring System. URL: <https://prometheus.io/docs/introduction/overview> (дата звернення: 12.06.2025).
22. Figma Interface Design Tool. URL: <https://www.figma.com> (дата звернення: 12.06.2025).
23. DBeaver – Universal Database Tool. URL: <https://dbeaver.io> (дата звернення: 12.06.2025).
24. Draw.io (Diagrams.net) – Free Online Diagram Software. URL: <https://www.diagrams.net> (дата звернення: 12.06.2025).
25. Gartner Reports. e-Commerce Trends and Technologies. URL: <https://www.gartner.com> (дата звернення: 11.06.2025).
26. Smith J., Lee A. Scalability in e-commerce systems under peak load

conditions. International Journal of eCommerce. 2023. URL: <https://ijec.com/scalability-ecommerce> (дата звернення: 11.06.2025).

27. Сидоренко О., Петренко І. Безпека та інтеграція платіжних шлюзів у вебдодатках. Київ, 2024. URL: <https://techjournal.kiev.ua/payment-security-integration> (дата звернення: 10.06.2025).

28. Іващенко О., Ковальчук М. Інтуїтивна навігація як ключовий фактор успіху в інтернет-магазинах. Львів, 2023. URL: <https://biztech.lviv.ua/intuitive-navigation-ecommerce> (дата звернення: 09.06.2025).

29. XB Software. Online Platforms for e-Commerce: Key Problems and Smart Solutions. URL: <https://xbsoftware.com/blog/ecommerce-platforms> (дата звернення: 08.06.2025).

30. Михаліченко О. Використання сучасних цифрових технологій у бізнес-процесах. URL: <https://naurok.com.ua/vikoristannya-suchasnih-cifrovih-tehnologiy-v-osvitnomu-procesi-342096.html> (дата звернення: 12.06.2025).

## ДОДАТКИ

## Додаток А

## ДІАГРАМИ ПОСЛІДОВНОСТІ

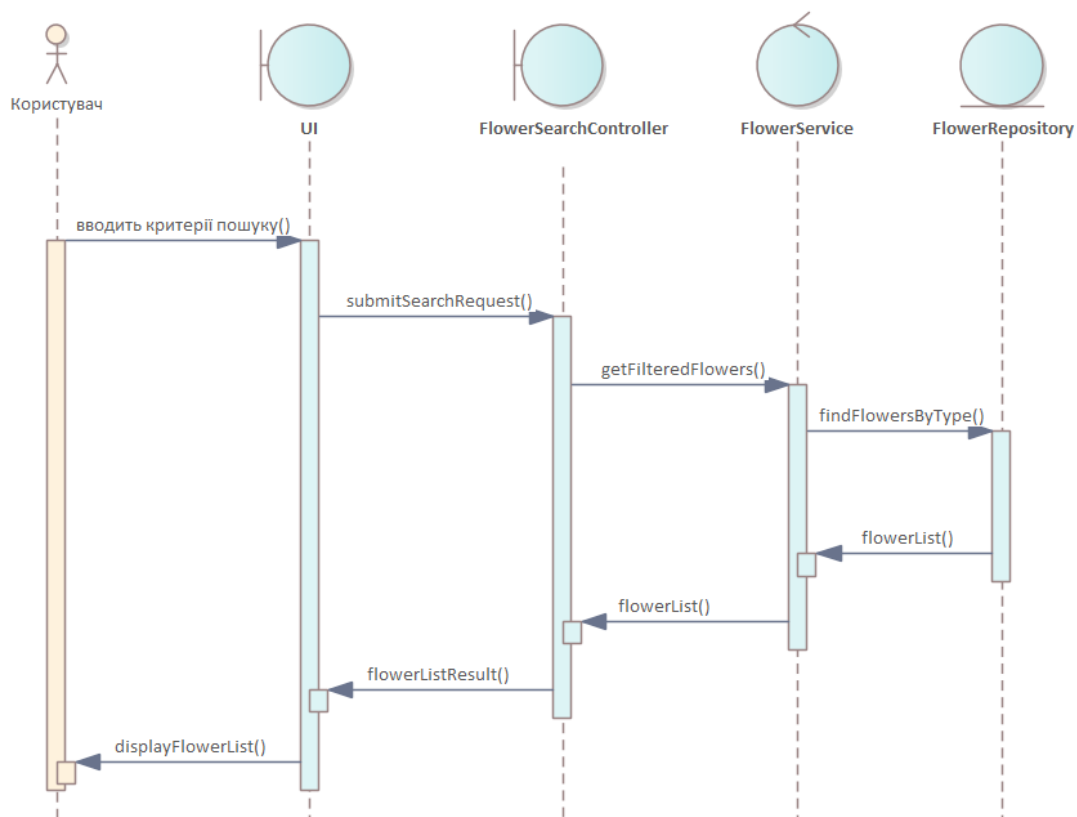


Рисунок А.1 – Діаграма послідовності процесу «Пошук та фільтрація квітів»

Джерело: сформовано автором самостіно

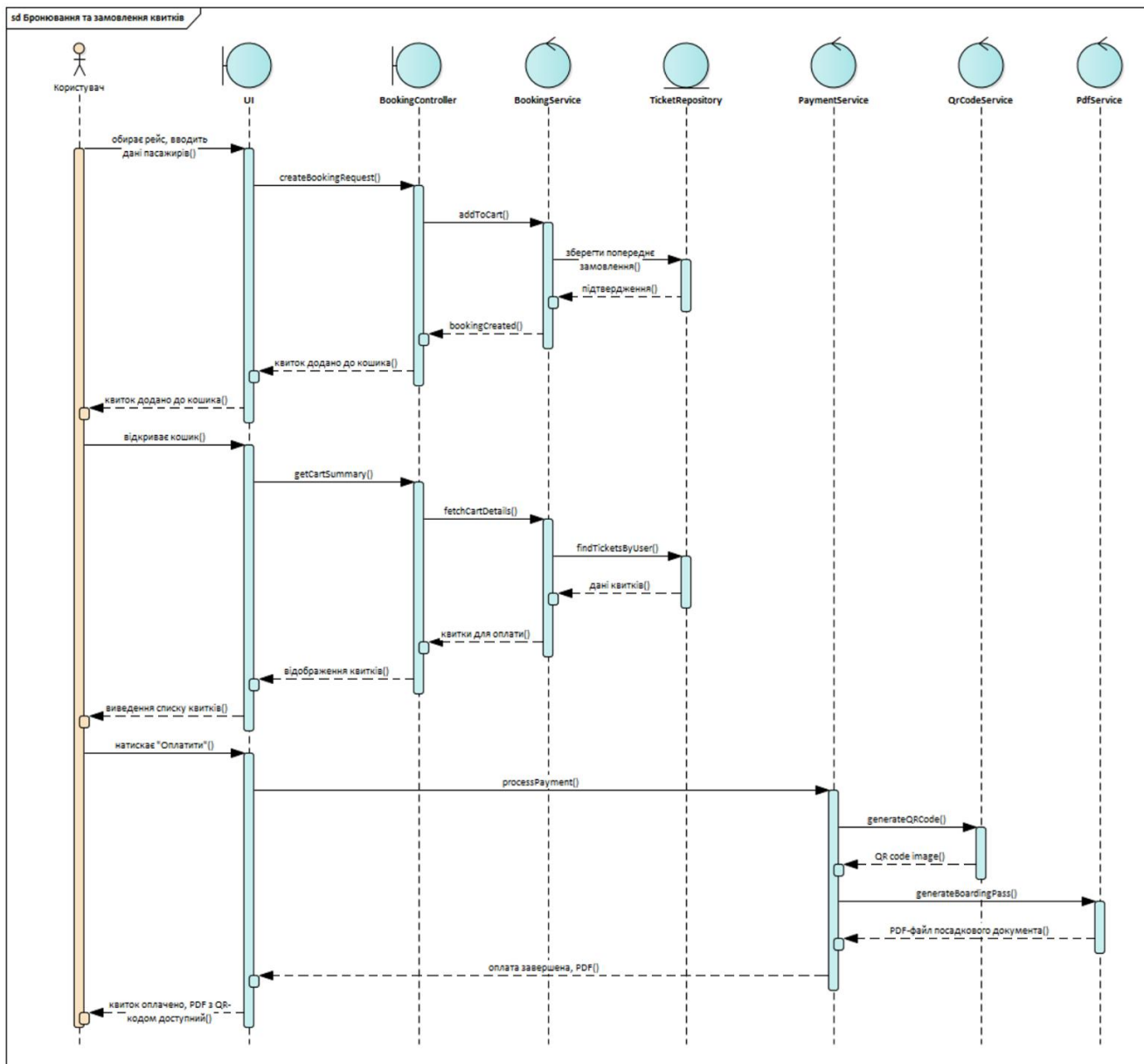


Рисунок А.2 – Діаграма послідовності процесу «Бронювання та замовлення квитків»

Джерело: сформовано автором самостіно

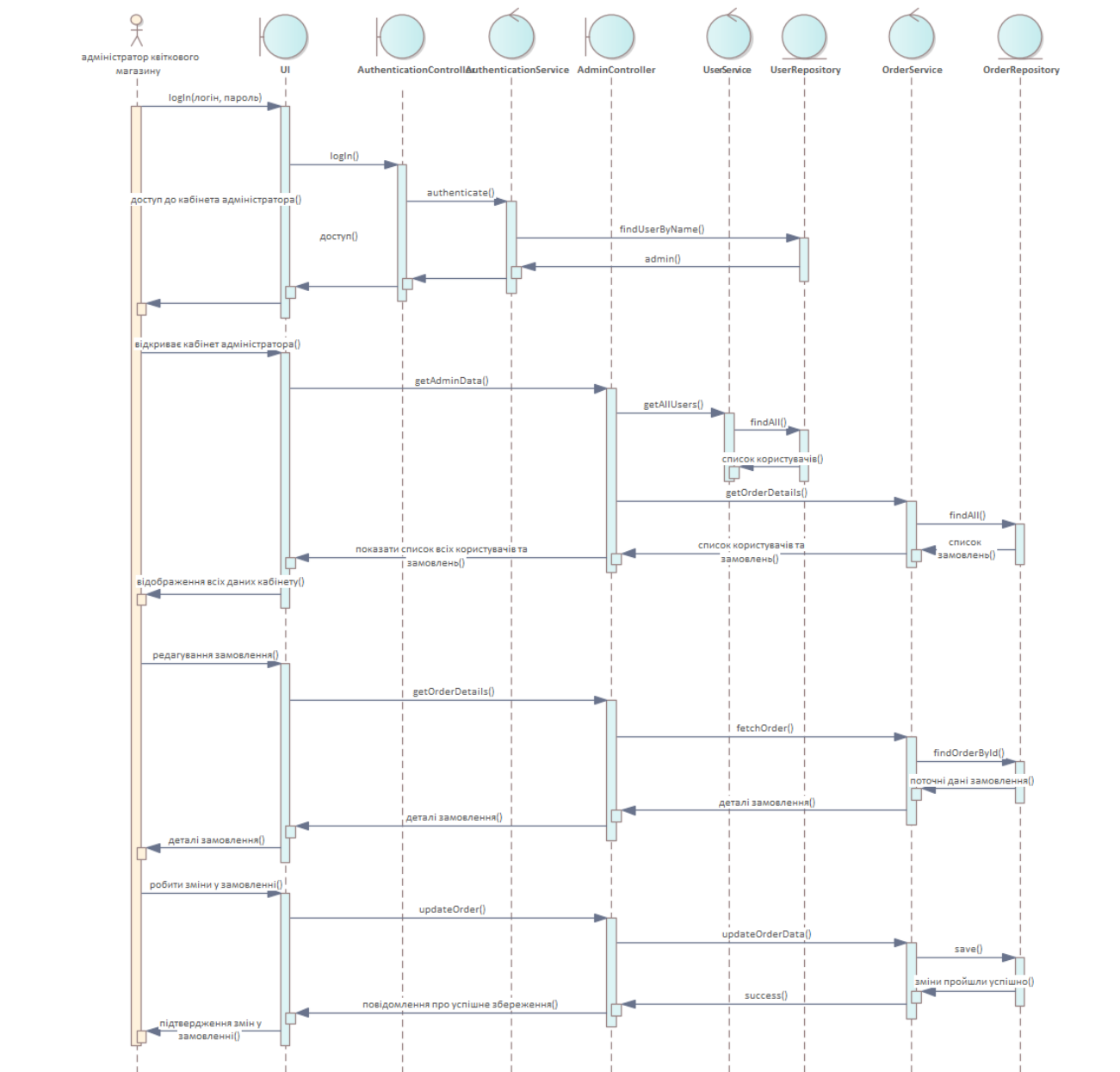


Рисунок А.3 – Діаграма послідовності процесу «Автентифікація адміністраторів та адміністрування»

Джерело: сформовано автором самостіно

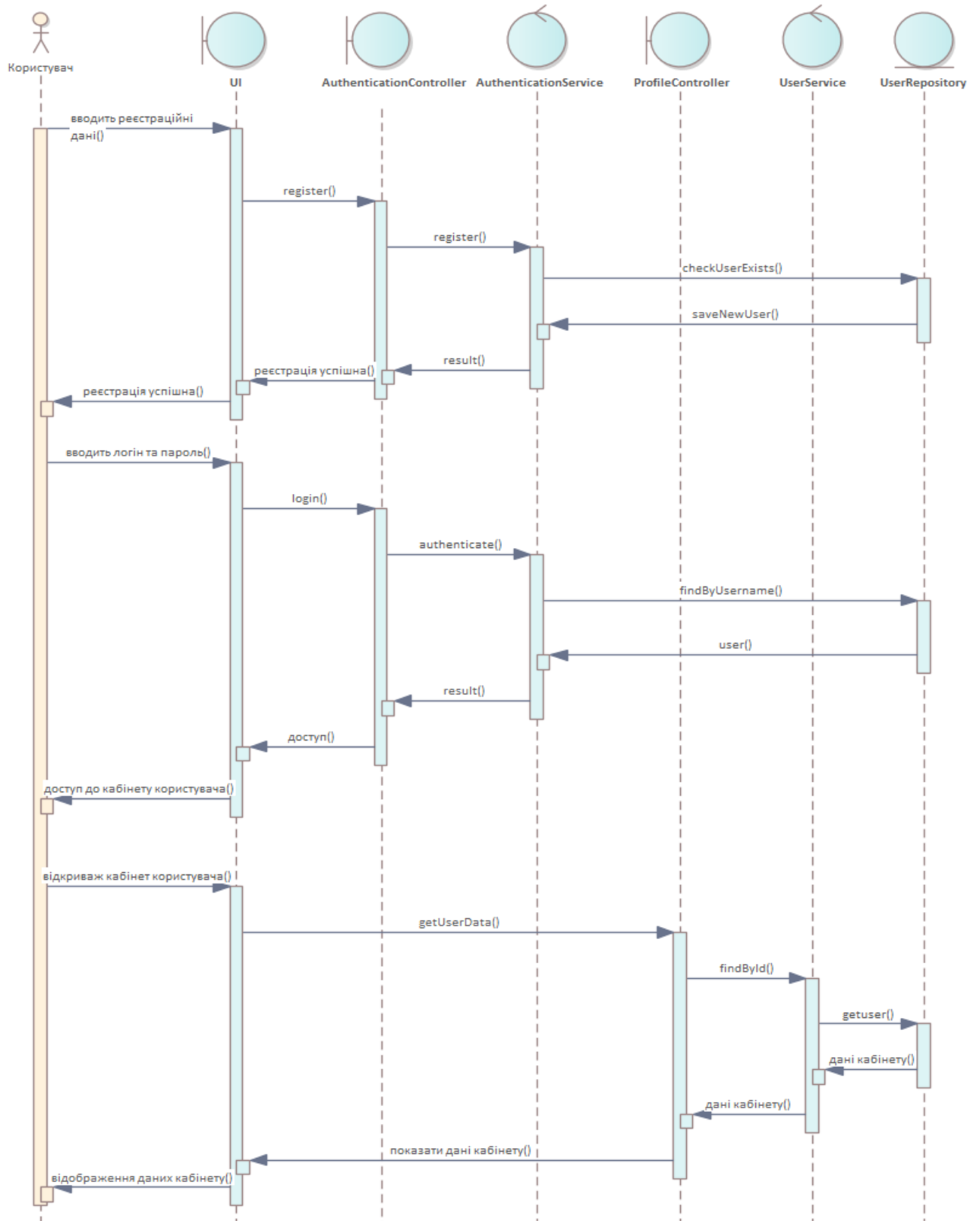


Рисунок А.4 – Діаграма послідовності процесу «Авторизації користувачів та особистий кабінет»

Джерело: сформовано автором самостіно

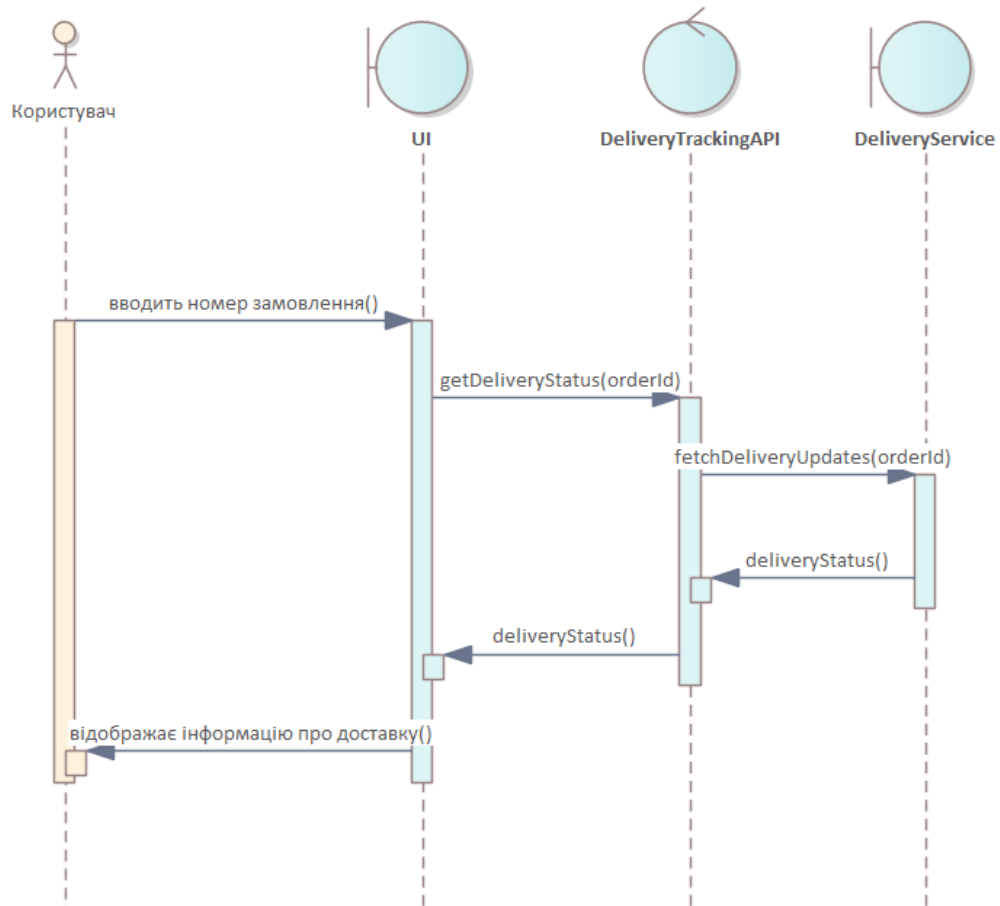


Рисунок А.5 – Діаграма послідовності процесу «перегляду інформації про доставку в системі автоматизації квіткового магазину»

*Джерело: сформовано автором самостіно*

## ДЕТАЛІЗОВАНІ СПЕЦИФІКАЦІЇ ВИМОГ

Item	Stereotype	Status	Difficulty	Priority
<input checked="" type="checkbox"/> Інформаційна система автоматизації квіткового магазину	Functional	Proposed	High	Medium
<input checked="" type="checkbox"/> Система пошуку квітів	Functional	Approved	Medium	Medium
<input checked="" type="checkbox"/> пошук квітів за категоріями	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> фільтрація результатів пошуку квітів	Functional	Approved	Low	Low

Рисунок Б.1 – Специфікація функціональних вимог «Система пошуку квітів»

*Джерело: сформовано автором самостіно*

<input checked="" type="checkbox"/> Система замовлення квітів	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> можливість вибору квітів	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> можливість додавання до кошику	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> можливість редагування кількості квітів у кошику	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> статус замовлення	Functional	Implemented	Low	Low

Рисунок Б.2 – Специфікація функціональних вимог «Система замовлення квітів»

*Джерело: сформовано автором самостіно*

<input checked="" type="checkbox"/> Система авторизації користувача	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> можливість входу зареєстрованого користувача	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> можливість реєстрації	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> доступ до кабінету користувача	Functional	Proposed	Medium	Medium

Рисунок Б.3 – Специфікація функціональних вимог «Система авторизації користувача»

*Джерело: сформовано автором самостіно*

<input checked="" type="checkbox"/> Система адміністрування	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> доступ до інформації щодо замовлень, клієнтів	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> керування авторизацією користувача	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> можливість редагування продукції, цін, назв	Functional	Proposed	Medium	Medium

Рисунок Б.4 – Специфікація функціональних вимог «Система адміністрування»

*Джерело: сформовано автором самостіно*

<input checked="" type="checkbox"/> Система оплати квітів	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> відображення інформації про замовлення перед оплатою	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> можливість оплати квітів, букеті	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> підтвердження замовлення після оплати	Functional	Proposed	Medium	Medium

Рисунок Б.5 – Специфікація функціональних вимог «Система оплати квітів»

*Джерело: сформовано автором самотіно*

Item	Stereotype	Status	Difficulty	Priority
<input checked="" type="checkbox"/> Інформаційна система автоматизації квіткового магазину	Functional	Proposed	High	Medium
<input checked="" type="checkbox"/> Інтерфейс користувача	Functional	Approved	Medium	Medium
<input checked="" type="checkbox"/> Адаптивний дизайн для коректного відображення на будь-якому екрані	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> Можливість змінювати тему оформлення інтерфейсу	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Інтуїтивно зрозуміла та легка для навігації система	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> Сумісність з різними пристроями	Functional	Approved	Low	Low

Рисунок Б.6 – Специфікація нефункціональних вимог «Інтерфейс користувача»

*Джерело: сформовано автором самотіно*

<input checked="" type="checkbox"/> Безпека	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Розмежування прав доступу між клієнтами та адміністраторами	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> Шифрування паролів користувачів у базі даних	Functional	Proposed	Medium	Medium

Рисунок Б.7 – Специфікація нефункціональних вимог «Безпека»

*Джерело: сформовано автором самотіно*

<input checked="" type="checkbox"/> Надійність	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Безперебійна доступність онлайн-сервісу	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Захист від втрати даних у разі помилок користувача	Functional	Proposed	Medium	Medium

Рисунок Б.8 – Специфікація нефункціональних вимог «Надійність»

*Джерело: сформовано автором самотіно*

<input checked="" type="checkbox"/> Швидкодія	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Автоматична синхронізація даних між підсистемами	Functional	Implemented	Low	Low
<input checked="" type="checkbox"/> Швидкість обробки замовлень та зміни статусів	Functional	Proposed	Medium	Medium

Рисунок Б.9 – Специфікація нефункціональних вимог «Швидкодія»

*Джерело: сформовано автором самостійно*

<input checked="" type="checkbox"/> Масштабованість	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Гнучка архітектура для інтеграції з платіжними системами, службами доставки тощо	Functional	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Підтримка збільшення кількості користувачів без зниження продуктивності	Functional	Proposed	Medium	Medium

Рисунок Б.10 – Специфікація нефункціональних вимог «Масштабованість»

*Джерело: сформовано автором самостійно*

## ФРАГМЕНТИ КОДУ ДЛЯ ВЕБЗАСТОСУНКУ

### index.html

```
{% extends 'base.html' %}
{% load static %}

{% block title %}Головна – Flower Shop{% endblock %}

{% block content %}
<div class="hero text-center">
  
  <div class="hero-text">
    <h1 class="display-5">Ласкаво просимо до Flower Shop</h1>
    <p class="hero-lead">
      Наша місія – дарувати радість і красу кожному завдяки неповторним букетам.
    </p>
    <a href="{% url 'catalog' %}" class="btn btn-primary btn-lg">Перейти до
каталогу</a>
  </div>
</div>
{% endblock %}
```

### catalog.html

```
{% extends 'base.html' %}
{% load static %}

{% block title %}Каталог – Flower Shop{% endblock %}
{% block content %}
<h1 class="mb-4 text-center">Каталог букетів</h1>

{% if flowers %}
  <div class="catalog-grid">
    {% for flower in flowers %}
      <div>
        <div class="card shadow-sm w-100 h-100">
          {% if flower.image %}
            <a href="{% url 'product_detail' flower.pk %}">
              
            </a>
          {% else %}
            <a href="{% url 'product_detail' flower.pk %}">
              
            </a>
          {% endif %}
        <div class="card-body d-flex flex-column">
          <h5 class="card-title text-center">{{ flower.name }}</h5>
          <p class="fw-bold text-center">{{ flower.price }} грн</p>
          <a href="{% url 'product_detail' flower.pk %}"
            class="mt-auto btn btn-primary w-100">
            Детальніше
          </a>
        </div>
      </div>
    {% endfor %}
  </div>
{% endif %}
```

```

        </div>
    </div>
</div>
{% endfor %}
</div>
{% else %}
    <p class="text-center">Каталог порожній. Додайте товари через адмінку.</p>
{% endif %}
{% endblock %}

```

## login.html

```

{% extends 'base.html' %}
{% load static %}

{% block title %}Вхід – Flower Shop{% endblock %}

{% block content %}
    <div class="row justify-content-center">
        <div class="col-md-4">
            <h2 class="mb-4 text-center">Вхід</h2>
            <form method="post" class="login-form">
                {% csrf_token %}
                {{ form.as_p }}
                <button type="submit" class="btn btn-success w-100">Увійти</button>
            </form>
        </div>
    </div>
{% endblock %}

```

## register.html

```

{% extends 'base.html' %}
{% load static %}

{% block title %}Реєстрація – Flower Shop{% endblock %}

{% block content %}
    <div class="row justify-content-center">
        <div class="col-md-4">
            <h2 class="mb-4">Реєстрація</h2>
            <form method="post">
                {% csrf_token %}
                {{ form.as_p }}
                <button type="submit" class="btn btn-primary">Зареєструватися</button>
            </form>
        </div>
    </div>
{% endblock %}

```

## base.html

```

{% load static %}
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

```

```

<title>{% block title %}Flower Shop{% endblock %}</title>

<!-- Bootstrap CSS (CDN) -->
<link
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.4.1/dist/css/bootstrap.min.css"
  rel="stylesheet"
>
<!-- Власні стилі -->
<link href="{% static 'css/style.css' %}" rel="stylesheet">
</head>
<body class="d-flex flex-column min-vh-100">

<!-- HEADER: навігаційна панель -->
<nav class="navbar navbar-expand-lg navbar-light bg-white shadow-sm">
  <div class="container">
    <a class="navbar-brand fw-bold" href="{% url 'index' %}">Flower Shop</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
      data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item"><a class="nav-link" href="{% url 'index'
%}">Головна</a></li>
        <li class="nav-item"><a class="nav-link" href="{% url 'catalog'
%}">Каталог</a></li>
        <li class="nav-item"><a class="nav-link" href="{% url 'promotions'
%}">Акції</a></li>
        <li class="nav-item"><a class="nav-link" href="{% url 'news'
%}">Новинки</a></li>
        <li class="nav-item"><a class="nav-link" href="{% url 'delivery'
%}">Доставка</a></li>

        <!-- Кошик -->
        <li class="nav-item">
          <a class="nav-link" href="{% url 'cart' %}">Кошик</a>
        </li>

        <!-- завжди два пункти: Вхід і Реєстрація -->
        <li class="nav-item">
          <a class="nav-link" href="{% url 'login' %}">Вхід</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{% url 'register' %}">Реєстрація</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

<!-- CONTENT: головний контент сторінки -->
<main class="container my-4 flex-grow-1">
  {% block content %}{% endblock %}
</main>

<!-- FOOTER -->
<footer class="bg-light py-3 mt-auto">
  <div class="container text-center">
    © 2025 Flower Shop
  </div>
</footer>

<!-- Bootstrap JS (CDN) -->
<script

```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.4.1/dist/js/bootstrap.bundle.min.js"  
>  
</script>  
<!-- Власні скрипти (за потреби) -->  
<script src="{% static 'js/main.js' %}"></script>  
</body>  
</html>
```

## КОРОТКИЙ ЗВІТ ПОДІБНОСТІ



Дата звіту 6/22/2025

Дата редагування ---



Звіт не був оцінений

## Звіт подібності

## метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

Проектування інформаційної системи для автоматизації процесів квіткового магазину

Автор

Науковий керівник / Експерт

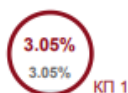
Шевченко Олександр Олександрович Артемчук В.О.

підрозділ

кафедра інформаційних систем в економіці

## Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



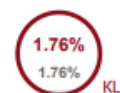
25

Довжина фрази для коефіцієнта подібності 2



9184

Кількість слів



73151

Кількість символів

## Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		2
Білі знаки		0
Парафрази (SmartMarks)		6

## Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

## 10 найдовших фраз

Копір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download">https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download</a>	44 0.48 %