



**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
Факультет інформаційних технологій**

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**  
**комп'ютерних наук**  
\_\_\_\_\_ (назва кафедри)

\_\_\_\_\_ **Голуб Б.Л.**  
(науковий ступінь, вчене звання) (підпис) (ПІБ)  
“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**З А В Д А Н Н Я**  
**на виконання бакалаврської кваліфікаційної роботи студенту**  
**Панчак Олександр Володимирович**

Спеціальність 122 – «Комп'ютерні науки»

1. Тема бакалаврської кваліфікаційної роботи «Розробка платформи електронної комерції для дистрибуції цифрового контенту» затверджена наказом ректора НУБІП України від 02.04.2025 № 526с

2. Термін подання завершеної роботи на кафедру \_\_\_\_\_  
(рік, місяць, число)

3. Вихідні дані до роботи: формування товарів (цифрових копій медіаресурсів) в електронному вигляді, надання права на володіння медіаресурсом.

4. Перелік питань, що розглядаються:

- Аналіз проблемної області
- Моделювання предметної області
- Проектування програмної системи
- Впровадження та експлуатація системи

Дата видачі завдання “ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

Керівник бакалаврської кваліфікаційної роботи \_\_\_\_\_ / **Саяпін С.П.** /  
( підпис ) (прізвище та ініціали)

Завдання прийняв до виконання: \_\_\_\_\_ / **Панчак О.В.** /  
( підпис ) (прізвище та ініціали)

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....  | 2  |
| ВСТУП .....   | 3  |
| 1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ.....                                      | 5  |
| 1.1 Опис предметної області .....                                     | 5  |
| 1.2 Огляд існуючих рішень .....                                       | 11 |
| 1.3 Постановка завдання.....  | 16 |
| 1.4 Функціональні та нефункціональні вимоги .....                     | 18 |
| 1.5 Вимоги до інтерфейсу користувача .....                            | 20 |
| 2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ .....                                | 23 |
| 2.1 Об'єктне та функціональне моделювання.....                        | 23 |
| 2.2 Абстракції предметної області .....                               | 32 |
| 2.3 Діаграма класів .....   | 34 |
| 3 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ .....                               | 37 |
| 3.1 Логічна модель даних .....  | 37 |
| 3.2 Вибір системи управління базою даних та її реалізація .....       | 41 |
| 3.3 Архітектура програмного забезпечення .....                        | 46 |
| 3.4 Організаційна структура програмного забезпечення.....             | 50 |
| 3.5 Вибір інструментарію для створення програмного забезпечення ..... | 52 |
| 4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ.....                           | 55 |
| 4.1 Вимоги до апаратного та програмного забезпечення .....            | 55 |
| 4.2 Тестування системи .....  | 57 |
| ВИСНОВКИ.....   | 67 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....                                       | 69 |
| ДОДАТОК А.....  | 72 |
| ДОДАТОК Б .....   | 74 |

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

|      |   |                                   |
|------|---|-----------------------------------|
| БД   | – | База даних                        |
| ОС   | – | Операційна система                |
| ПЗ   | – | Програмне забезпечення            |
| СКБД | – | Система керування баз даних       |
| API  | – | Application Programming Interface |
| Bash | – | Bourne Again SHell                |
| MVVM | – | Model-View-ViewModel              |
| SQL  | – | Structed Query Language           |
| UI   | – | User Interface                    |

## ВСТУП

Цифрова економіка швидко розвивається, змінюючи спосіб споживачів отримати доступ до медіа-продуктів і придбати їх. Ігри, книги та фільми зараз переважно розповсюджуються онлайн, що усуває потребу у фізичних копіях і забезпечує миттєвий доступ до вмісту. Ця зміна призвела до зростання значення спеціалізованих платформ електронної комерції, призначених для цифрової дистрибуції. На відміну від традиційних онлайн-магазинів, які переважно обслуговують фізичні товари, платформи, що продають цифрові продукти, вимагають додаткових функцій, таких як безпечні транзакції, ефективна доставка вмісту та доступ на основі підписки. Актуальність цієї розробки в тому, що багатьом існуючим рішенням бракує надійних заходів безпеки, ефективного керування цифровими правами (DRM) або гнучких моделей монетизації. Ці виклики підкреслюють необхідність розробки сучасної, безпечної та масштабованої системи електронної комерції для розповсюдження цифрового контенту.

Програмне рішення, запропоноване в цій роботі, розроблено з метою покращення існуючих систем продажу та розповсюдження медіаконтенту та для вирішення проблем доступу до придбаних цифрових копій, надаючи веб-платформу, де користувачі можуть купувати, завантажувати та підписуватися на цифрові медіа-продукти. Система надає пріоритет користувацькому досвіду, пропонуючи плавний процес перегляду та купівлі, водночас забезпечуючи високі стандарти безпеки через механізми автентифікації, зашифровані платежі та інтеграцію DRM. Створена з використанням PHP і фреймворку Laravel із MySQL як системою керування базами даних, платформа використовує сучасні технології веб-розробки для досягнення оптимальної продуктивності та масштабованості. Крім того, він включає безпечні платіжні шлюзи, такі як PayPal або Stripe, для полегшення транзакцій,

підтримуючи як одноразові покупки, так і моделі підписки для доступу до цифрового вмісту.

Процес розробки дотримується структурованого підходу розробки програмного забезпечення, починаючи з аналізу потреб ринку та існуючих рішень. Потім створюється детальний архітектурний проєкт, який охоплює структури бази даних, серверну логіку та зовнішні інтерфейси. Етап впровадження включає кодування, інтеграцію платіжних систем і застосування заходів безпеки для захисту даних користувачів і запобігання несанкціонованому обміну вмістом. Комплексне тестування, включаючи модульне тестування, оцінку продуктивності та перевірки безпеки, гарантує надійність і стабільність платформи перед розгортанням.

Щоб підтвердити ефективність розробленого рішення, результати цього дослідження будуть представлені в академічному та професійному середовищі. Заплановані заходи включають участь у конференціях, публікацію наукових статей і обговорення з професіоналами галузі для збору відгуків і подальшого вдосконалення платформи. Ці зусилля не лише підтвердять практичну значущість проєкту, але й внесуть внесок у ширшу сферу електронної комерції та поширення цифрового контенту.

# 1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ

## 1.1 Опис предметної області

Швидкий розвиток цифрових технологій докорінно змінив спосіб розповсюдження та споживання медіа-продуктів. Традиційний фізичний продаж книг, фільмів та ігор поступився місцем платформам цифрового розповсюдження, які забезпечують миттєвий доступ до вмісту. Ця трансформація призвела до зростання попиту на ефективні, безпечні та зручні рішення для електронної комерції, спеціально розроблені для продажу цифрових медіа. Однак, незважаючи на зростаючу залежність від таких платформ, у галузі залишається ряд проблем, які вимагають інноваційних рішень для покращення розповсюдження цифрового контенту.

Одним із ключових питань цифрової комерції є забезпечення безпечних і надійних транзакцій. На відміну від фізичних товарів, цифрові продукти можна легко скопіювати та розповсюдити без належного дозволу. Це робить захист інтелектуальної власності та керування цифровими правами (DRM) важливими для платформ електронної комерції, які обробляють медіаконтент. Без надійних систем DRM творці та розповсюджувачі контенту ризикують зазнати фінансових втрат через піратство та несанкціоноване поширення. Крім того, безпечна обробка платежів має вирішальне значення, оскільки цифрові магазини повинні захищати конфіденційні фінансові дані користувачів від шахрайства та кіберзагроз. Багато існуючих платформ мають проблеми з інтеграцією як безпеки, так і простоти використання, що призводить до скомпрометованої довіри користувачів [3].

Інша проблема полягає в оптимізації взаємодії з користувачем, зберігаючи гнучкість бізнесу. Платформи цифрового контенту повинні забезпечувати інтуїтивно зрозумілий і безперебійний процес купівлі, дозволяючи користувачам переглядати, купувати та отримувати доступ до продуктів без зусиль. Такі функції, як миттєве завантаження, послуги на основі

підписки та персоналізовані рекомендації, покращують залучення та утримання клієнтів. Однак багато платформ не можуть ефективно реалізувати ці функції, що призводить до громіздкої навігації, повільної доставки вмісту або обмеження варіантів оплати. Крім того, підприємствам потрібні масштабовані рішення, які можуть адаптувати різні моделі ціноутворення, рекламні акції та засоби контролю доступу, не вимагаючи значних технічних модифікацій.

Однією з найбільш відомих платформ у сфері цифрової дистрибуції є Steam, розроблена компанією Valve. Вона пропонує користувачам широкий вибір ігор, автоматичні оновлення, систему досягнень, оглядів і соціальні функції. Steam створила цілу екосистему навколо ігрового контенту, однак зосереджена виключно на цьому сегменті, що обмежує її універсальність. Крім того, платформа стягує високі комісійні з розробників, що може бути стримуючим фактором для нових учасників ринку.

Іншим прикладом є Google Play — універсальний маркетплейс для Android-пристроїв, який охоплює не лише ігри, а й книги, музику, фільми та додатки. Ця платформа глибоко інтегрована в мобільну екосистему Google, забезпечує автоматичні оновлення та аналітичні інструменти для розробників. Водночас, через надвисоку конкуренцію і суворі вимоги до модерації, Google Play не завжди є комфортним середовищем для нових проєктів, а можливості кастомізації інтерфейсу є доволі обмеженими.

Серед молодших конкурентів варто згадати Epic Games Store, який активно просуває свою платформу за рахунок зменшеної комісії (лише 12% порівняно зі стандартними 30%) та ексклюзивних релізів. Хоча за функціональністю він поступається Steam, простота використання та прозорі умови для розробників роблять його привабливим варіантом. Проте, як і Steam, Epic Games Store зосереджений лише на ігровому контенті.

Аналіз ринку також показує прогалину в доступності та налаштуваннях. Багато існуючих цифрових ринків або надто загальні, обслуговуючи широкий спектр типів медіа без спеціалізації, або дуже

обмежені, обмежуючи можливість продавців налаштовувати свої пропозиції. Добре розроблена платформа електронної комерції для цифрового контенту повинна збалансувати гнучкість для постачальників, зберігаючи послідовну та надійну взаємодію з користувачем. Для цього потрібна модульна архітектура, яка дозволяє компаніям інтегрувати різні стратегії монетизації, від одноразових покупок до багаторівневих підписок і пакетних пропозицій.

Продуктивність і масштабованість є ключовими викликами для платформ цифрової дистрибуції, особливо в умовах високих навантажень, великих обсягів трафіку та необхідності обробки транзакцій у реальному часі. Без належної інфраструктури платформи ризикують зіткнутися з повільним завантаженням, невдалими транзакціями та збоями системи під час пікових навантажень.

Впровадження механізмів кешування, мереж доставки вмісту (CDN) та оптимізованих запитів до бази даних має вирішальне значення для забезпечення безперебійної взаємодії з користувачем. Наприклад, використання CDN може зменшити затримку доставки контенту на 80–90%, скорочуючи середній час відгуку з 135 мс до 13 мс у Європі та з 246 мс до 34 мс у Південній Америці.

Оптимізація запитів до бази даних також є критичною. Згідно з дослідженнями, впровадження індексів може підвищити швидкість виконання запитів у 1000 разів, а оптимізація запитів може зменшити час виконання на 50%.

Кешування часто використовуваних даних у пам'яті може зменшити час відповіді на запити на 80%, що особливо важливо для платформ з великою кількістю одночасних користувачів.

Таким чином, для забезпечення високої продуктивності та масштабованості платформ цифрової дистрибуції необхідно впроваджувати сучасні технології кешування, використовувати CDN та оптимізувати запити до баз даних. Ці заходи дозволяють забезпечити стабільну роботу системи навіть під час пікових навантажень та покращити взаємодію з користувачем.

Сектор електронної комерції цифрового контенту постійно розвивається, вимагаючи від платформ адаптації до нових технологій і ринкових тенденцій. Хмарне сховище, рекомендації на основі штучного інтелекту та блокчейн для безпечних транзакцій є одними з останніх інновацій, які формують галузь. Хоча ці технології пропонують потенційні рішення для існуючих проблем, їх впровадження вимагає ретельного планування для забезпечення сумісності, ефективності та економічності [3].

Галузь розповсюдження цифрового контенту стикається з кількома проблемами, включаючи ризики безпеки, оптимізацію взаємодії з користувачем, проблеми масштабованості та потребу в гнучких моделях монетизації. Розробка вдосконаленої платформи електронної комерції, розробленої спеціально для цифрових медіа, має вирішити ці проблеми шляхом інтеграції надійних функцій безпеки, інтуїтивно зрозумілого інтерфейсу, масштабованої інфраструктури та адаптованих бізнес-моделей. Це дослідження має на меті усунути ці прогалини шляхом створення комплексного рішення, яке підвищує як задоволеність користувачів, так і прибутковість бізнесу, одночасно забезпечуючи безпеку вмісту та доступність.

Швидка цифровізація споживання медіаконтенту докорінно змінила те, як користувачі отримують доступ і купують контент. Платформи цифрового контенту замінили традиційні звичайні магазини, пропонуючи миттєвий доступ до медіа-продуктів, таких як ігри, книги та фільми. У результаті значно зріс попит на платформи електронної комерції, спеціально розроблені для розповсюдження цифрового контенту. Ці платформи не лише спрощують продаж і доставку цифрових товарів, але й включають заходи безпеки, моделі ліцензування та системи керування клієнтами, щоб забезпечити безперебійний процес транзакцій як для продавців, так і для покупців.

На відміну від традиційних платформ електронної комерції, які зосереджені на продажу фізичних продуктів, системи розповсюдження

цифрового контенту вимагають принципово іншого підходу. Процес купівлі повинен дозволяти користувачам отримати свій вміст одразу після оплати, усуваючи потребу в доставці та логістиці. Крім того, ці платформи повинні підтримувати різні бізнес-моделі, включаючи прямі покупки, підписки та варіанти оренди. Гнучкість стратегій монетизації відіграє вирішальну роль в успіху таких платформ, дозволяючи творцям і розповсюджувачам контенту вибирати найкращий підхід до своїх продуктів.

Безпека є критично важливим аспектом розповсюдження цифрового контенту. Оскільки цифрові продукти можна легко копіювати та розповсюджувати без авторизації, для захисту інтелектуальної власності необхідне впровадження Digital Rights Management (DRM). Технології DRM регулюють доступ до цифрового вмісту, гарантуючи, що лише авторизовані користувачі можуть завантажувати або транслювати придбані медіафайли. Безпечна обробка платежів, шифрування даних і механізми запобігання шахрайству також є важливими компонентами надійної системи електронної торгівлі цифровими товарами.

Взаємодія з користувачем відіграє ключову роль в ефективності платформ цифрового контенту. Добре продуманий інтерфейс має дозволяти користувачам швидко шукати та купувати вміст, керувати своєю цифровою бібліотекою та отримувати доступ до своїх покупок із кількох пристроїв. Персоналізовані рекомендації на основі вподобань користувача та історії веб-перегляду можуть посилити залучення та збільшити продажі. Крім того, повна інтеграція зі сторонніми службами, такими як хмарне сховище, потокові платформи та соціальні мережі, може розширити охоплення та зручність використання платформи.

Технічна архітектура такої системи зазвичай складається з кількох ключових компонентів. Сервер, створений за допомогою PHP з інфраструктурою Laravel, обробляє бізнес-логіку, автентифікацію користувачів, обробку платежів і DRM. База даних, що працює на основі MySQL, зберігає інформацію про користувачів, історію покупок, метадані

продукту та деталі ліцензії. Інтерфейс, розроблений з використанням HTML, CSS і JavaScript, забезпечує інтуїтивно зрозумілий і адаптивний інтерфейс для клієнтів. Крім того, сторонні API, такі як платіжні шлюзи (PayPal, Stripe) і постачальники хмарних сховищ, сприяють безпечним транзакціям і доставці вмісту.

Масштабованість і оптимізація продуктивності є життєво важливими міркуваннями при розробці платформи електронної комерції для цифрового вмісту. Система повинна бути здатна обробляти великі обсяги трафіку, особливо під час рекламних кампаній або випуску нових продуктів. Впровадження механізмів кешування, мереж доставки вмісту (CDN) і асинхронної обробки для завантажень великих файлів забезпечує безперебійну роботу користувача навіть за високого навантаження.

У досліджуваній предметній області можна виділити такі основні бізнес-процеси:

1. реєстрація та авторизація користувачів;
2. перегляд та пошук контенту;
3. додавання товарів у кошик;
4. оформлення замовлення та оплата;
5. завантаження та доступ до контенту;
6. управління підписками;
7. рекомендації та персоналізація;
8. управління обліковим записом користувача.

Індустрія електронної комерції цифрового контенту продовжує розвиватися з появою таких тенденцій, як рекомендації на основі штучного інтелекту, захист авторських прав на основі блокчейна та децентралізовані мережі розподілу, які формують майбутнє цифрової комерції. Оскільки споживачі все більше переходять на онлайн-покупки, компанії повинні адаптуватися, пропонуючи безпечні, ефективні та привабливі платформи, які задовольняють їхні потреби.

## 1.2 Огляд існуючих рішень

На ринку розповсюдження цифрового контенту домінують кілька визнаних платформ електронної комерції, які пропонують користувачам доступ до медіа-продуктів, таких як ігри, книги та фільми. Ці платформи відрізняються за своїм підходом, моделями ціноутворення, механізмами захисту вмісту та користувацьким досвідом. Аналіз існуючих рішень дає змогу зрозуміти галузеві стандарти, загальні проблеми та області, які потрібно вдосконалити під час розробки нової системи.

Steam, розроблений Valve Corporation і запущений у 2003 році, зарекомендував себе як провідна цифрова платформа для розповсюдження відеоігор, програмного забезпечення та іншого цифрового контенту. З роками він перетворився на комплексну екосистему, яка надає розробникам ігор потужні інструменти розповсюдження, стратегії монетизації та функції залучення спільноти. Маючи мільйони користувачів у всьому світі, Steam продовжує формувати ігрову індустрію, підтримуючи різні бізнес-моделі, включаючи прямі покупки, внутрішньоігрові транзакції та послуги на основі підписки [1].

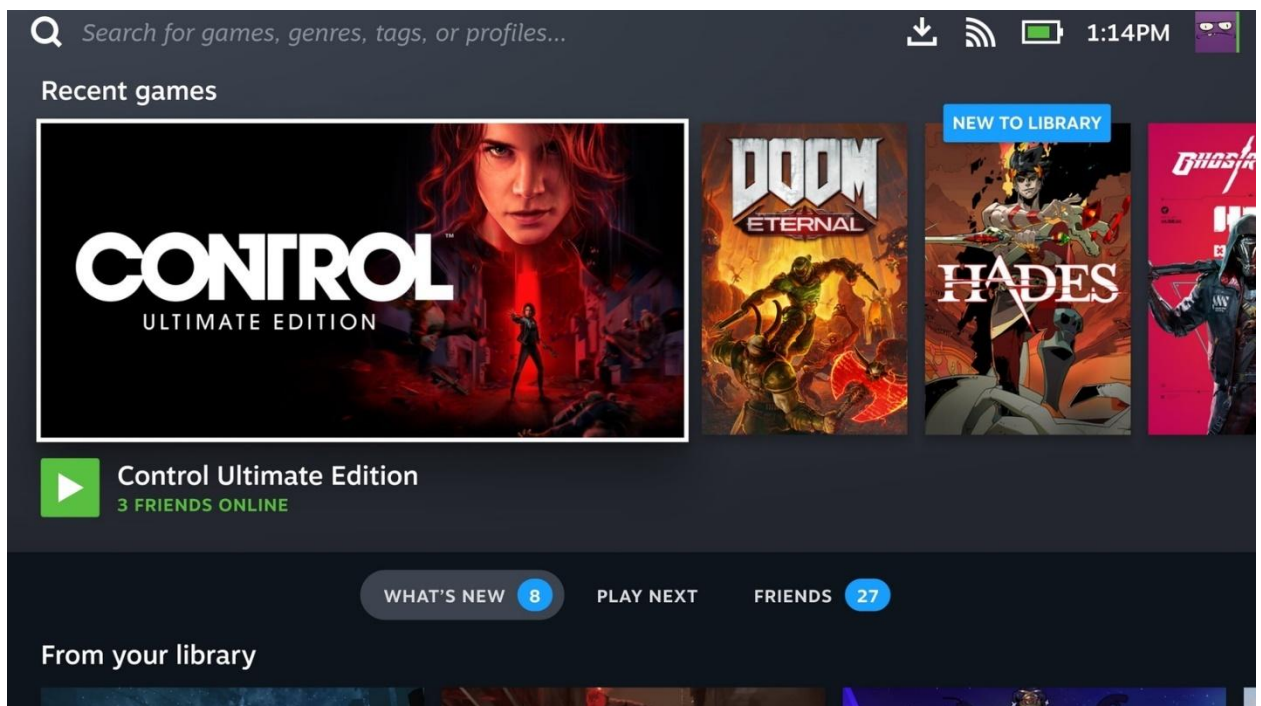


Рис. 1 Сервіс “Steam”

Однією з головних переваг Steam є система цифрової бібліотеки, яка дозволяє користувачам купувати та миттєво завантажувати ігри. На відміну від фізичних копій, цифрові покупки зберігаються в обліковому записі користувача, що забезпечує доступ із кількох пристроїв. Автоматичні оновлення гарантують, що ігри залишаються актуальними без ручного втручання. Крім того, Steam містить власну систему керування цифровими правами (DRM), яка допомагає захистити від піратства. Хоча обмеження DRM вимагають автентифікації через обліковий запис користувача Steam, для певних ігор доступний офлайн-режим, що дозволяє гравцям насолоджуватися своїми покупками без постійного підключення до Інтернету [1].

Платформа пропонує кілька варіантів монетизації, включаючи одноразові покупки, мікротранзакції та послуги за підпискою. Унікальною функцією є програма раннього доступу Steam, яка дозволяє розробникам випускати незавершені версії своїх ігор, дозволяючи гравцям тестувати та залишати відгуки перед офіційним запуском. Valve отримує частку доходу від усіх продажів, зазвичай близько 30%, хоча для найбільш продаваних ігор були внесені коригування, щоб залучити великих видавців.

Окрім вітрини, Steam сприяє активній соціальній та ігровій спільноті. Його екосистема включає вбудований список друзів, функції обміну повідомленнями та голосового чату, а також майстерню Steam, де гравці можуть створювати та ділитися модифікаціями, щоб продовжити термін служби гри. Користувачі також можуть залучатися за допомогою досягнень, колекційних карток і обміну предметами в грі на ринку спільноти Steam. Крім того, платформа містить оглядові та дискусійні форуми, де гравці обмінюються думками та рекомендаціями.

Іншим ключовим аспектом успіху Steam є його акцент на сумісності між платформами та хмарному сховищі. Функція Steam Cloud дозволяє гравцям синхронізувати свій прогрес на різних пристроях, забезпечуючи безперебійну гру. Платформа підтримує Windows, macOS, Linux і Steam Deck, що розширює її доступність. Крім того, ринок спільноти Steam надає

користувачам простір для торгівлі та продажу внутрішньоігрових предметів, створюючи додатковий потік доходу як для гравців, так і для розробників.

Незважаючи на домінування в галузі, Steam стикається з кількома проблемами. Високі комісії залишаються предметом суперечок серед розробників, оскільки Valve зазвичай займає значний відсоток продажів. Крім того, хоча захист DRM підвищує безпеку, він може обмежити гнучкість володіння грою. Зростаюча конкуренція з боку альтернативних платформ, таких як Epic Games Store, GOG і Microsoft Store, продовжує спонукати Steam до інновацій та адаптації.

Загалом Steam залишається лідером у розповсюдженні цифрових ігор, поєднуючи величезний каталог ігор із розширеними соціальними функціями та функціями безпеки. Його постійна еволюція відображає мінливі потреби як розробників, так і гравців, забезпечуючи його постійну актуальність на цифровому ринку.

Розглянемо інше програмне рішення на ринку.

Apple Books, раніше відома як iBooks, — це цифрова платформа для читання, розроблена Apple Inc., яка дозволяє користувачам купувати, завантажувати та читати широкий спектр електронних книг, аудіокниг та інших цифрових публікацій. Запущений у 2010 році як частина екосистеми Apple, Apple Books легко інтегрується з іншими службами та пристроями Apple, забезпечуючи читачам зручний досвід [2].

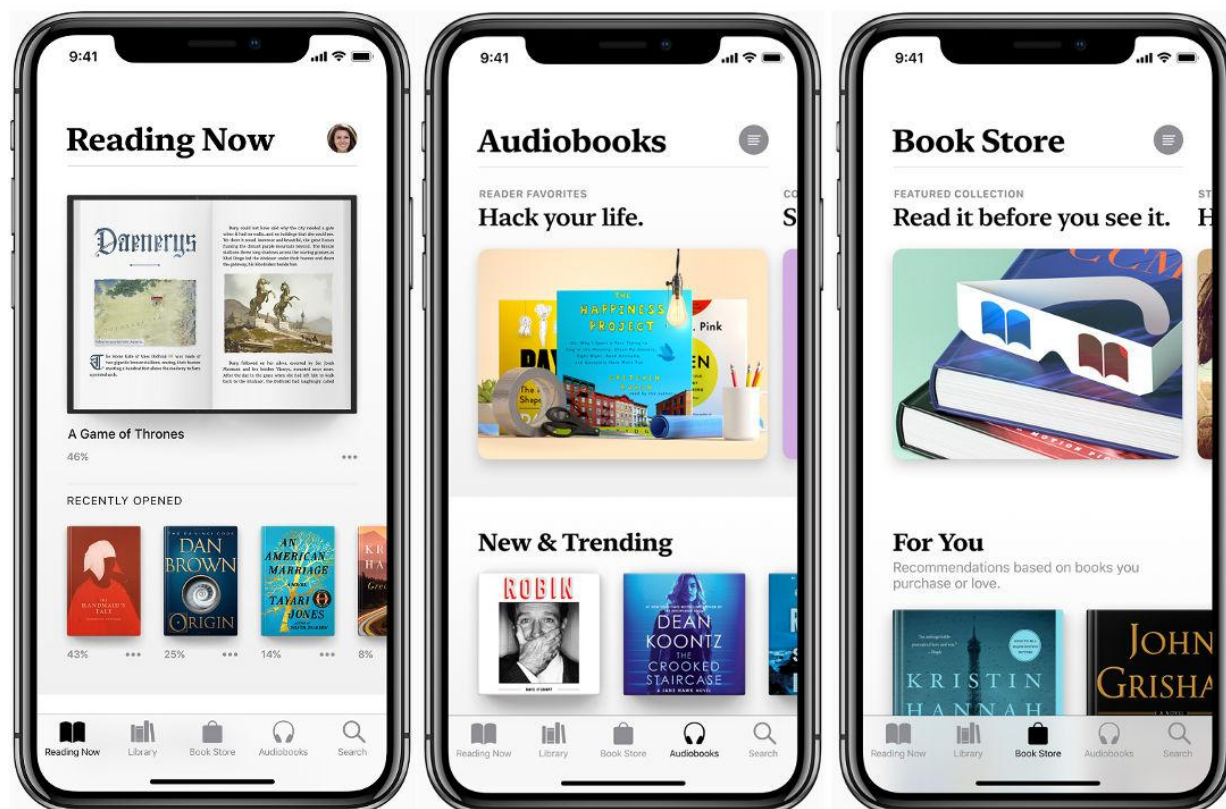


Рис. 2 Сервіс “Apple Books”

Однією з ключових особливостей Apple Books є його велика бібліотека, яка включає мільйони назв із різних жанрів, включаючи художню літературу, науково-дослідну літературу, підручники та аудіокниги. Користувачі можуть переглядати та шукати книги через додаток, використовуючи підібрані списки, персоналізовані рекомендації та категорії, щоб відкривати новий вміст. Apple Books також дозволяє користувачам купувати та завантажувати заголовки безпосередньо з додатка, полегшуючи доступ до матеріалів і насолоджуючись читанням на своїх пристроях.

Платформа створена для забезпечення зручного читання на пристроях Apple, включаючи iPhone, iPad і Mac. Користувачі можуть синхронізувати свої бібліотеки та прогрес читання через iCloud, гарантуючи, що вони зможуть продовжити з того місця, де зупинилися, незалежно від пристрою, який вони використовують. Apple Books підтримує різні формати читання, включаючи ePub і PDF, і пропонує такі функції, як регулювання розміру тексту, вибір шрифту та варіанти кольору фону для покращення читабельності.

Окрім традиційних електронних книг, Apple Books пропонує багатий вибір аудіокниг, призначених для користувачів, які віддають перевагу слуханню своїх улюблених книг. Додаток має спеціальний розділ для аудіокниг, що дозволяє користувачам переглядати, купувати та завантажувати аудіоверсії книг. Apple Books також пропонує такі функції, як закладки, нотатки та виділення, що дозволяє користувачам ефективніше взаємодіяти з вмістом [2].

Apple Books містить зручний інтерфейс, який надає перевагу простоті та зручності використання. Додаток містить вбудований магазин для придбання книг і аудіокниг, а також дозволяє користувачам організовувати свої бібліотеки в колекції для кращого керування. Користувачі також можуть прочитати зразки книг перед покупкою, надаючи їм можливість оцінити вміст перед тим, як здійснити покупку.

Однією з відмінних рис Apple Books є її інтеграція з екосистемою Apple. Користувачі можуть скористатися перевагами таких функцій, як Siri для керування без використання рук, а додаток може синхронізуватися з Apple Pay для безпечних і зручних транзакцій. Крім того, Apple Books підтримує такі спеціальні функції, як VoiceOver, що дозволяє користувачам із вадами зору насолоджуватися читанням.

Незважаючи на свої переваги, Apple Books стикається з проблемами на конкурентному ринку. Інші платформи електронних книг, такі як Amazon Kindle і Google Play Books, пропонують свої унікальні переваги, зокрема великі бібліотеки, конкурентоспроможні ціни та сумісність із різними платформами. Однак Apple Books вирізняється бездоганною інтеграцією з екосистемою Apple і зосередженістю на взаємодії з користувачем [2].

Таким чином, Apple Books — це комплексна платформа для цифрового читання, яка надає користувачам доступ до великої бібліотеки електронних та аудіокниг. Його зручний інтерфейс, можливості синхронізації та інтеграція зі службами Apple роблять його привабливим вибором для читачів у екосистемі Apple. Оскільки попит на цифрове читання продовжує

зростати, Apple Books залишається значним гравцем на ринку електронних книг, пропонуючи функції та вміст, які задовольняють різноманітну аудиторію читачів.

### **1.3 Постановка завдання**

Основною метою цього проекту є розробка платформи електронної комерції, призначеної для розповсюдження цифрового контенту, зосереджуючись на покращенні досвіду покупки для користувачів, одночасно надаючи надійні інструменти керування для творців і розповсюджувачів контенту. Ця платформа має на меті полегшити продаж і доставку цифрових продуктів, таких як електронні книги, аудіокниги, ігри та мультимедійний вміст, дозволяючи як споживачам, так і постачальникам безперешкодно працювати на цифровому ринку. Завдяки повній автоматизації процесу транзакцій і покращенню взаємодії з користувачем платформа мінімізує ризики, пов'язані з ручною обробкою замовлень, помилками при введенні даних і неефективним керуванням контентом, що призведе до скорочення витрат і підвищення ефективності роботи.

Для досягнення цієї мети необхідно створити безпечну та надійну базу даних. Ця база даних зберігатиме важливу інформацію, включаючи облікові записи користувачів, каталоги продуктів, історію транзакцій і відомості про ліцензування цифрового вмісту. Впровадження комплексної системи безпеки забезпечить користувачам швидкий і безпечний доступ до їхніх покупок, а також захистить права інтелектуальної власності та дані користувачів.

Платформа надаватиме такі функції, як:

- ефективне керування обліковими записами користувачів, що дозволяє персоналізувати досвід, історію покупок і рекомендації щодо вмісту;
- спрощені процеси транзакцій, які дозволяють користувачам легко переглядати, купувати та завантажувати цифровий вміст;

- відстеження запасів у реальному часі для ефективного керування цифровими активами та запобігання проблемам, пов'язаним із несанкціонованим розповсюдженням;
- розширені можливості звітності та аналітики, які надають розуміння поведінки користувачів, тенденцій продажів і ефективності вмісту.

Платформа електронної комерції матиме інтуїтивно зрозумілий інтерфейс користувача, розроблений для покращення навігації та зручності використання як для споживачів, так і для постачальників контенту. Цей інтерфейс дозволить користувачам легко досліджувати різні категорії цифрового вмісту, керувати своїми бібліотеками та отримувати персоналізовані рекомендації на основі їхніх уподобань та історії веб-перегляду. Крім того, інструменти для створення звітів про продажі, залучення користувачів і показники запасів дозволять творцям і розповсюджувачам контенту приймати обґрунтовані рішення та оптимізувати свої маркетингові стратегії.

Впровадження платформи електронної комерції для розповсюдження цифрового контенту передбачає складну взаємодію між фронтендом, бекендом і сторонніми сервісами. Фронтенд реалізує зручний і адаптивний інтерфейс, що забезпечує легкий доступ користувачів до каталогу контенту, процесу покупки та управління обліковим записом. Бекенд, побудований на сучасному фреймворку, відповідає за обробку бізнес-логіки, автентифікацію, управління транзакціями та ліцензіями, а також зберігання даних у надійній базі. Важливою складовою є інтеграція з надавачами контенту, яка забезпечує автоматизовану доставку цифрових продуктів, оновлення метаданих та контроль доступу. Така архітектура дозволяє компаніям і творцям контенту оптимізувати процеси продажу, швидко адаптуватися до змін ринку та підвищувати якість обслуговування клієнтів, що в підсумку сприяє збільшенню доходів і стійкому розвитку бізнесу.

## 1.4 Функціональні та нефункціональні вимоги

### *Функціональні вимоги:*

1. система повинна дозволяти користувачам створювати облікові записи за допомогою електронної пошти та пароля або через соціальні мережі. Він повинен підтримувати автентифікацію користувача, включаючи відновлення пароля та перевірку електронної пошти;
2. адміністратори повинні мати можливість керувати каталогом продуктів, зокрема додавати, оновлювати та видаляти цифровий вміст. Система повинна підтримувати різні мультимедійні формати, такі як електронні книги, аудіокниги та відео;
3. користувачі повинні мати можливість шукати цифровий вміст за допомогою ключових слів і фільтрів, таких як жанр, автор і ціна. Платформа також має надавати параметри перегляду за категоріями та підібраними списками;
4. користувачі повинні мати можливість додавати товари до кошика для покупок, переглядати кошик і переходити до оформлення замовлення. Система має сприяти безпечній обробці платежів і дозволяти користувачам вибирати методи оплати (кредитна картка, PayPal тощо);
5. платформа повинна впроваджувати рішення DRM для захисту цифрового вмісту від несанкціонованого поширення та піратства, забезпечуючи доступ до придбаних товарів лише авторизованим користувачам;
6. користувачі повинні мати можливість переглядати історію своїх замовлень, включаючи деталі покупок, посилання для завантаження та створення рахунків-фактур. Адміністратори повинні мати можливість керувати замовленнями, включаючи оновлення статусу замовлень і відшкодування;

7. система повинна дозволяти користувачам надсилати відгуки та оцінки для придбаного контенту, сприяючи залученню спільноти та надаючи цінні відгуки творцям контенту;
8. платформа повинна використовувати алгоритми для надання персоналізованих рекомендацій щодо вмісту на основі поведінки користувачів, уподобань та історії покупок;
9. платформа повинна містити довідковий центр або функцію підтримки, яка надає користувачам відповіді на поширені запитання, варіанти зв'язку та систему продажу квитків для вирішення проблем.

***Нефункціональні вимоги:***

1. система повинна бути здатна обробляти велику кількість одночасних користувачів і транзакцій без значного погіршення часу відгуку. Час завантаження сторінки має бути оптимізовано для безпроблемної взаємодії з користувачем;
2. платформа повинна впроваджувати надійні заходи безпеки, включаючи шифрування даних, безпечну обробку платежів і захист від поширених уразливостей, таких як впровадження SQL і міжсайтовий сценарій (XSS);
3. архітектура системи повинна підтримувати масштабованість, дозволяючи збільшити навантаження, пов'язане ростом аудиторії користувачів і додавати нові функції, не вимагаючи значних переробок;
4. інтерфейс користувача має бути інтуїтивно зрозумілим і зручним, що дозволяє користувачам легко орієнтуватися на платформі та виконувати транзакції з мінімальними зусиллями;
5. платформа має бути сумісна з різними пристроями та операційними системами, включаючи настільні ПК, планшети та смартфони. Він повинен підтримувати основні веб-браузери, щоб забезпечити доступність для всіх користувачів;

6. система повинна мати високу доступність, забезпечуючи постійний доступ користувачів до платформи. Час простою для технічного обслуговування слід звести до мінімуму та заздалегідь повідомити користувачів;
7. база коду та архітектура системи повинні бути розроблені таким чином, щоб їх можна було підтримувати, дозволяючи ефективно оновлення, виправлення помилок і вдосконалення функцій з часом [4];
8. платформа повинна підтримувати кілька мов і валют, щоб задовольнити різноманітну базу користувачів і покращити загальний досвід користувача.

Задовольняючи як функціональні, так і нефункціональні вимоги, платформа електронної комерції для розповсюдження цифрового контенту буде добре обладнана для задоволення потреб користувачів і постачальників контенту, одночасно забезпечуючи безпечний, ефективний і приємний досвід.

## **1.5 Вимоги до інтерфейсу користувача**

Інтерфейс користувача (UI) платформи електронної комерції, призначеної для розповсюдження цифрового контенту, життєво важливий для покращення взаємодії з користувачем і сприяння безперебійній взаємодії. Щоб створити ефективну та привабливу платформу, важливо зосередитися на зручності використання, доступності та естетиці.

Інтуїтивно зрозуміла структура навігації має вирішальне значення для керування користувачами платформою. Меню та категорії мають бути логічно організовані, щоб користувачі могли швидко знаходити цифровий вміст, який вони шукають. Впровадження навігації шляхом навігації допоможе користувачам зрозуміти своє поточне місцезнаходження на сайті та легко повернутися до попередніх розділів. Помітна панель пошуку має бути

доступна на кожній сторінці, щоб користувачі могли шукати певний цифровий вміст за допомогою ключових слів або фільтрів, таких як жанр, автор і ціна.

Макет і дизайн платформи повинні віддавати пріоритет узгодженості та оперативності. Візуальні елементи, зокрема колірні схеми, шрифти та піктограми, мають підтримувати єдиний вигляд на всіх сторінках, сприяючи єдиній взаємодії з користувачем. Крім того, користувальницький інтерфейс має бути чутливим і оптимізованим для різних пристроїв, включаючи настільні комп'ютери, планшети та смартфони, забезпечуючи, щоб усі користувачі могли легко орієнтуватися на платформі. Цифрові продукти мають відображатися впорядковано, із чіткими зображеннями, назвами, авторами та короткими описами, а спеціальні сторінки продуктів надають детальну інформацію [5].

Взаємодія з користувачем є ще одним ключовим аспектом інтерфейсу користувача. Кнопки із закликом до дії, як-от «Додати в кошик», «Купити зараз» і «Завантажити», мають бути помітними та легко впізнаваними, з чіткими мітками, щоб орієнтувати користувачів. Необхідно інтегрувати механізми негайного зворотного зв'язку для підтвердження дій користувача, наприклад, коли товар додано в кошик або завершено покупку. Візуальні індикатори, як-от лічильники завантаження або повідомлення про успіх, допоможуть ефективно повідомляти про зміни статусу. Форми введення для реєстрації та оформлення замовлення мають бути зручними для користувача, мати чіткі мітки, корисні заповнювачі та перевірку помилок, щоб полегшити процес.

Доступність має вирішальне значення для охоплення різноманітної аудиторії. Інтерфейс користувача має бути сумісним із програмами зчитування з екрана та допоміжними технологіями, забезпечуючи користувачам із вадами зору доступ до всього вмісту та функцій. Крім того, користувачі повинні мати можливість переміщатися по платформі за допомогою комбінацій клавіш і навігації вкладками, що забезпечує плавну роботу без використання миші. Слід також звернути увагу на колірний

контраст, щоб відповідати стандартам доступності та підтримувати користувачів із вадами зору.

Для ефективної підтримки користувачів до інтерфейсу користувача слід інтегрувати чітко доступний довідковий центр або розділ поширених запитань, що дозволить користувачам знаходити відповіді на типові запитання та проблеми з мінімальними зусиллями. Якщо можливо, включення функції живого чату для підтримки клієнтів у режимі реального часу дозволить користувачам отримувати швидку допомогу під час взаємодії.

Естетичні міркування відіграють значну роль у привабливості платформи. Інтерфейс користувача має бути візуально привабливим, використовувати високоякісні зображення та сучасний дизайн, який відображає поточні тенденції розповсюдження цифрового контенту. Послідовні елементи брендингу, такі як логотипи та слогани, посилять ідентичність бренду та зміцнять довіру користувачів.

Зосереджуючись на цих вимогах до інтерфейсу користувача, платформа електронної комерції для розповсюдження цифрового вмісту може створити зручний, доступний і привабливий досвід. Такий підхід не тільки привабить користувачів, але й підвищить їхню лояльність, що зрештою сприятиме успіху платформи на конкурентному цифровому ринку.

## 2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

### 2.1 Об'єктне та функціональне моделювання

Уніфікована мова моделювання (UML) — це стандартизована мова моделювання, яка широко використовується в розробці програмного забезпечення для візуалізації, специфікації, побудови та документування різних компонентів програмної системи. Він пропонує набір методів графічної нотації, які дозволяють зацікавленим сторонам, включаючи розробників, дизайнерів, аналітиків і клієнтів, створювати чіткі візуальні моделі об'єктно-орієнтованих програмних систем.

Діаграма — це візуальне зображення, призначене для передачі інформації, понять або зв'язків між різними елементами. Використовуючи символи, форми, лінії та кольори, діаграми представляють ідеї в чіткій і стислій формі, роблячи складну інформацію більш доступною та легшою для аналізу. Вони служать різним цілям, наприклад, ілюструють процеси, структури, потоки даних або зв'язки між різними об'єктами.

**2.2.1 Діаграма прецедентів.** Діаграми варіантів використання є важливим аспектом уніфікованої мови моделювання (UML), яка використовується в розробці програмного забезпечення для візуального зображення функціональних вимог системи. Ці діаграми ілюструють, як користувачі, яких називають акторами, взаємодіють із програмним забезпеченням, висвітлюючи різні доступні їм функції. Надаючи чітке представлення цих взаємодій, діаграми варіантів використання допомагають зацікавленим сторонам зрозуміти поведінку системи з точки зору кінцевого користувача [7].

Діаграма варіантів використання містить кілька ключових елементів. Актори представляють користувачів або зовнішні системи, які взаємодіють із системою, що моделюється. Ними можуть бути окремі особи, наприклад клієнти чи адміністратори, або інші системи, наприклад платіжні

шлюзи. На діаграмі актори зображені у вигляді фігурок і розташовані за межами системи.

Варіанти використання описують конкретні функції або характеристики системи з точки зору користувача, представляючи цілі, яких учасники прагнуть досягти шляхом їх взаємодії. Вони зображені овалами з назвою варіанта використання всередині. Межа системи, представлена прямокутником, інкапсулює всі випадки використання, відрізняючи те, що є частиною системи, від зовнішніх елементів, таким чином забезпечуючи контекст для зображених функцій.

Відносини між акторами та варіантами використання є ще одним важливим аспектом діаграм варіантів використання. Основним зв'язком є асоціація, позначена суцільною лінією, що з'єднує актора з варіантом використання, що означає участь актора в цьому варіанті використання. Крім того, взаємозв'язок включення зображено пунктирною стрілкою з позначкою `include`, що свідчить про те, що варіант використання є обов'язковим компонентом іншого варіанту використання, що дозволяє повторно використовувати спільну функціональність. Відношення розширення, представлене пунктирною стрілкою з позначкою `extend`, вказує

на те, що варіант використання може бути вдосконалений іншим варіантом використання, як правило, для додавання необов'язкової поведінки.

Розроблена діаграма прецедентів використання представлена на рис. 3.

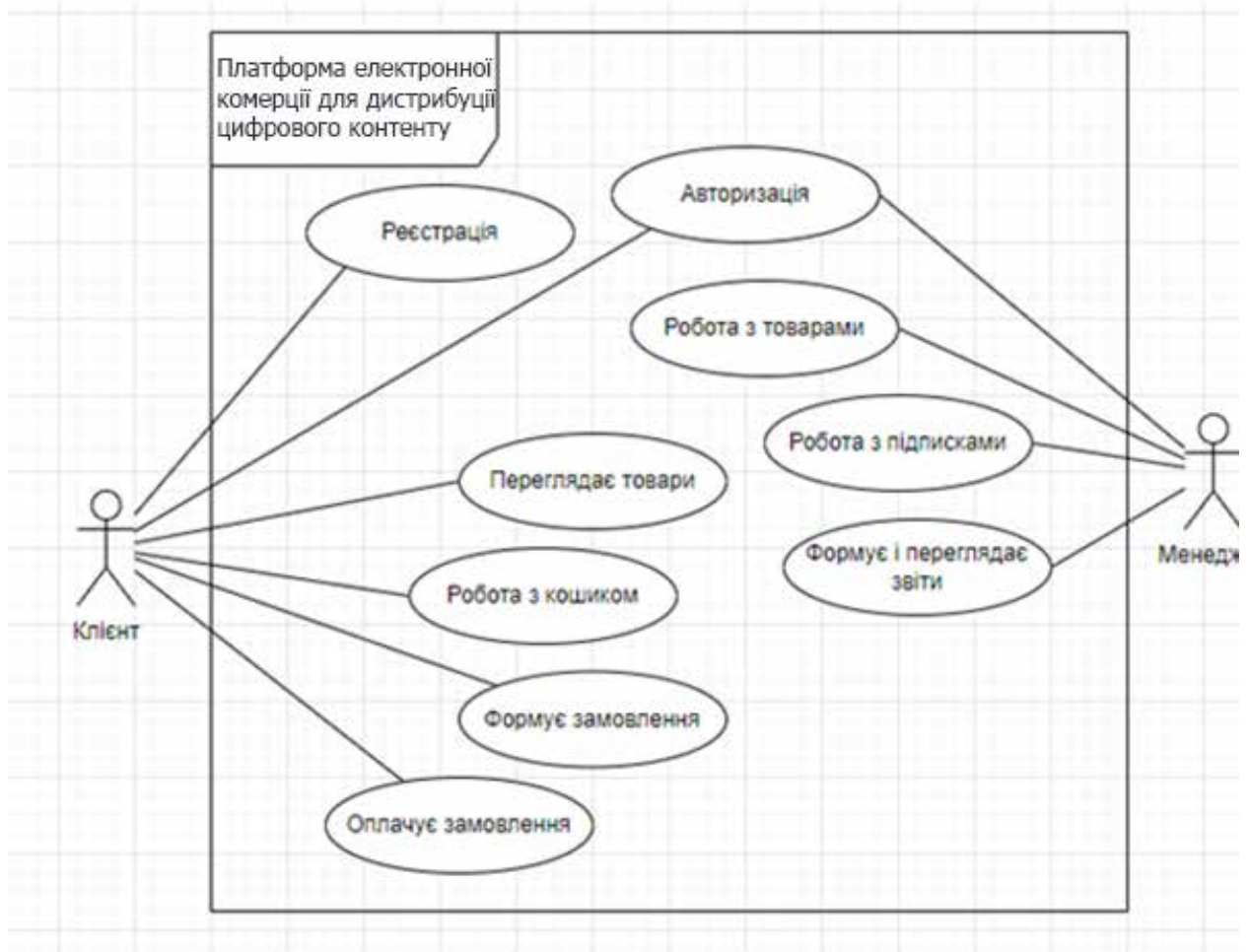


Рис. 3 Діаграма прецедентів

Створена діаграма прецедентів містить акторів:

- “Клієнт”;
- “Менеджер”.

Актор «Клієнт» включає такі прецеденти:

- реєстрація;
- авторизація;

- перегляд товарів;
- робота з кошиком;
- формування замовлення;
- оплата заказу.

Актор «Менеджер» включає такі прецеденти:

- авторизація;
- робота з товарами;
- робота з підписками;
- формування та перегляд звітів.

Прецеденти певним чином залежать одне від одного.

Розглянемо детальніше вищеописані прецеденти.

**Назва випадку використання:** Придбання товарів

**Актор:** Клієнт

**Передумови:** Клієнт має активний обліковий запис в інтернет-магазині ігор. Клієнт увійшов у свій обліковий запис. У кошику клієнта є товари.

**Основний потік:**

1. клієнт переміщується в онлайн-магазині, переглядаючи доступні товари та вміст. Вони використовують фільтри та функції пошуку, щоб знайти певні назви чи жанри;
2. знайшовши потрібну річ, клієнт натискає кнопку «Додати в кошик». Система оновлює кошик для покупок, щоб відобразити доданий товар. Клієнт може продовжувати перегляд і додавати більше товарів у кошик;
3. коли клієнт готовий зробити покупку, він натискає значок кошика для покупок, щоб переглянути вибрані товари. Система відображає короткий перелік товарів у кошику, включаючи назву, ціну та кількість;

4. клієнт натискає кнопку «Оформити замовлення», щоб розпочати процес покупки. Система запропонує клієнту підтвердити вибрані позиції та надасть огляд загальної вартості, включаючи будь-які застосовні податки та збори;
5. клієнт перенаправляється на захищену платіжну сторінку, де він вводить свої платіжні дані, наприклад інформацію про кредитну картку, або вибирає альтернативний спосіб оплати (наприклад, PayPal). Система перевіряє платіжну інформацію на точність;
6. клієнт переглядає підсумок замовлення, переконуючись у правильності всієї інформації, включно з елементами, загальною вартістю та платіжними деталями. Вони можуть внести необхідні зміни;
7. клієнт натискає кнопку «Підтвердити покупку», щоб завершити операцію. Система обробляє платіж і оновлює інвентар;
8. після успішної обробки платежу система генерує підтвердження замовлення. Клієнт отримує сповіщення електронною поштою з деталями покупки, включаючи квитанцію та інструкції щодо завантаження придбаних ігор;
9. тепер клієнт може отримати доступ до придбаних товарів через свою бібліотеку в онлайн-магазині, готових для завантаження або гри.

#### **Альтернативні потоки:**

1. якщо платіж не вдається через недійсну інформацію, система запропонує клієнту повторно ввести платіжні дані та повторити транзакцію;
2. якщо клієнт вирішує скасувати покупку під час оформлення замовлення, він може повернутися до кошика або переглянути магазин, не завершуючи транзакцію.

Цей сценарій описує типовий потік купівлі товарів у магазині онлайн-ігор, деталізуючи взаємодію між клієнтом і системою, підкреслюючи основні функції, задіяні в процесі купівлі.

Розглянемо ще один сценарій використання.

**Назва випадку використання:** Створення підписки

**Актор:** менеджер

**Передумови:** Менеджер увійшов в адміністративний інтерфейс інтернет-магазину ігор. Менеджер має необхідні дозволи для створення та керування підписками.

**Основний потік:**

1. менеджер переходить до розділу керування підписками адміністративного інтерфейсу. Ця область містить параметри для створення, перегляду та зміни існуючих підписок;
2. менеджер вибирає варіант створення нової підписки. У системі представлена форма для введення реквізитів підписки;
3. менеджер заповнює необхідні поля у формі створення підписки, яка може містити: назву підписки, опис, ціну, тривалість, включені ігри;
4. після введення всієї необхідної інформації менеджер перевіряє правильність даних підписки. Система відображає зведення параметрів підписки;
5. менеджер натискає кнопку «Створити підписку», щоб надіслати інформацію. Система обробляє запит і створює підписку в базі даних;
6. після успішного створення підписки система виводить менеджеру повідомлення про підтвердження. Сповіщення електронною поштою також може бути надіслано менеджеру з підтвердженням нових деталей підписки;

7. щойно створена підписка додається до списку доступних підписок, що робить її доступною для клієнтів через інтернет-магазин ігор.

#### **Альтернативні потоки:**

1. якщо менеджер намагається створити підписку з відсутньою або недійсною інформацією (наприклад, від'ємна ціна), система видасть повідомлення про помилку та випросить виправлення, перш ніж продовжити;
2. якщо менеджер скасовує процес створення, він може повернутися до розділу керування підписками без збереження змін.

Цей сценарій описує типовий процес створення підписки в онлайн-магазині ігор, деталізуючи взаємодію між менеджером і системою, підкреслюючи основні функції, задіяні в процесі створення підписки.

**2.1.2 Діаграма послідовності.** Діаграми послідовності — це тип діаграми взаємодії в уніфікованій мові моделювання (UML), яка ілюструє, як об'єкти взаємодіють у конкретному сценарії використання. Ці діаграми зосереджені на порядку повідомлень, якими обмінюються об'єкти або компоненти, забезпечуючи детальне уявлення про динамічну поведінку системи. Візуалізуючи послідовність взаємодій у часі, діаграми послідовностей допомагають зацікавленим сторонам зрозуміти потік керування та даних у системі [8].

Діаграма послідовності зазвичай складається з кількох ключових компонентів. Вертикальна вісь представляє різні об'єкти або учасників, які беруть участь у взаємодії, часто зображуються у вигляді позначених прямокутників у верхній частині діаграми. Кожен об'єкт пов'язаний із лінією життя, представленою пунктирною вертикальною лінією, яка

тягнеться вниз, що вказує на існування об'єкта протягом певного часу під час взаємодії.

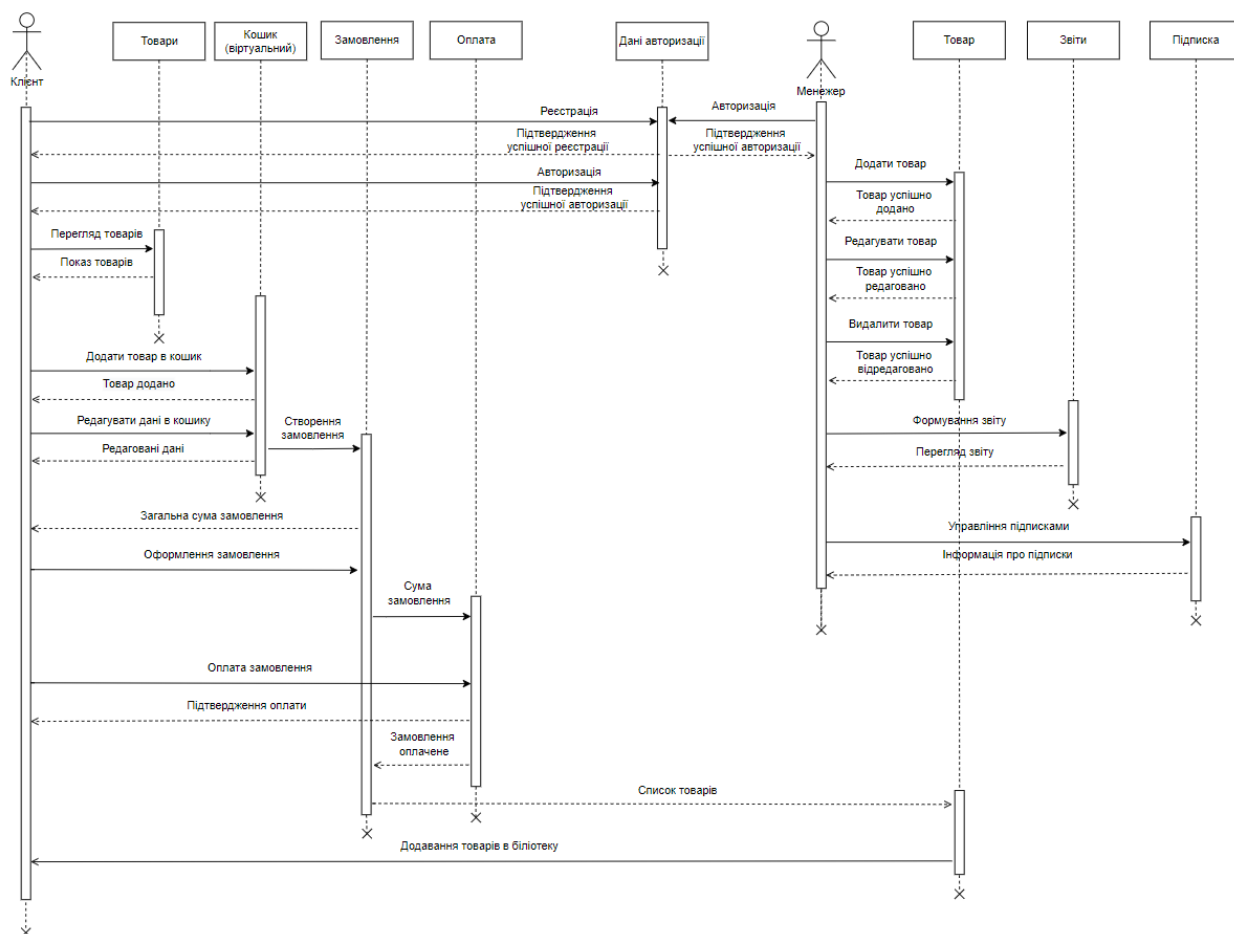


Рис. 4 Діаграма послідовності

Діаграма послідовності, зображена на рис. 4, включає наступні об'єкти:

- клієнт;
- товари;
- кошик(віртуальний);
- замовлення;
- оплата;
- дані авторизації;
- менеджер;
- товар;
- звіти;
- підписка.

Процес починається з того, що клієнт взаємодіє з об'єктом «Авторизація», щоб увійти до свого облікового запису. Після входу в систему клієнт може переглядати доступні продукти на веб-сайті. Вони надсилають запит на перегляд товарів, а у відповідь отримують список доступних товарів. Після цієї взаємодії об'єкт «Продукти» виконує свою функцію.

Далі клієнт додає вибраний товар у свій кошик. У цей момент реквізити клієнта та продукту передаються в об'єкт «Кошик», що дозволяє подальше маніпулювання цими даними. Потім кошик обчислює загальну суму замовлення та відображає її клієнту. Клієнт вибирає спосіб оплати, надсилає свою платіжну інформацію та отримує підтвердження сформованого замовлення.

**2.1.3 Діаграма активності.** Діаграми діяльності — це тип діаграм поведінки в уніфікованій мові моделювання (UML), які представляють потік дій і дій у системі чи процесі. Вони особливо корисні для моделювання складних робочих процесів і бізнес-процесів, надаючи візуальне представлення того, як виконуються різні завдання та як вони пов'язані одне з одним. Ілюструючи послідовність дій і моменти прийняття рішень, діаграми дій допомагають зацікавленим сторонам зрозуміти динамічні аспекти системи [8].

Діаграма діяльності зазвичай складається з кількох ключових елементів. Діаграма починається з початкового вузла, представленого у вигляді зафарбованого кола, що означає початкову точку робочого процесу. Від цього початкового вузла стрілки вказують на потік керування та підключаються до різних дій, представлених у вигляді округлених прямокутників. Кожна діяльність представляє конкретне завдання або дію, яка відбувається в процесі.

Точки прийняття рішення позначаються вузлами у формі ромба, які зображують шляхи розгалуження на основі конкретних умов або вибору. Ці точки прийняття рішень дозволяють представити альтернативні потоки в

рамках діяльності, ілюструючи, як різні шляхи можуть бути прийняті в залежності від результату конкретної умови.

Діаграма також містить смуги синхронізації, які є горизонтальними або вертикальними лініями, що представляють паралельні дії. Коли дії можуть відбуватися одночасно, смужки синхронізації вказують, де сходяться або розходяться кілька потоків. Ця функція необхідна для моделювання складних процесів, у яких завдання можуть виконуватися одночасно.

Розроблена діаграма активності представлена на рис. 5.

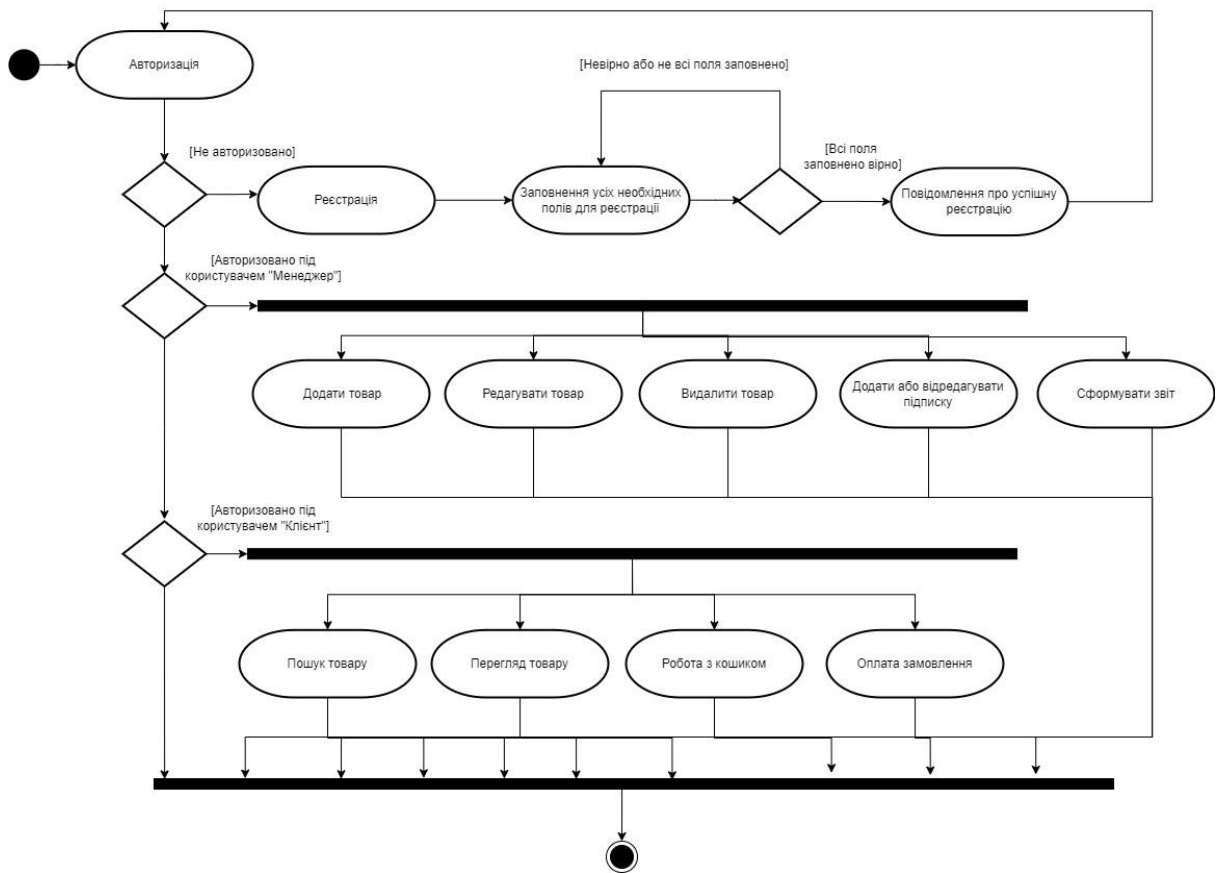


Рис. 5 Діаграма активності

## 2.2 Абстракції предметної області

Абстракції предметної області передбачають створення спрощених уявлень, які підкреслюють основні характеристики, концепції та зв'язки в межах певної області чи системи, виключаючи сторонні деталі. Цей підхід

має вирішальне значення в різних сферах, включаючи розробку програмного забезпечення, бізнес-аналіз та проектування системи, оскільки він дозволяє зацікавленим сторонам зосередитися на критичних елементах системи чи проблемному просторі, не перевантажуючись складністю.

Основною метою абстракції є покращення розуміння та комунікації між зацікавленими сторонами шляхом представлення високорівневого погляду на систему. Зосереджуючись на найважливіших компонентах, абстракції пояснюють цілі, функції та взаємодію в межах предметної області. Це спрощення зрештою призводить до покращення аналізу, проектування та прийняття рішень.

У моделюванні зазвичай використовуються кілька типів абстракцій. Концептуальні моделі представляють фундаментальні ідеї та відносини в межах домену. Наприклад, у контексті платформи електронної комерції концептуальна модель може ідентифікувати ключові сутності, такі як користувачі, продукти, замовлення та платежі, а також їхні взаємозв'язки. Ця перспектива високого рівня допомагає зацікавленим сторонам зрозуміти основні компоненти системи [9].

Переваги абстракції значні. Спрощуючи складність, абстракції полегшують зацікавленим сторонам досягнення основних понять і зв'язків у предметній області. Вони також покращують спілкування між членами команди, забезпечуючи спільне розуміння цілей і функцій системи.

Більше того, абстракція забезпечує цілеспрямований аналіз і проектування, дозволяючи командам зосередитися на основних елементах, що призводить до більш ефективних процесів розробки. Це допомагає визначити вимоги та потенційні проблеми на початку циклу. Крім того, абстракції забезпечують гнучку структуру, яку можна адаптувати та вдосконалювати, коли стає доступною більше інформації або змінюються вимоги, що має вирішальне значення в динамічних середовищах [9].

Підсумовуючи, абстракції предметної області являють собою спрощені форми, які фіксують істотні елементи та зв'язки системи чи

домену. Підкреслюючи критичні аспекти та пропускаючи непотрібні деталі, ці абстракції покращують розуміння, полегшують спілкування та підтримують ефективний аналіз і дизайн. Вони відіграють важливу роль у складності навігації та прийнятті обґрунтованих рішень протягом усього процесу розробки.

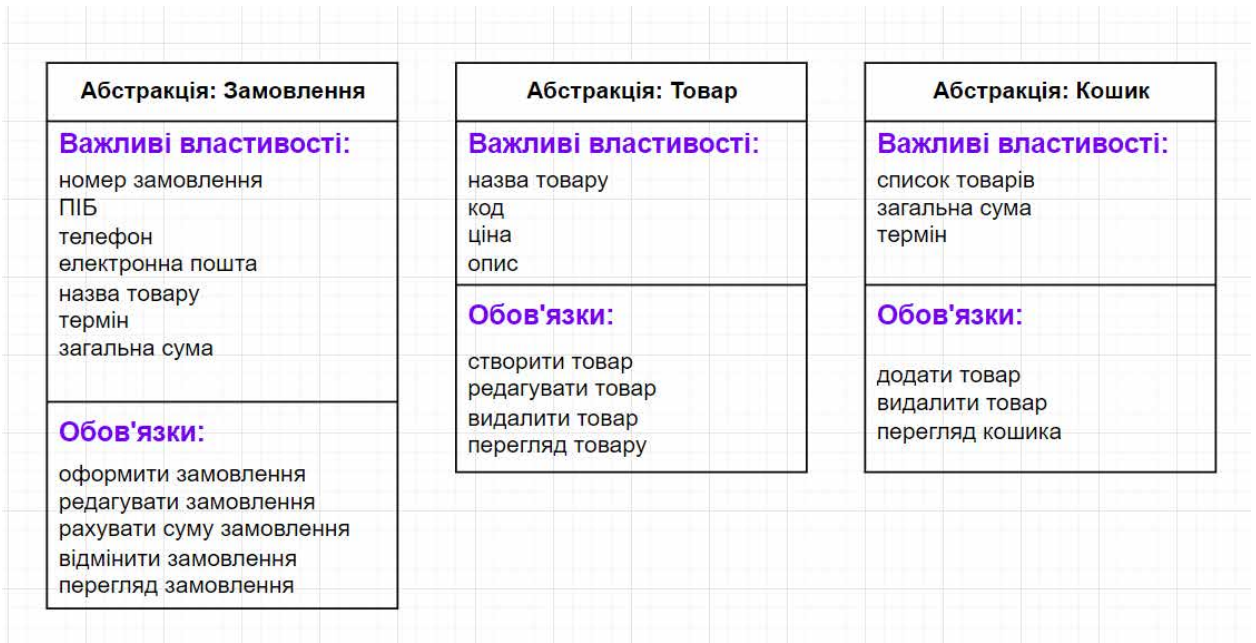


Рис. 6 Абстракції предметної області

## 2.3 Діаграма класів

Діаграми класів є фундаментальним компонентом уніфікованої мови моделювання (UML) і відіграють вирішальну роль в об'єктно-орієнтованому проектуванні та моделюванні. Вони забезпечують статичне уявлення про систему, ілюструючи класи, їхні атрибути, методи та зв'язки між ними. Діаграми класів служать схемою для розробки програмного забезпечення, дозволяючи розробникам зрозуміти структуру та організацію системи перед впровадженням.

Діаграма класів складається з кількох ключових елементів. Класи представлені у вигляді прямокутників, розділених на три відсіки. Верхній відсік містить назву класу, середній відсік перераховує атрибути (властивості або члени даних), а нижній відсік детально описує методи

(функції або операції), пов'язані з класом. Цей структурований формат дозволяє чітко представити, що представляє кожен клас і як він поводить себе в системі [10].

Переваги діаграм класів значні. Вони забезпечують чітке та організоване представлення структури системи, що полегшує зацікавленим сторонам розуміння того, як різні класи взаємодіють і відносяться один до одного. Візуально зображуючи класи та їхні взаємозв'язки, ці діаграми сприяють кращому спілкуванню між членами команди, забезпечуючи спільне розуміння архітектури системи.

Діаграми класів також відіграють важливу роль на етапі проєктування розробки програмного забезпечення. Вони допомагають визначити необхідні класи, їхні атрибути та методи, а також зв'язки між ними, дозволяючи розробникам створювати добре структуровану кодову базу. Крім того, діаграми класів можуть бути корисними для цілей документації, надаючи посилання для майбутнього обслуговування та оновлень системи.

Таким чином, діаграми класів є важливим інструментом об'єктно-орієнтованого проєктування та моделювання, пропонуючи статичне уявлення про структуру системи. Ілюструючи класи, їхні атрибути, методи та зв'язки, діаграми класів служать схемою для розробки програмного забезпечення, покращуючи розуміння та спілкування між зацікавленими сторонами, одночасно сприяючи ефективній розробці та реалізації.

Для нашого веб-сайту була створена власна діаграма класів за об'єктно-орієнтованим підходом, яка представлена на рисунку 7.

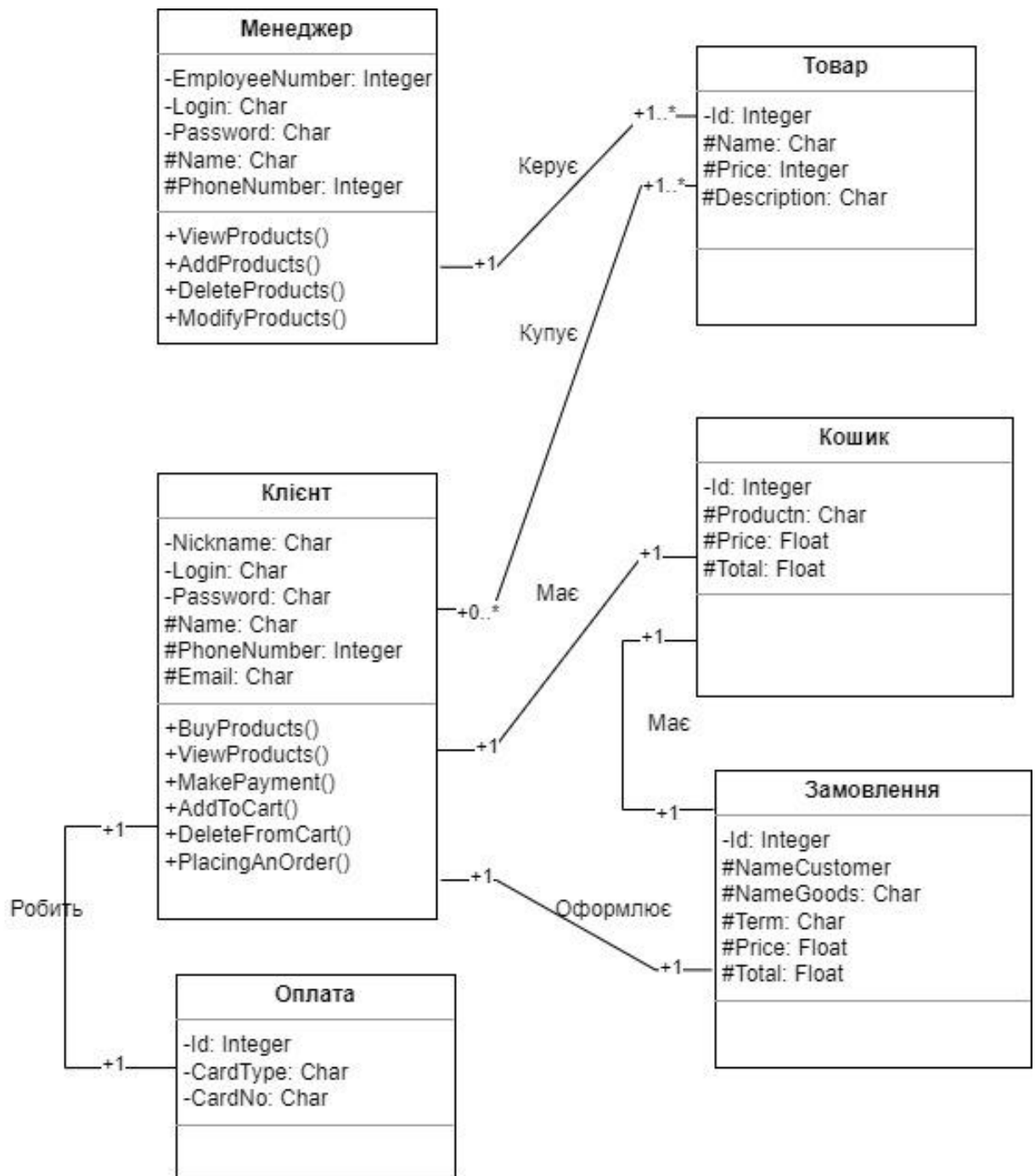


Рис. 7 Діаграма класів

## 3 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 3.1 Логічна модель даних

Логічна модель даних — це детальне представлення, яке описує структуру й організацію даних у системі, зосереджуючись на логічних взаємозв'язках між різними елементами даних, а не на тому, як дані фізично зберігатимуться чи реалізовуватимуться. Ця модель служить важливим посередником між концептуальними моделями високого рівня та фізичними моделями даних, які використовуються в розробці баз даних. Його основна мета полягає в тому, щоб чітко визначити вимоги до даних і зв'язки в спосіб, який не залежить від конкретних систем керування базами даних (СУБД) або технологій [11].

Центральними для логічної моделі даних є кілька ключових компонентів. Сутності представляють окремі об'єкти або поняття в системі, наприклад «Клієнт», «Продукт», «Замовлення» та «Платіж» на платформі електронної комерції. Кожна сутність відповідає об'єкту реального світу або концепції, що має відношення до системи. Атрибути описують характеристики або властивості сутності; наприклад, сутність «Клієнт» може містити такі атрибути, як «Ідентифікатор клієнта», «Ім'я», «Електронна адреса» та «Номер телефону». Ці атрибути надають конкретні відомості про кожну сутність.

Зв'язки в моделі визначають, як сутності пов'язані одна з одною. Ці зв'язки можуть приймати різні форми, зокрема «один до одного», «один до багатьох» або «багато до багатьох». Наприклад, «Клієнт» може розміщувати кілька «Замовлень», встановлюючи зв'язок «один до багатьох» між двома об'єктами. Первинні ключі також є важливим аспектом моделі; вони служать унікальними ідентифікаторами для кожного екземпляра сутності, забезпечуючи чітке розпізнавання кожного запису. У випадку сутності «Клієнт» «Ідентифікатор клієнта» може функціонувати як його первинний

ключ. Крім того, зовнішні ключі створюють зв'язки між об'єктами, встановлюючи зв'язки; наприклад, сутність "Замовлення" може містити зовнішній ключ "CustomerID" для посилання на відповідного клієнта.

Логічна модель даних служить багатьом важливим цілям. Він пропонує структуроване та чітке представлення вимог до даних і зв'язків у системі, що полегшує зацікавленим сторонам розуміння того, як взаємодіють різні елементи даних. Ця чіткість покращує спілкування між членами команди, включаючи бізнес-аналітиків, розробників і адміністраторів баз даних, забезпечуючи загальну структуру для обговорення вимог до даних.

Крім того, логічна модель даних діє як план для розробки фізичної моделі даних, яка визначає, як дані зберігатимуться в конкретній системі бази даних. Він гарантує, що всі необхідні елементи даних і взаємозв'язки ідентифікуються перед впровадженням, оптимізуючи процес проектування та мінімізуючи ймовірність помилок або упущень [11].

Крім того, добре структурована логічна модель даних сприяє покращенню якості даних, забезпечуючи цілісність і узгодженість даних через чітко визначені зв'язки та обмеження. Її незалежність від будь-якої конкретної технології забезпечує гнучкість, дозволяючи за потреби адаптувати модель до різних систем баз даних. Ця адаптивність полегшує перемикання технологій із зміною вимог.

Підсумовуючи, логічна модель даних відіграє життєво важливу роль в управлінні даними та проектуванні системи, надаючи повне представлення структур даних і зв'язків у системі. Підкреслюючи сутності, атрибути, зв'язки та ключі, ця модель сприяє чіткій комунікації, підтримує

ефективне проєктування та покращує якість даних, що зрештою сприяє ефективному управлінню даними та успішному впровадженню системи.

Логічна модель системи представлена на рисунку 8.

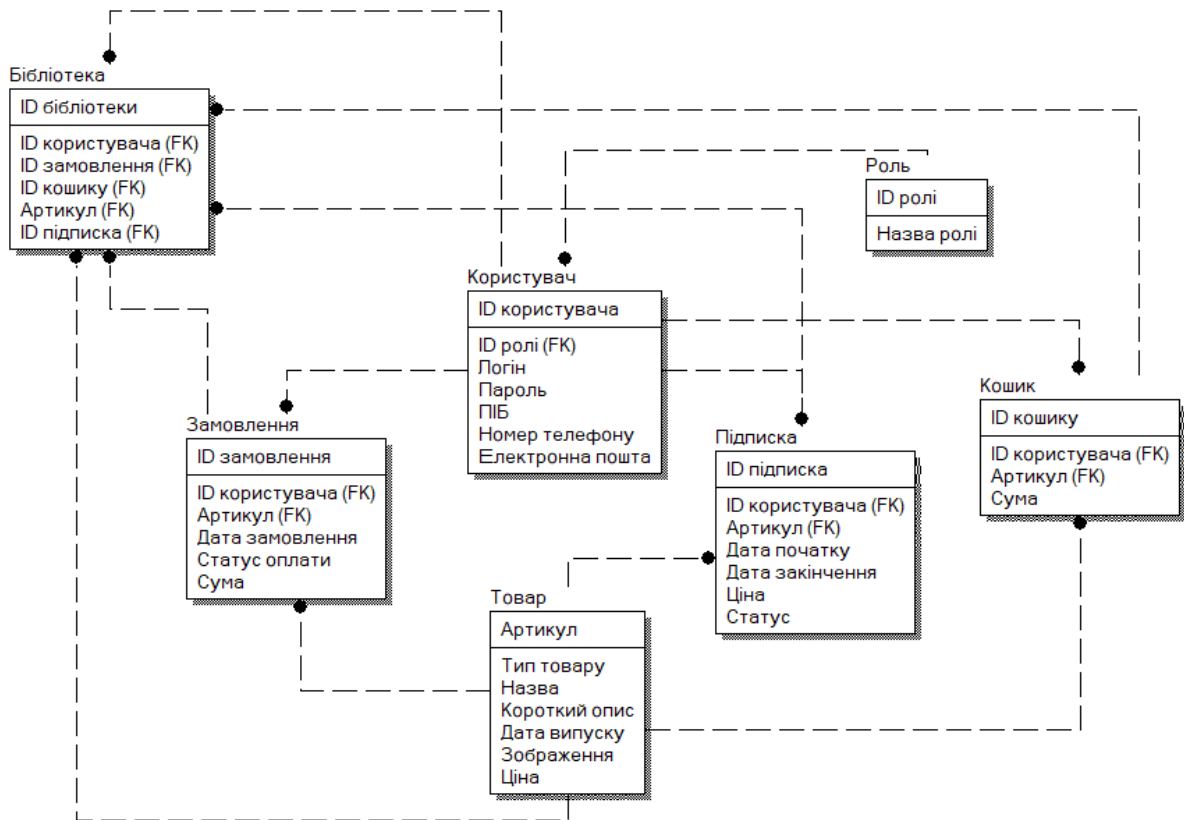


Рис. 8 ER-діаграма

Логічна модель складається з таких сутностей:

1. “Користувач”;
2. “Бібліотека”;
3. “Замовлення”;
4. “Товар”;
5. “Підписка”;
6. “Роль”;
7. “Кошик”.

Сутність “Користувач” має такі атрибути:

- id\_користувача;
- id\_ролі;
- логін;

- пароль;
- піб;
- номер телефону;
- електронна пошта.

Сутність “Бібліотека” має такі атрибути:

- id\_бібліотеки;
- id\_кошику;
- id\_користувача;
- id\_замовлення;
- артикул;
- id\_підписки.

Сутність “Замовлення” має такі атрибути:

- id\_замовлення;
- id\_користувача;
- артикул;
- дата\_замовлення;
- статус\_оплати;
- сума.

Сутність “Товар” має такі атрибути:

- артикул;
- тип\_товару;
- назва;
- короткий\_опис;
- ціна;
- зображення;
- дата\_випуску.

Сутність “Підписка” має такі атрибути:

- id\_користувача;
- артикул;

- дата\_початку;
- дата\_закінчення;
- статус;
- ціна.

Сутність “Роль” має такі атрибути:

- id\_ролі;
- назва\_ролі.

Сутність “Кошик” має такі атрибути:

- id\_кошику;
- id\_користувача;
- артикул;
- сума.

Сутність «Користувач» пов’язана з сутностями «Бібліотека», «Підписка», «Кошик» та «Замовлення». Також сутність «Товар» зв’язана з сутностями «Підписка», «Замовлення» та «Кошик». І сутність “Роль” зв’язана з сутністю “Користувач”.

### **3.2 Вибір системи управління базою даних та її реалізація**

Вибір правильної системи керування базами даних (СУБД) має вирішальне значення для розробки платформи електронної комерції для розповсюдження цифрового контенту. СУБД ляже в основу платформи, керуючи зберіганням, пошуком і організацією даних, що стосуються користувачів, продуктів, замовлень і транзакцій. У цьому розділі обговорюються критерії вибору відповідної СУБД, оцінюються потенційні варіанти та описується процес впровадження.

При виборі СУБД необхідно враховувати кілька важливих факторів. Він має відповідати вимогам до структури даних програми, особливо тому, що платформа електронної комерції зазвичай використовує реляційну базу

даних. Така база даних здатна обробляти складні запити, зберігаючи цілісність даних через встановлені зв'язки.

Масштабованість є ще одним важливим фактором. Обрана СУБД повинна підтримувати зростаючий обсяг даних і навантаження на користувачів у міру розширення платформи. Він має запропонувати можливості горизонтального масштабування, щоб забезпечити стабільність продуктивності в періоди підвищеного попиту [13].

Продуктивність також має вирішальне значення. СУБД має забезпечувати швидкий пошук даних та ефективну обробку транзакцій, оскільки користувачі очікують швидкого часу відповіді під час перегляду та купівлі продуктів. Функції безпеки також важливі, оскільки платформа оброблятиме конфіденційні дані користувачів і інформацію про транзакції. Обрана система має забезпечувати надійні заходи безпеки, включаючи шифрування та контроль доступу.

Крім того, СУБД має бездоганно інтегруватися в технологічний стек платформи електронної комерції, включаючи вибрану мову програмування та фреймворки. Також корисно вибрати СУБД із сильною підтримкою спільноти та документацією, яка з часом може допомогти у вирішенні проблем і розробці.

Кілька популярних систем керування реляційними базами даних виділяються як потенційні кандидати для цього проєкту. MySQL, рішення з відкритим кодом, відоме своєю надійністю та продуктивністю, підтримує широкий спектр програм і пропонує надійні функції безпеки. Він має переваги від великої спільноти та великої документації, що робить його сильним суперником.

PostgreSQL — ще один потужний варіант із відкритим кодом, який пропонує розширені функції, такі як підтримка складних запитів і обробка географічних даних. Відомий своєю зосередженістю на цілісності та розширюваності даних, PostgreSQL також пропонує потужні параметри

безпеки, що робить його придатним для керування конфіденційною інформацією.

Microsoft SQL Server пропонує комплексне рішення реляційної бази даних із розширеними функціями аналітики та керування даними. Хоча це комерційний продукт із потенційно вищими витратами на ліцензування, він забезпечує чудову продуктивність і масштабованість.

SQLite — це легка реляційна база даних на основі файлів, яку легко налаштувати. Хоча він підходить для невеликих додатків або середовищ розробки, він може бути не ідеальним для повномасштабної платформи електронної комерції через обмеження масштабованості та паралельності.

Після оцінки цих варіантів MySQL стає найбільш підходящим вибором для платформи електронної комерції. Його поєднання продуктивності, простоти інтеграції та підтримки спільноти робить його ідеальним рішенням для цього проєкту.

Реалізація бази даних MySQL включає кілька етапів. Першим кроком є інсталяція, яку можна виконати в локальному середовищі розробки або на хмарному сервері. Цей процес зазвичай включає завантаження інсталятора MySQL, налаштування необхідних параметрів і захист інсталяції за допомогою пароля root.

Після встановлення починається етап проєктування бази даних. На основі попередньо розробленої логічної моделі даних створюється схема бази даних, що визначає таблиці, стовпці, типи даних, первинні ключі, зовнішні ключі та зв'язки. Ключові таблиці для платформи електронної комерції включатимуть користувачів, продукти, замовлення та платежі, кожна з яких призначена для ефективного зберігання відповідної інформації.

Якщо існують дані, які потрібно імпортувати в нову базу даних, сценарії або інструменти міграції даних можуть полегшити передачу інформації. Впровадження заходів безпеки має вирішальне значення для

захисту бази даних, включаючи дозволи користувача, SSL для безпечних з'єднань і конфігурації резервного копіювання для запобігання втраті даних.

Після налаштування бази даних важливо перевірити її продуктивність. Це включає оптимізацію запитів для забезпечення ефективного пошуку даних і обробки транзакцій. Створення індексів для часто запитуваних стовпців може значно підвищити продуктивність.

База даних MySQL повинна бути інтегрована з прикладним рівнем платформи електронної комерції. Це передбачає використання відповідних конекторів бази даних або бібліотек для забезпечення зв'язку між програмою та базою даних [13].

Підсумовуючи, вибір MySQL як системи керування базами даних для платформи електронної комерції є добре зваженим рішенням, заснованим на її продуктивності, масштабованості, функціях безпеки та простоті інтеграції. Процес впровадження включає встановлення, проєктування бази даних, міграцію даних, налаштування безпеки, тестування та інтеграцію.

Рішення обрати MySQL як систему керування базою даних для платформи електронної комерції, призначеної для розповсюдження цифрового контенту, було зумовлене кількома ключовими факторами, які відповідають вимогам і цілям проєкту.

Однією з головних причин вибору MySQL є його висока продуктивність і надійність. MySQL відома своєю здатністю обробляти великі обсяги транзакцій і ефективно отримувати дані, що є вкрай важливим для платформи електронної комерції, де користувачі очікують швидкого часу відповіді під час перегляду продуктів і обробки транзакцій. Оптимізація бази даних для операцій читання та запису гарантує, що платформа може підтримувати оптимальну продуктивність, навіть якщо трафік користувачів зростає.

Ще однією істотною перевагою MySQL є її масштабованість. У міру розвитку платформи електронної комерції кількість даних, отриманих із

взаємодії користувачів, списків продуктів і транзакцій, значно зросте. MySQL забезпечує надійні можливості масштабування, що дозволяє системі розширюватися горизонтально шляхом додавання додаткових серверів за потреби. Ця масштабованість гарантує, що платформа може пристосуватися до майбутнього зростання без шкоди для продуктивності.

Гнучкість MySQL в обробці даних також є вирішальним фактором. Платформа вимагатиме структурованого підходу до керування різними типами даних, включаючи інформацію про користувача, деталі продукту, записи про замовлення та платіжну інформацію. Структура реляційної бази даних MySQL дозволяє встановлювати чіткі зв'язки між цими об'єктами, сприяючи ефективному управлінню та пошуку даних за допомогою складних запитів.

Безпека має першочергове значення в будь-якій програмі електронної комерції, особливо під час обробки конфіденційних даних користувачів і фінансових операцій. MySQL пропонує комплексні функції безпеки, включаючи автентифікацію користувача, рольовий контроль доступу та шифрування даних. Ці функції допомагають захистити цілісність і конфіденційність даних, що зберігаються в базі даних, забезпечуючи спокій як операторам платформи, так і її користувачам.

Крім того, MySQL — це система керування базами даних із відкритим вихідним кодом, що є економічно вигідною для проєкту. Відкритий вихідний код означає відсутність ліцензійних зборів, пов'язаних з його використанням, що дозволяє проєкту більш ефективно розподіляти ресурси. Крім того, MySQL має велику спільноту користувачів і обширну документацію, що забезпечує цінну підтримку та ресурси для розробників під час впровадження та поточного обслуговування [14].

Підсумовуючи, рішення вибрати MySQL для впровадження платформи електронної комерції для розповсюдження цифрового контенту ґрунтується на його високій продуктивності, масштабованості, гнучкості в управлінні даними, надійних функціях безпеки, економічній ефективності

та сумісності з різними середовищами програмування. Ці фактори в сукупності гарантують, що MySQL добре підходить для підтримки вимог платформи та сприяє бездоганній взаємодії з користувачем.

Отриману схему бази даних зображено на малюнку 9.

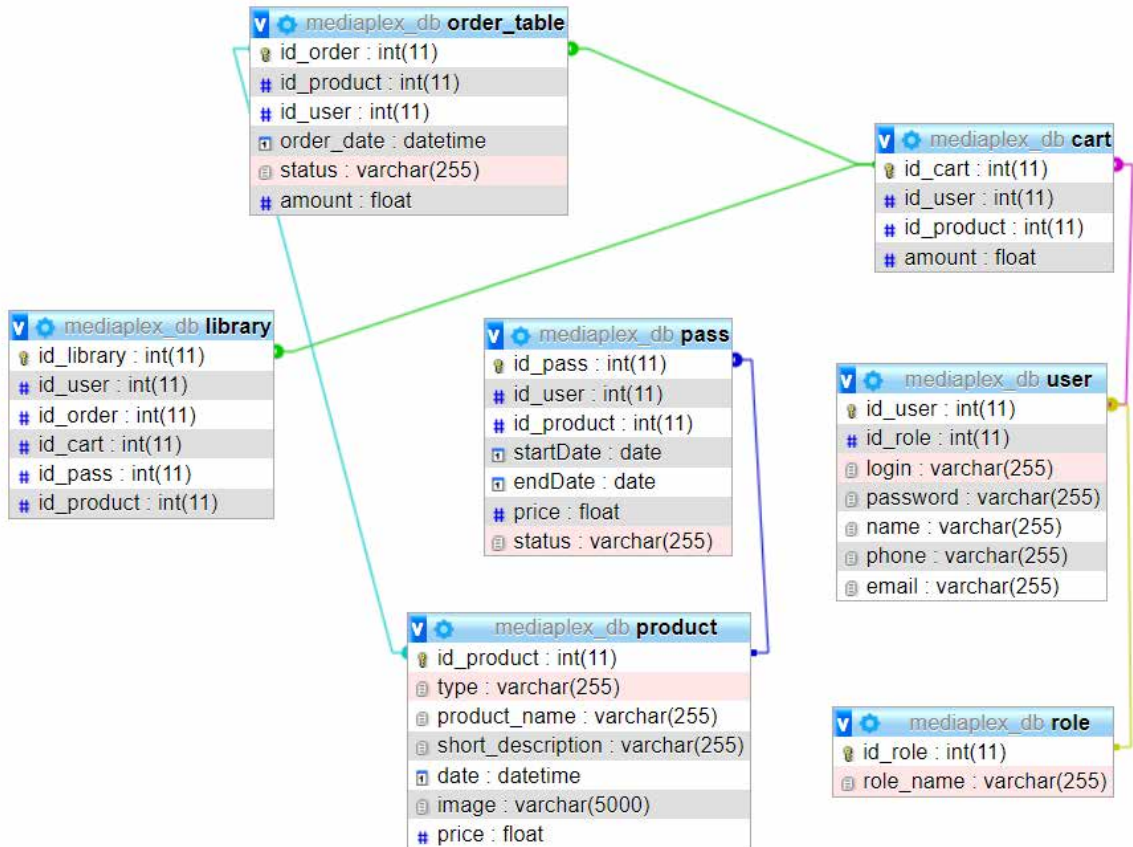


Рис. 9 База даних системи

### 3.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення відноситься до високорівневої структури та організації програмної системи, що охоплює її фундаментальні компоненти та їх взаємодію. Ця архітектурна структура служить схемою для проектування та розробки програмних додатків, гарантуючи, що вони відповідають як технічним, так і бізнес-вимогам. Він

містить вказівки щодо того, як різні частини системи працюють разом і як вони взаємодіють із зовнішніми елементами.

Чітко визначена архітектура включає кілька критичних елементів. Її основою є компоненти, які є модульними частинами програмного забезпечення, які інкапсулюють певні функції. Ці компоненти можуть включати модулі, класи, служби та бібліотеки, кожен з яких має чітко визначений інтерфейс і набір обов'язків, що сприяє розподілу завдань. Взаємозв'язки між цими компонентами однаково важливі, оскільки вони визначають, як дані та керування протікають через систему та як різні частини взаємодіють між собою.

Архітектура програмного забезпечення часто використовує усталені архітектурні моделі та стилі, такі як багаторівнева архітектура, мікросервіси та архітектури клієнт-сервер. Ці шаблони забезпечують перевірені рішення типових проблем проектування та керують організацією системи. Крім того, архітектурні рішення мають значний вплив на різні якості програмного забезпечення, включаючи продуктивність, масштабованість, надійність, безпеку та зручність обслуговування. Архітектура повинна бути розроблена з урахуванням цих атрибутів, щоб оптимізувати ефективність системи [15].

Архітектура програмної системи має вирішальне значення для її успіху з кількох причин. Він закладає основу для всього процесу розробки, надаючи керівні принципи, які забезпечують послідовність і узгодженість у реалізації. Ця загальна структура полегшує спілкування між зацікавленими сторонами, такими як розробники, архітектори, керівники проєктів і клієнти, забезпечуючи кращу співпрацю та чіткіші обговорення проєктних рішень і системних вимог.

Добре структурована архітектура покращує масштабованість і гнучкість, дозволяючи системі розвиватися та адаптуватися до зростаючих вимог. Це полегшує інтеграцію нових функцій або зміну існуючих, не порушуючи роботу системи в цілому. Крім того, продуманий вибір

архітектури допомагає зменшити ризики на ранніх стадіях процесу розробки, виявляючи потенційні проблеми до того, як вони стануть серйозними.

Технічне обслуговування — це ще одна сфера, де архітектура програмного забезпечення відіграє життєво важливу роль. Оскільки програмні системи вимагають регулярних оновлень і налаштувань, чітка архітектурна структура сприяє модульності та розподілу завдань. Ця модульність спрощує завдання з обслуговування, дозволяючи розробникам змінювати або замінювати окремі компоненти, не впливаючи на всю систему.

У розробці програмного забезпечення зазвичай використовуються різні архітектурні стилі. Рівнева архітектура організовує систему на окремі рівні, кожен з яких має певні обов'язки, тоді як архітектура мікросервісів розбиває програму на невеликі незалежні служби, які спілкуються через API, підвищуючи гнучкість і масштабованість. Клієнт-серверна архітектура розділяє зовнішні та серверні компоненти, дозволяючи їм функціонувати незалежно, тоді як керована подіями архітектура дозволяє компонентам спілкуватися через події та повідомлення, сприяючи слабкому зв'язку та асинхронній обробці.

Підсумовуючи, архітектура програмного забезпечення є фундаментальним аспектом розробки програмного забезпечення, який описує структуру високого рівня та організацію системи. Визначаючи компоненти, їхні взаємозв'язки та керівні принципи проектування, архітектура програмного забезпечення діє як план для створення надійних, масштабованих і підтримуваних програм. Його важливість виходить за рамки простого проектування, полегшення спілкування між зацікавленими сторонами, керування процесами розробки та забезпечення відповідності системи як технічним специфікаціям, так і бізнес-цілям.

Нижче буде продемонстровано багат шарова архітектура даного програмного забезпечення (рис. 10)

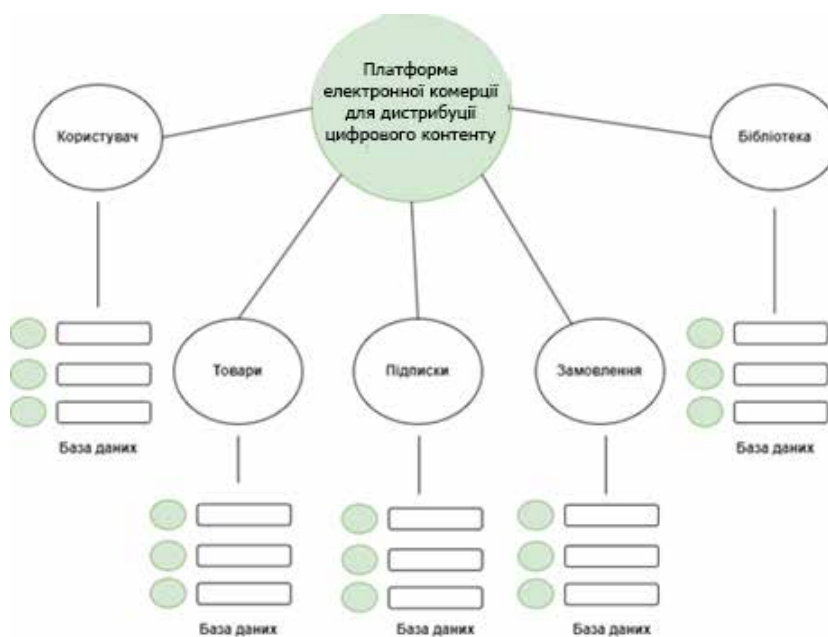


Рис. 10 Багатошарова архітектура веб-сайту

Архітектура нашої системи розроблена як багаторівнева структура, яка складається з трьох окремих рівнів: рівня презентації, рівня бізнес-логіки та рівня бази даних. Інформація послідовно перетікає з одного рівня на інший, створюючи цілісну та організовану структуру.

Перший рівень - це інтерфейс користувача, який відображається на екрані комп'ютера або мобільного пристрою. Тут користувачі взаємодіють із системою, користуючись різними функціями та функціями.

Другий рівень охоплює основні функції програми, з якими користувачі активно взаємодіють. На цьому рівні користувачі можуть виконувати операції, пов'язані з продуктами, підписками та замовленнями, що полегшує їхній загальний досвід використання платформи.

Третій рівень складається з бази даних, яка зберігає всю необхідну системі інформацію. Така архітектурна конструкція ефективно організовує компоненти системи, забезпечуючи плавний зв'язок між ними. Це

забезпечує ефективну обробку даних і покращує взаємодію з користувачем, що зрештою сприяє бездоганному досвіду для користувачів.

### **3.4 Організаційна структура програмного забезпечення**

**3.4.1 Діаграма пакетів.** Діаграми пакетів — це тип структурної діаграми в уніфікованій мові моделювання (UML), яка допомагає організовувати та групувати пов'язані елементи в програмній системі. Вони особливо корисні для управління складністю, розбиваючи систему на керовані компоненти або пакети. Кожен пакет може представляти різні елементи, такі як класи, інтерфейси або підсистеми, що полегшує візуалізацію загальної організації системи [16].

В основі діаграм пакетів – самі пакети, зображені у вигляді прямокутників із невеликою вкладкою у верхньому лівому куті. Ці пакунки містять елементи UML, що дозволяє розробникам групувати пов'язані компоненти для кращої ясності та організації. Окрім пакетів, у цих діаграмах вирішальними є залежності. Вони ілюструють, як один пакет залежить від іншого, представлені пунктирними стрілками, які вказують від залежного пакета до пакета, з яким він пов'язаний. Розуміння цих залежностей допомагає визначити потенційний вплив, коли зміни відбуваються в одному пакеті.

Діаграми пакетів служать багатьом цілям у проектуванні та розробці програмного забезпечення. Вони допомагають упорядковувати компоненти системи шляхом групування пов'язаних елементів, що зменшує складність і покращує навігацію в архітектурі системи. Така організована структура не тільки полегшує розуміння, але й сприяє ефективній комунікації між членами команди та зацікавленими сторонами.

Діаграма пакета (рис. 11) для цього програмного забезпечення використовується для візуального представлення архітектури та організації системи. У ньому зображено різні пакети, що становлять програмне забезпечення, і показано зв'язки та залежності між ними.

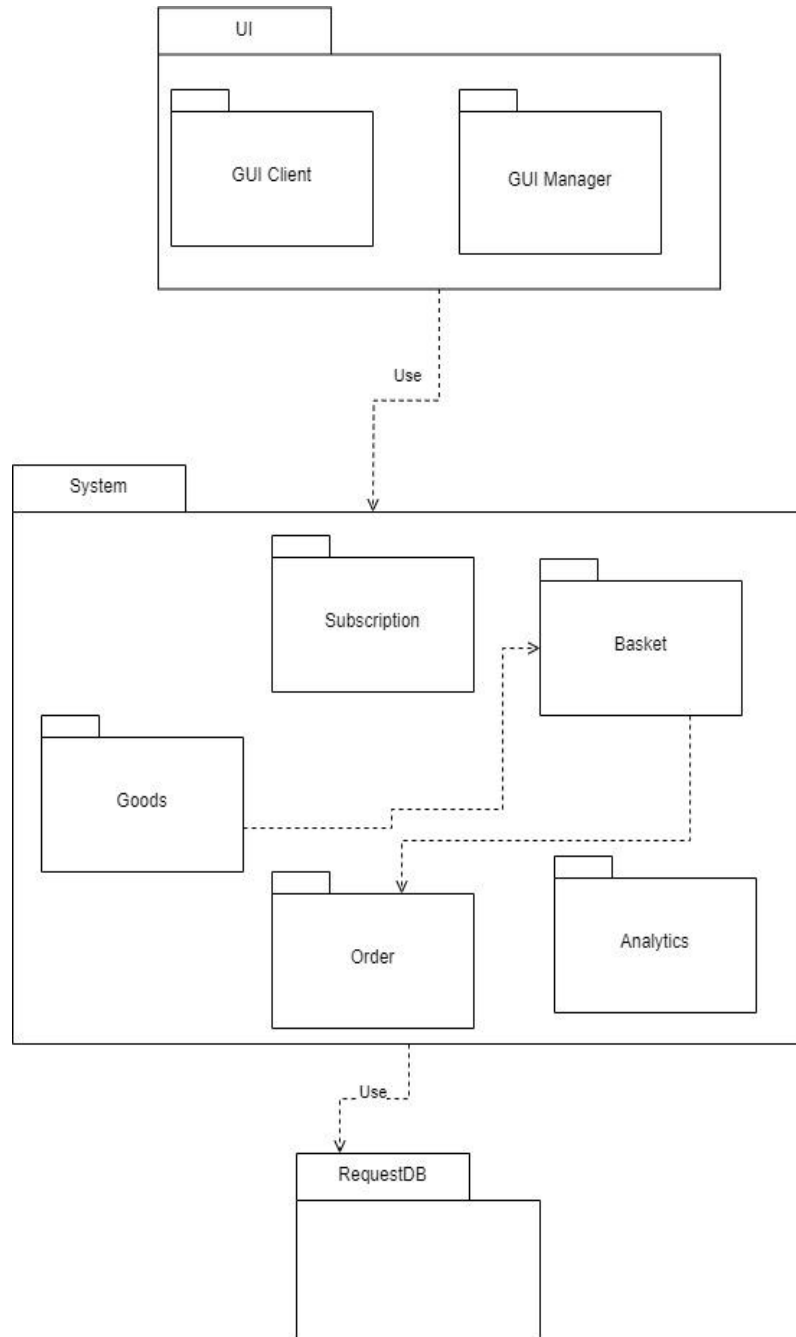


Рис. 11 Діаграма пакетів

### 3.5 Вибір інструментарію для створення програмного забезпечення

У розробці платформи електронної комерції для розповсюдження цифрового вмісту вибір правильних інструментів і технологій має вирішальне значення для забезпечення надійної, ефективної та зручної програми. Для цього проекту було вибрано поєднання мов програмування, фреймворків і програмних засобів, кожен з яких служить певній меті в процесі розробки. Вибрані інструменти включають PHP, Laravel, MySQL, HTML, CSS, JavaScript і Photoshop CS6.

PHP є основною серверною мовою сценаріїв, обраною для цього проекту. Відомий своєю гнучкістю та простотою використання, PHP добре підходить для створення динамічних веб-додатків. Він надає широкий спектр вбудованих функцій, які спрощують типові завдання, такі як обробка форм, керування сесіями та взаємодія з базами даних. Популярність PHP і широка підтримка спільноти роблять його ідеальним вибором для розробки платформи електронної комерції.

Для покращення процесу розробки та оптимізації архітектури програми було обрано фреймворк Laravel. Laravel — це потужний фреймворк PHP, який відповідає архітектурному шаблону MVC (Model-View-Controller). Він пропонує численні функції, включаючи виразний синтаксис, вбудовану автентифікацію та надійну систему маршрутизації. Акцент Laravel на читабельності коду та зручності обслуговування дозволяє розробникам ефективно створювати масштабовані програми. Крім того, його багата екосистема, включаючи такі інструменти, як Eloquent ORM для взаємодії з базами даних і шаблони Blade для візуалізації інтерфейсу, значно прискорює розробку.

MySQL було обрано як систему керування базами даних для платформи. MySQL, відома своєю продуктивністю та надійністю, є реляційною базою даних із відкритим кодом, яка широко використовується у

веб-додатках. Його здатність обробляти складні запити та підтримувати цілісність даних через зв'язки робить його придатним вибором для керування різними об'єктами даних у платформі електронної комерції, такими як користувачі, продукти, замовлення та платежі. Сумісність MySQL з PHP і Laravel додатково покращує його інтеграцію в загальну архітектуру програми.

Для розробки інтерфейсу платформи електронної комерції HTML, CSS і JavaScript були обрані як основні технології. HTML (HyperText Markup Language) забезпечує структуру веб-сторінок, дозволяючи розробникам створювати необхідні елементи, такі як заголовки, абзаци та посилання. CSS (каскадні таблиці стилів) використовується для стилізації веб-сторінок, покращуючи візуальну привабливість і забезпечуючи адаптивний дизайн, який адаптується до різних пристроїв і розмірів екрана. JavaScript додає програмі інтерактивність, забезпечуючи динамічне оновлення вмісту, перевірку форм і покращення взаємодії з користувачем. Разом ці технології гарантують, що інтерфейс є функціональним і візуально привабливим.

Photoshop CS6 було обрано як інструмент графічного дизайну для створення та редагування візуальних ресурсів для платформи. Він надає потужні функції для розробки інтерфейсів користувача, оптимізації зображень і створення маркетингових матеріалів. Використовуючи Photoshop CS6, дизайнери можуть створювати високоякісну графіку та макети, які покращують загальну естетику платформи електронної комерції, забезпечуючи її привабливість і зручність для користувача.

Щодо UI/UX, інтерфейс розроблений з орієнтацією на зручність і простоту використання, що відповідає функціональним потребам платформи. Для користувачів передбачено інтуїтивну навігацію по каталогу цифрового контенту з можливістю швидкого пошуку та фільтрації товарів за категоріями. Процес оформлення замовлення спрощений до кількох кроків, включаючи чітке відображення деталей покупки, вибір способу оплати та підтвердження транзакції. Особливу увагу приділено адаптивності інтерфейсу — він коректно відображається на різних пристроях, від мобільних телефонів до

десктопів. Використання JavaScript дозволяє реалізувати динамічне оновлення контенту без перезавантаження сторінки, а також перевірку введених даних у формах у режимі реального часу.

Крім того, в UI інтегровані механізми безпеки, такі як захист від несанкціонованого доступу і багаторівнева автентифікація, що підвищує довіру користувачів. Важливою складовою UX є зворотній зв'язок — система надає повідомлення про успішне завершення дій, а також інформує про помилки чи затримки у процесі покупки. Все це сприяє підвищенню залученості клієнтів і загальному позитивному враженню від користування платформою.

## 4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ

### 4.1 Вимоги до апаратного та програмного забезпечення

Встановлення відповідних вимог до обладнання та програмного забезпечення є життєво важливим для успішного розвитку платформи електронної комерції, призначеної для розповсюдження цифрового контенту. Ці специфікації гарантують, що система працює ефективно та результативно, задовольняючи потреби користувачів і очікування продуктивності.

Що стосується апаратного забезпечення, інфраструктура має складатися з виділеного веб-сервера або екземпляра хмарного сервера, наприклад, наданого AWS, Azure або Google Cloud. Рекомендується, щоб сервер був оснащений багатоядерним процесором, таким як Intel Xeon або AMD EPYC, що містить мінімум чотири ядра. Для підтримки роботи платформи під різними навантаженнями необхідно принаймні 8 ГБ оперативної пам'яті, а для підвищення продуктивності бажано 16 ГБ або більше. Зберігання також має вирішальне значення; Рекомендується твердотільний накопичувач (SSD) з мінімальною ємністю 100 ГБ для розміщення операційної системи, файлів програми та бази даних. Залежно від обсягу цифрового вмісту, що розповсюджується, може знадобитися додаткове сховище. Крім того, рекомендовано надійне підключення до Інтернету з мінімальною пропускною здатністю 100 Мбіт/с, щоб забезпечити оптимальну продуктивність для користувачів, які мають доступ до платформи.

Платформа електронної комерції також повинна обслуговувати різні клієнтські пристрої. Користувачі, ймовірно, отримають доступ до платформи через настільні та ноутбуки за допомогою сучасних веб-браузерів, таких як Chrome, Firefox або Safari, а також мобільних пристроїв, включаючи

смартфони та планшети. Забезпечення чутливості платформи забезпечить безперебійну роботу на всіх типах пристроїв.

Що стосується програмного забезпечення, сервер повинен працювати на стабільній і безпечній операційній системі, такій як Ubuntu Server, яка сумісна з PHP і MySQL. Альтернативні варіанти включають CentOS або Debian. Вибір програмного забезпечення веб-сервера також важливий; Рекомендовано такі параметри, як Apache HTTP Server або Nginx, причому Nginx часто віддають перевагу через його продуктивність і ефективність використання ресурсів.

MySQL слугуватиме системою керування реляційною базою даних для платформи, ефективно керуючи даними користувачів, інформацією про продукти, замовленнями та транзакціями. Що стосується інструментів розробки, остання стабільна версія PHP буде використовуватися як мова сценаріїв на стороні сервера разом із фреймворком Laravel для оптимізації розробки програм. Composer керуватиме залежностями PHP, а HTML5, CSS3 і JavaScript формуватимуть основу для розробки інтерфейсу.

Для потреб графічного дизайну Adobe Photoshop CS6 буде використовуватися для створення та редагування візуальних ресурсів, таких як дизайн інтерфейсу користувача та маркетингові матеріали. Крім того, Git забезпечить контроль версій, забезпечуючи ефективну співпрацю між розробниками та ефективне керування змінами коду.

Щоб переконатися, що платформа функціонує належним чином, можуть бути використані додаткові засоби тестування, розгортання та моніторингу. PHPUnit полегшить модульне тестування для додатків PHP, тоді як Postman використовуватиметься для тестування кінцевих точок API. Якщо це можливо, Docker можна використовувати для контейнеризації, а інструменти моніторингу, такі як New Relic або Grafana, можуть допомогти відстежувати продуктивність.

Викладаючи рекомендовані специфікації для серверної інфраструктури та інструментів розробки, платформа буде добре оснащена,

щоб забезпечити безперебійну та ефективну роботу користувача для тих, хто шукає цифровий контент.

На рисунку 12 зображено діаграму розгортання даної системи. Клієнт-серверна архітектура реалізована на двох окремих серверах.

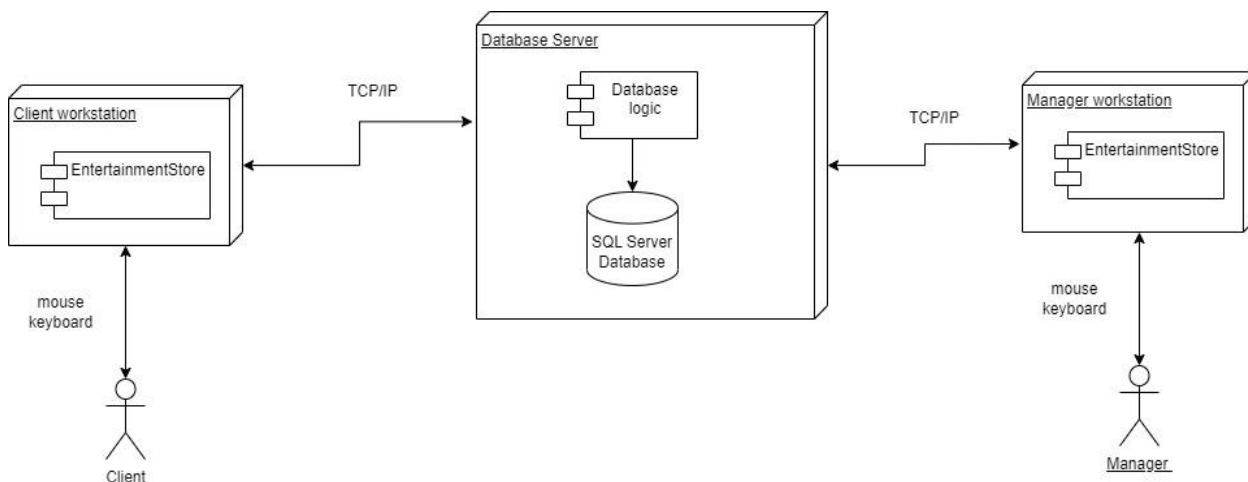


Рис. 12 Діаграма розгортання

## 4.2 Тестування системи

Запустивши систему, бачимо головну сторінку програмного забезпечення (рис. 13-14).

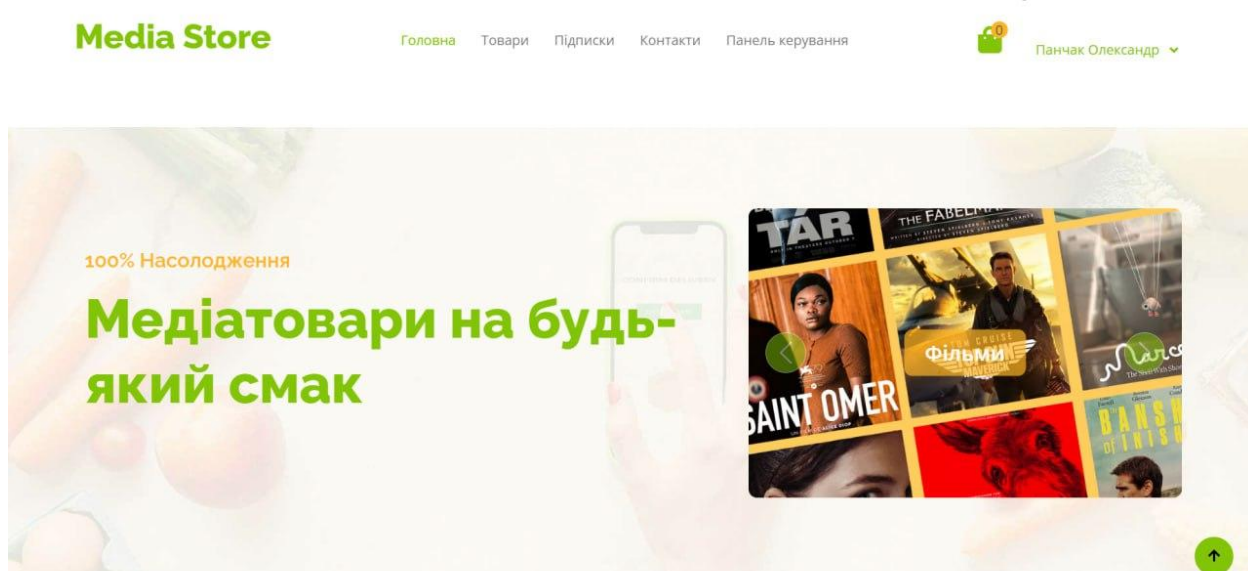


Рис. 13 Головна сторінка

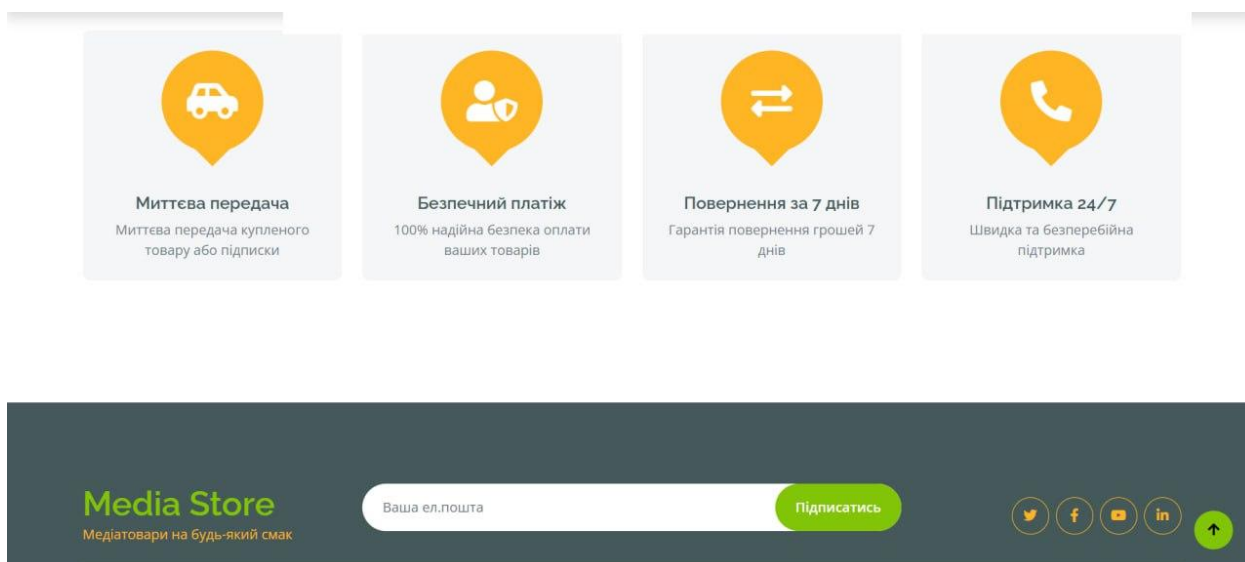


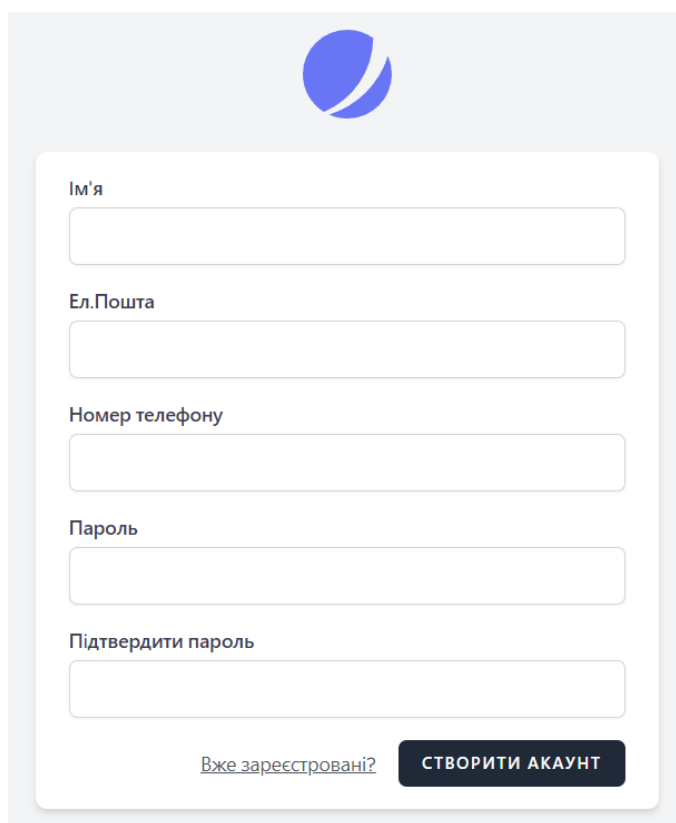
Рис. 14 Головна сторінка

На головній сторінці користувач бачить приємну сторінку, що його зустрічає. Зверху є навігаційна панель з переходами у всі сторінки, що цікавляться, а також можливість авторизації або реєстрації. Якщо користувача вже авторизовано, виводиться саме його ім'я.

На рис. 15 представлена форма авторизації в системі, яку необхідно заповнити при першому вході в систему.

Рис. 15 Форма авторизації

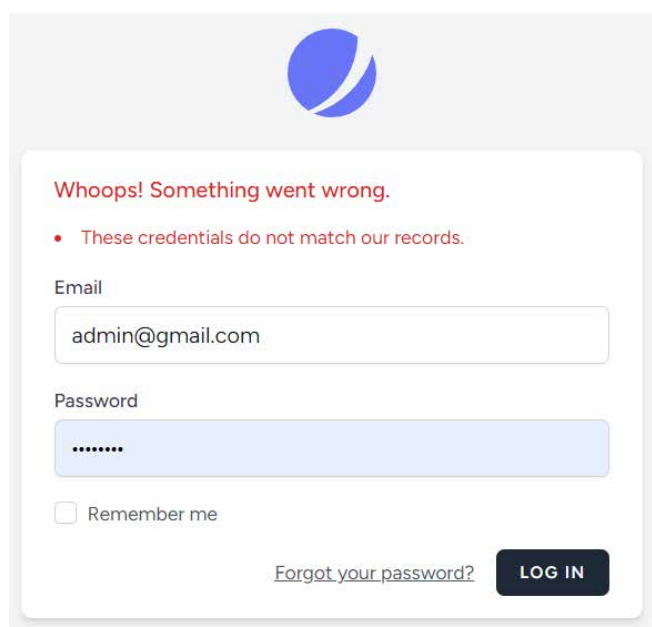
Наступним кроком є реєстрації в системі – створення профілю користувача та заповнення його особистими даними. Ця форма представлена на рис. 16.



The registration form features a blue circular logo at the top center. Below it, the form is organized into several sections, each with a label and a corresponding input field: 'Ім'я' (Name), 'Ел.Пошта' (Email), 'Номер телефону' (Phone number), 'Пароль' (Password), and 'Підтвердити пароль' (Confirm password). At the bottom of the form, there is a link for users who are already registered ('Вже зареєстровані?') and a dark blue button labeled 'СТВОРИТИ АКАУНТ' (Create Account).

Рис. 16 Форма реєстрації

При вході систему ввівши неправильно дані буде виводитись помилка входу та показуватиметься повідомленню користувачеві (рис. 17).



The login error message is displayed on a light gray background with the same blue circular logo at the top. The message is in red text: 'Whoops! Something went wrong.' Below this, a red bullet point states: 'These credentials do not match our records.' The form fields for 'Email' (containing 'admin@gmail.com') and 'Password' (masked with dots) are visible. There is also a 'Remember me' checkbox and a dark blue 'LOG IN' button. A link for 'Forgot your password?' is located at the bottom left of the form area.

Рис. 17 Помилка при спробі авторизації

Увійшовши в систему можна обрати вкладку Товари та переглянути товари (рис. 18). На цій сторінці можна сортувати товари за категоріями.

## Цифрові товари на будь-який вибір

### Категорії

- Всі
- Ігри
- Фільми
- Книжки



#### The Witcher 3

Одна з найвідоміших рольових ігор усіх часів. Тепер готова до нового витка своєї історії. ПРО ГРУ Ви — Геральт із Рівії, найманий мисливець.

500 грн

[Додати в кошик](#)



#### Battlefield 4

Тільки в Battlefield ви зможете підірвати основу греблі або повністю зруйнувати величезний кмарочос або навіть очолити висадку морського десанту.

650 грн

[Додати в кошик](#)

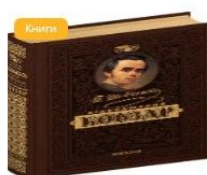


#### Red Dead Redemption 2

Розроблена творцями Grand Theft Auto V і Red Dead Redemption, Red Dead Redemption 2 — це епічна історія життя в невідомій Америці

1500 грн

[Додати в кошик](#)



#### Кобзар

Відроджений "КОБЗАР". Найповніша збірка. Унікальне колекційне видання преміум-класу

200 грн

[Додати в кошик](#)



#### Інтерстеллар

Інтерстеллар посів лише 5 місце у списку найкращих sci-fi фільмів (a Net)

350 грн / (50 грн)

[Додати в кошик](#)

Рис. 18 Сторінка “Товари”

Далі перед нами представлена сторінка з підписками, на ній розташовані підписки так само за категоріями товарів. У підписці вказується тип та її тривалість (Рис. 19).

**Media Store** [Головна](#) [Товари](#) [Підписки](#) [Контакти](#) [Панель керування](#) Паніна Олександра

## Підписки

[Головна](#) [Підписки](#)

### Підписки на будь-які категорії

Категорії

- Всі
- Ігри
- Фільми
- Книжки

|  |  |   |
|--|--|---|
| <p><b>Game Pass</b><br/>Basic</p> <p>200 грн / 30 днів</p> <p><a href="#">Додати в кошик</a></p> | <p><b>Game Pass</b><br/>Premium</p> <p>500 грн / 30 днів</p> <p><a href="#">Додати в кошик</a></p> | <p><b>Film Pass</b><br/>Normal</p> <p>600 грн / 30 днів</p> <p><a href="#">Додати в кошик</a></p> |
|--|--|---|

Рис. 19 Сторінка “Підписки”

Після додавання товарів до кошика, зверху в навігаційній панелі біля іконки кошика у нас буде збільшуватися лічильник товарів, які ми вибрали (Рис. 20).



Рис. 20 Лічильник товарів у кошику

Переходимо в кошик і в ньому у нас відображаються обрані нами товари, їх ціна, загальна ціна за все та кнопка оформлення замовлення(Рис. 21).

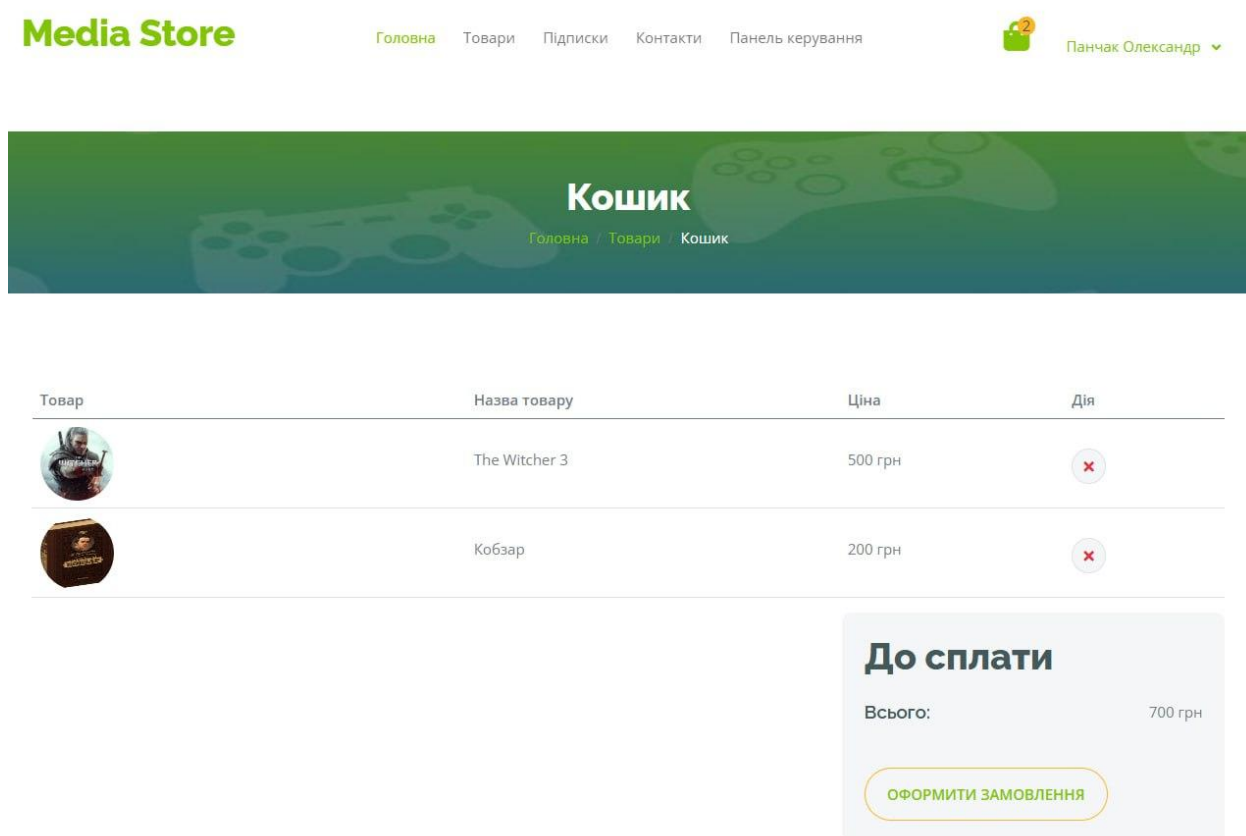


Рис. 21 Кошик користувача

Тут ми бачимо деталі замовлення, можемо заповнити поля, за замовчуванням підтягуються ім'я та пошта авторизованого користувача. Після того, як переглянули всі наші дані, можемо підтверджувати оплату. Оплата є з підтвердженням(Рис. 22) та оплата карткою(Рис. 23).



## Замовлення

[Головна](#) / [Товари](#) / [Кошик](#) / [Замовлення](#)

### Деталі замовлення

Ім'я \*

Панчак Олександр

Ел.Пошта \*

sashka@gmail.com

Телефон \*

| Товар | Назва товару  | Ціна    |
|-------|---------------|---------|
|       | The Witcher 3 | 500 грн |
|       | Кобзар        | 200 грн |
|       |               | 700 грн |

ПІДТВЕРДИТИ ЗАМОВЛЕННЯ

ОПЛАТИТИ КАРТКОЮ

Рис. 22 Підтвердження замовлення

## Безготівкова оплата

Деталі платежу

**Ім'я власника**

**Номер карти**

**CVC**      **Місяць придатності**      **Рік придатності**

123      MM      PPPP

**Сплатити (800 грн)**

[Повернутись](#)

Рис. 23 Безготівкова оплата

Тепер можемо перейти до панелі адміністратора, куди є доступ тільки користувачеві з правами адміністратора. На головній сторінці у нас представлена невелика статистика по магазину (Рис. 24).

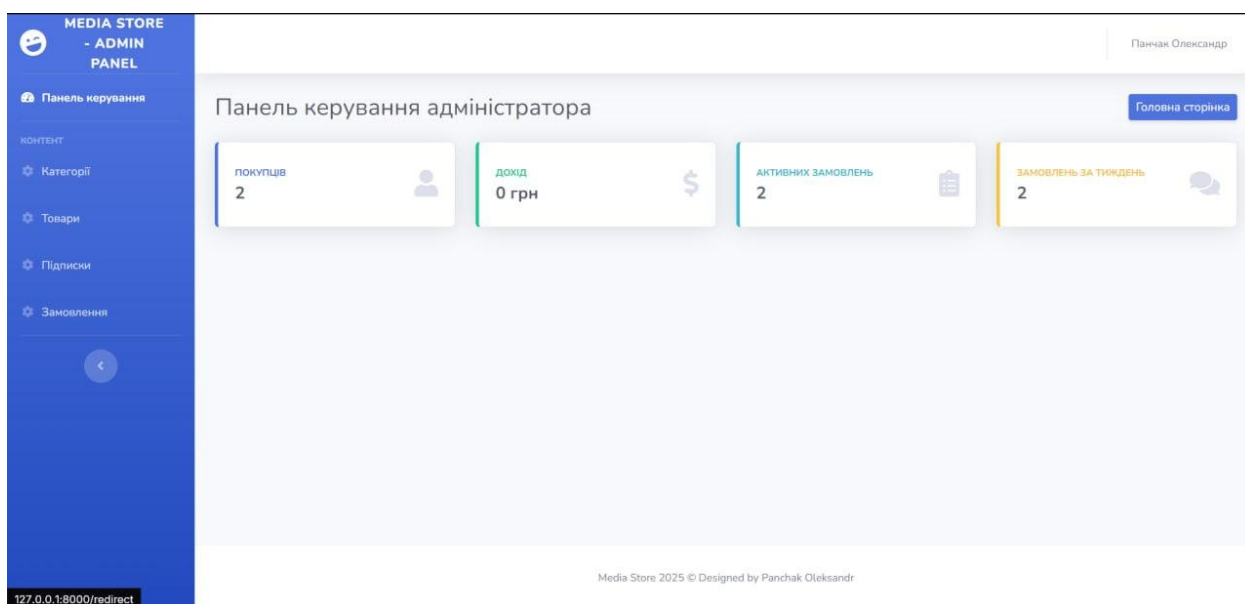


Рис. 24 Панель керування адміністратора

На сторінці з категоріями ми можемо створити категорію, чи вказати, чи буде доступна по ній підписка (Рис. 25).

The screenshot shows the 'Категорії' (Categories) page in the admin panel. It includes a 'Додати категорію' (Add category) button and a table listing existing categories:

| ID | Назва категорії | Підписка | Дія      |
|----|-----------------|----------|----------|
| 1  | Ігри            | Можлива  | Видалити |
| 6  | Фільми          | Можлива  | Видалити |
| 7  | Книги           | Можлива  | Видалити |

The footer at the bottom reads: 'Media Store 2025 © Designed by Panchak Oleksandr'.

Рис. 25 Сторінка категорій

На сторінці з товарами ми можемо створити товар, редагувати чи видалити товар із сайту(Рис. 26).

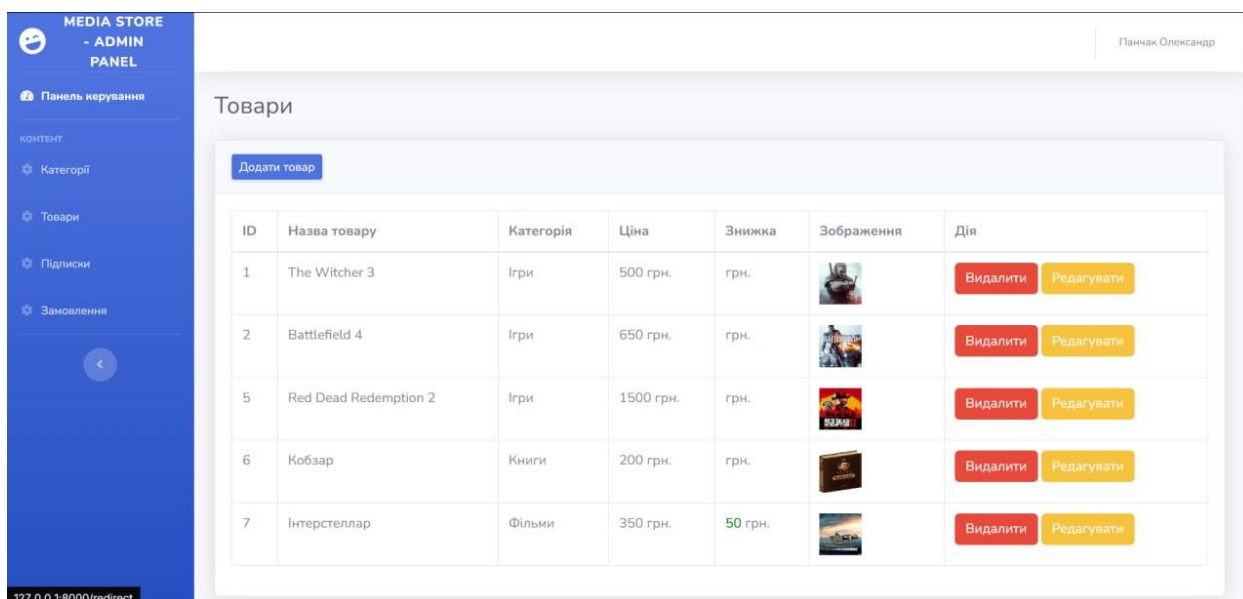


Рис. 26 Сторінка товарів

Додаючи товар, ми заповнюємо усі необхідні поля (Рис. 27).

Назва товару

Опис

Категорія

Обрати категорію

Ціна

Знижка

Зображення

Выберите файл Файл не выбран

Назад ДОДАТИ

Рис. 27 Сторінка створення товару

На сторінці з підписками у нас представлені підписки, вони поділяються так само на категорії та на типи підписок, які мають різні привілеї(Рис. 28).

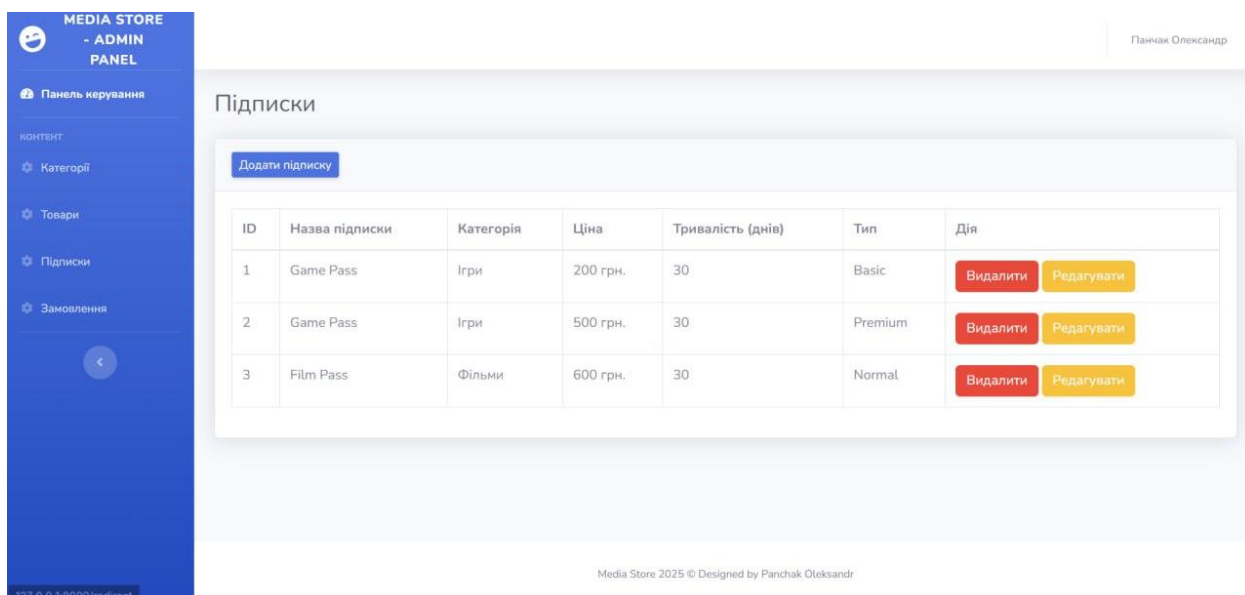


Рис. 28 Сторінка підписок

На сторінці із замовленнями(Рис. 29) адміністратор бачить усі підтвержені замовлення, які робили покупці. Він бачить всю інформацію клієнта, його товару і може підтвердити відправку замовлення, після чого цей товар з'явиться на сторінці користувача(Рис. 30).

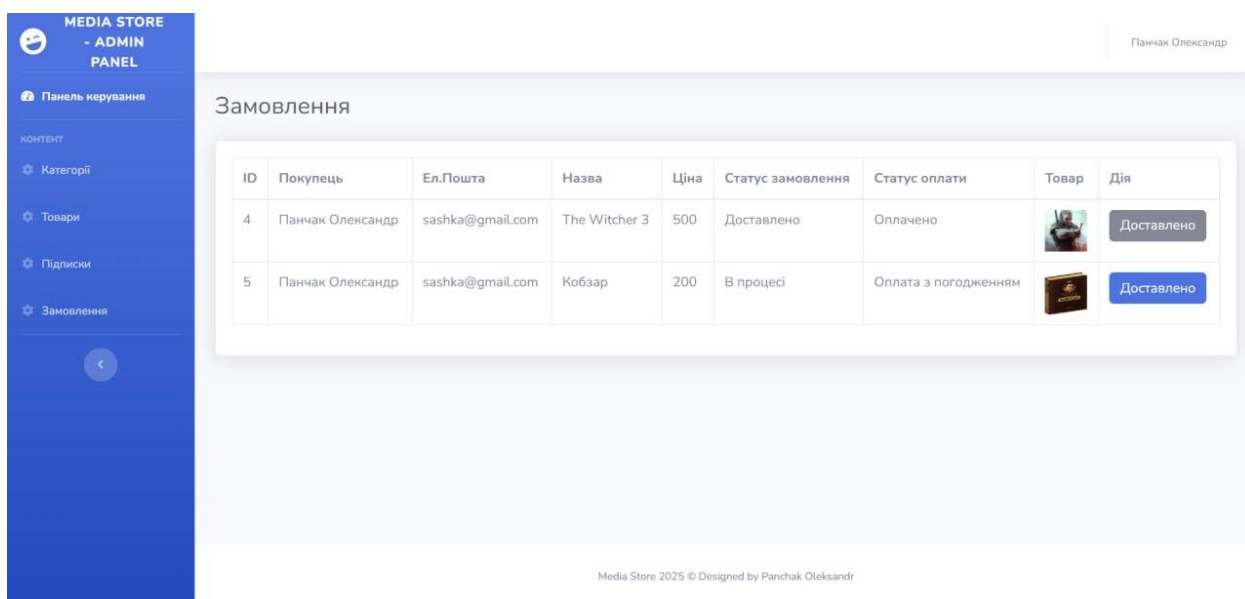


Рис. 29 Сторінка замовлень

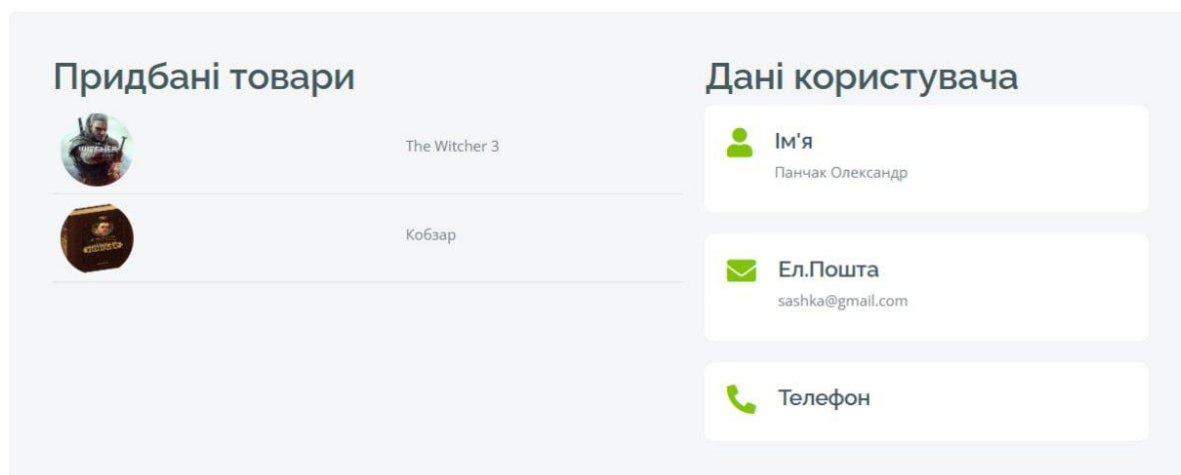
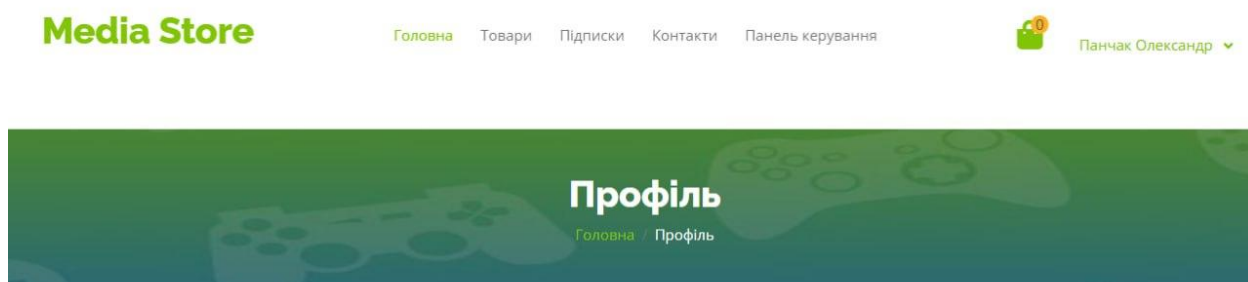


Рис. 30 Сторінка користувача

## ВИСНОВКИ

Розробка платформи електронної комерції для розповсюдження цифрового контенту є значним прогресом у тому, як користувачі отримують доступ і взаємодіють із медіа-продуктами, такими як електронні книги, аудіокниги та онлайн-курси. У цій дисертації ми досліджували важливі компоненти, необхідні для створення надійної, зручної та ефективної платформи, яка відповідає потребам як споживачів, так і постачальників контенту.

Ми почали з аналізу ключових вимог до платформи, включаючи функціональні та нефункціональні специфікації, які керують її проєктуванням і розробкою. Вибір технологій, зокрема вибір PHP і інфраструктури Laravel для розробки на стороні сервера, поряд із MySQL для керування базами даних, був зроблений для забезпечення оптимальної продуктивності, масштабованості та безпеки. Інтеграція HTML, CSS і JavaScript для інтерфейсу ще більше покращує взаємодію з користувачем, надаючи привабливий і чуйний інтерфейс.

Окрім технологічних аспектів, ми підкреслили важливість архітектури програмного забезпечення для ефективної організації компонентів системи. Впровадивши багаторівневу архітектуру, ми забезпечили чіткий розподіл проблем, що полегшує обслуговування та масштабованість. Використання діаграм, таких як діаграми варіантів використання, діаграми діяльності та діаграми класів, допомогло візуалізувати структуру системи та взаємодію, сприяючи більш чіткому розумінню функціональності платформи.

Вибір відповідних вимог до обладнання та програмного забезпечення також був критичним для підтримки роботи платформи. Ми окреслили необхідні специфікації для серверної інфраструктури, клієнтських пристроїв і програмних засобів для створення середовища, сприятливого для безперервного розповсюдження цифрового контенту.

Завершуючи цю тезу, важливо визнати потенційний вплив платформи електронної комерції на індустрію цифрового контенту. Забезпечуючи користувачам надійні та ефективні засоби для доступу та придбання цифрових медіа, платформа може посилити залучення користувачів, підвищити доступність контенту та підтримати творців контенту в охопленні ширшої аудиторії.

Заглядаючи вперед, майбутня робота може бути зосереджена на інтеграції розширених функцій, таких як персоналізовані рекомендації, моделі підписки та покращені заходи безпеки для подальшого покращення взаємодії з користувачем і адаптації до змінних вимог ринку. Постійний моніторинг і оновлення платформи будуть важливими для забезпечення її конкурентоспроможності та відповідності технологічним досягненням і очікуванням користувачів.

Таким чином, успішний розвиток платформи електронної комерції для розповсюдження цифрового контенту є не лише технологічним досягненням, але й кроком до трансформації того, як доступ до цифрових медіа є доступним і споживається. Цей проєкт закладає основу для майбутніх інновацій на цифровому ринку, сприяючи більш динамічному та інтерактивному середовищу як для користувачів, так і для творців.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Steam – [Електронний ресурс] – Режим доступу: <https://store.steampowered.com/?l=russian>
2. Apple Books – [Електронний ресурс] – Режим доступу: <https://www.ibooks.com>
3. The Top 10 Social Media Sites & Platforms – [Електронний ресурс] – Режим доступу: <https://www.searchenginejournal.com/social-media/social-media-platforms/>
4. Багаторівнева архітектура – [Електронний ресурс] – Режим доступу: <https://simpleone.ru/glossary/mnogourovnevaya-arhitektura/>
5. The Clean Architecture – [Електронний ресурс] – Режим доступу: <https://medium.com/clean-code-channel/clean-architecture-the-solution-to-have-a-reusable-flexible-and-testable-code-ac7e296d1a75>
6. Типи архітектури програмного забезпечення – [Електронний ресурс] – Режим доступу: <https://medium.com/nuances-of-programming/4-типа-архитектуры-программного-обеспечения-917133174724>
7. What is Unified Modeling Language (UML)? – [Електронний ресурс] – Режим доступу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
8. ПРОЄКТУВАННЯ ER-ДІАГРАМИ – [Електронний ресурс] – Режим доступу: <https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/>
9. Діаграма варіантів використання (UseCase diagram) – [Електронний ресурс] – Режим доступу: [https://flexberry.github.io/ru/fd\\_use-case-diagram.html](https://flexberry.github.io/ru/fd_use-case-diagram.html)
10. ПРОЄКТУВАННЯ USE CASE ДІАГРАМИ. ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СИСТЕМИ – [Електронний ресурс] – Режим доступу:

- <https://nationalteam.worldskills.ru/skills/proektirovanie-use-case-diagrammy-opredelenie-funktsionalnykh-vozmozhnostey-sistemy/>
11. What is Sequence Diagram? – [Електронний ресурс] – Режим доступу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
  12. UML - Activity Diagrams – Tutorialspoint – [Електронний ресурс] – Режим доступу: [https://www.tutorialspoint.com/uml/uml\\_activity\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_activity_diagram.htm)
  13. Побудова діаграми класів – [Електронний ресурс] – Режим доступу: [https://flexberry.github.io/ru/gpg\\_class-diagram.html](https://flexberry.github.io/ru/gpg_class-diagram.html)
  14. UML-діаграми класів – [Електронний ресурс] – Режим доступу: <https://prog-cpp.ru/uml-classes/>
  15. Entity Relationship Diagram - Data Modeling – [Електронний ресурс] – Режим доступу: <https://www.visualparadigm.com/VPGallery/datamodeling/EntityRelationshipDiagram.html>
  16. UML 2 Tutorial - Package Diagram - Sparx Systems – [Електронний ресурс] – Режим доступу: <https://sparxsystems.com/resources/tutorials/uml2/package-diagram.html>
  17. Документація Bootstrap (офіційний сайт) – [Електронний ресурс] – Режим доступу: [www.getbootstrap.com/documentation](http://www.getbootstrap.com/documentation).
  18. W3Schools – [Електронний ресурс] – Режим доступу: [www.w3schools.com](http://www.w3schools.com)
  19. Джонсон, Р. (2021). «Тенденції електронної комерції в індустрії цифрових товарів: аналіз динаміки ринку». Міжнародний журнал досліджень електронної комерції, 15 (2), 87-104.
  20. Патель, А. (2019). «Моделі цифрової підписки: порівняльне дослідження ефективності та задоволеності користувачів». Journal of Digital Media Management, 8(4), 231-248.
  21. Томпсон, Л. (2020). «Вплив мультимедіа на поведінку покупців в Інтернеті: мета-аналіз». Journal of Consumer Psychology, 25(3), 356-372.

22. Гарсія, М. (2018). «Тенденції, що розвиваються в онлайн-ігрових платформах: аналіз уподобань користувачів». Комп'ютери в розвагах, 16(4), 87-102.
23. Харріс, Е. (2022). «Майбутнє цифрового кінопрокату: думки експертів галузі». Journal of Media Economics, 30(1), 12-28.
24. Лі, С. (2020). «Схеми споживання електронних книг: огляд уподобань і поведінки читачів». Дослідження бібліотечної та інформаційної науки, 42 (2), 135-150.
25. Ван, Х. (2019). «Книжкові онлайн-магазини: порівняльний аналіз досвіду та задоволеності клієнтів». Міжнародний журнал управління інформацією, 36 (5), 785-800.
26. Мартінес, П. (2021). «Зростання послуг на основі передплати: наслідки для ринку цифрових товарів». Журнал бізнес-досліджень, 45 (3), 213-228.
27. Microsoft Docs. (2021). Layered architecture pattern – [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/en-us/azure/architecture/patterns/layered>
28. What is Component Diagram? – [Електронний ресурс] – Режим доступу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/>
29. Deployment Diagram in UML: Definition, Examples & Components – [Електронний ресурс] – Режим доступу: <https://study.com/academy/lesson/deployment-diagram-in-uml-definition-examples-components.html>
30. What is a data flow diagram? – [Електронний ресурс] – Режим доступу: <https://www.lucidchart.com/pages/data-flow-diagram>

## Фрагменти програмного коду. Функція створення товару

```
<?php

namespace App\Http\Controllers;

use App\Models\Product;
use App\Models\Category;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class ProductController extends Controller
{
    public function view_products()
    {
        $product = Product::all();
        return view('admin.product', compact('product'));
    }

    public function add_products()
    {
        $category = Category::all();
        return view('admin.add_products', compact('category'));
    }

    public function products_add(Request $request)
    {
        $product = new Product;
        $product->product_name=$request->product_name;
        $product->description=$request->description;
        $product->category=$request->category;
        $product->price=$request->price;
        $product->discount_price=$request->discount_price;

        $image = $request->image;
        $imagename = time().'.'.$image->getClientOriginalExtension();
        $request->image->move('product', $imagename);
        $product->image = $imagename;

        $product->save();
        return redirect()->back();
    }

    public function delete_products($id)
    {
        $data = Product::find($id);
        $data->delete();
        return redirect()->back()->with('message', 'Товар видалений');
```

```
}

public function update_products($id)
{
    $product = Product::find($id);
    $category = Category::all();
    return view('admin.update_products', compact('product', 'category'));
}

public function update_products_confirm(Request $request, $id)
{
    if(Auth::id())
    {
        $product = Product::find($id);

        $product->product_name=$request->product_name;
        $product->description=$request->description;
        $product->category=$request->category;
        $product->price=$request->price;
        $product->discount_price=$request->discount_price;

        $image = $request->image;
        if($image)
        {
            $imagename = time().'.'.$image->getClientOriginalExtension();
            $request->image->move('product', $imagename);
            $product->image = $imagename;
        }

        $product->save();
        return redirect()->back()->with('message', 'Товар редаговано');
    }
    else
    {
        return redirect('auth/login');
    }
}
}
```

## Фрагменти програмного коду. Функція створення замовлення

```
public function order()
{
    $user = Auth::user();
    $userId = $user->id;
    $category = Category::all();
    $product = Product::paginate(18);
    $data = Cart::where('user_id', '=', $userId)->get();

    foreach($data as $data)
    {
        $checkout = new Checkout;
        $checkout->name=$data->name;
        $checkout->email=$data->email;
        $checkout->phone=$data->phone;
        $checkout->user_id=$data->user_id;
        $checkout->product_name=$data->product_name;
        $checkout->price=$data->price;
        $checkout->image=$data->image;
        $checkout->product_id=$data->product_id;
        $checkout->payment_status='Оплата з погодженням';
        $checkout->delivery_status='В процесі';
        $checkout->save();

        $cart_id=$data->id;
        $cart = Cart::find($cart_id);
        $cart->delete();
    }
    return view('home/products', compact('category', 'product'));
}

public function stripe($totalprice)
{
    return view('home.stripe', compact('totalprice'));
}

public function stripePost(Request $request, $totalprice)
{
    Stripe::setApiKey(env('STRIPE_SECRET'));

    Charge::create ([
        "amount" => $totalprice * 100,
        "currency" => "UAH",
        "source" => $request->stripeToken,
        "description" => "Дякуємо за покупку"
    ]);
}
```

```
$user = Auth::user();
$userId = $user->id;
$category = Category::all();
$product = Product::paginate(18);
$data = Cart::where('user_id', '=', $userId)->get();

foreach($data as $data)
{
    $checkout = new Checkout;
    $checkout->name=$data->name;
    $checkout->email=$data->email;
    $checkout->phone=$data->phone;
    $checkout->user_id=$data->user_id;
    $checkout->product_name=$data->product_name;
    $checkout->price=$data->price;
    $checkout->image=$data->image;
    $checkout->product_id=$data->product_id;
    $checkout->payment_status='Оплачено';
    $checkout->delivery_status='В процесі';
    $checkout->save();

    $cart_id=$data->id;
    $cart = Cart::find($cart_id);
    $cart->delete();
}

return back()->with('success', 'Оплата пройшла успішно');
}
```