

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

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»
Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: «ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРНЕТ
МАГАЗИНУ КОМП'ЮТЕРНИХ ІГОР»

здобувача Бокея Гліба Вячеславовича _____

(ПІБ)

(Підпис)

Науковий керівник:

к.е.н., доцент

_____ Помазун О.М.

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

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

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

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

_____Помазун О.М.
“ ____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____Тішков Б.О.
“ ____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Бокея Гліба Вячеславовича

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи
на тему: «ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРНЕТ
МАГАЗИНУ КОМП'ЮТЕРНИХ ІГОР»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р.
№ 466- ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах

План кваліфікаційної бакалаврської роботи

Розділ I ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Розділ II РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Розділ III ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

Об'єкт дослідження Проєктування інформаційної системи інтернет-магазину комп'ютерних ігор

Предмет дослідження Проєктування архітектури та ключових компонентів системи

Мета кваліфікаційної бакалаврської роботи полягає в проєктуванні інтернет-магазину комп'ютерних ігор

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I надати стислу характеристику предметної галузі, показати її роль і місце в загальній системі, як наприклад галузі економіки. Привести особливості даної предметної галузі, які впливають на організацію інформаційної системи, комп'ютерної технології розв'язання задач. Навести схему організаційної структури.

У розділі II розробити вимоги до інформаційної системи, описати характеристику задачі, вихідну та вхідну інформацію і навести інформаційну модель задачі.

У розділі III описати інформаційне, технічне, програмне та організаційне забезпечення, описати структури інформаційних масивів, створити прототип та програмну реалізацію системи.

Завдання підготував

науковий керівник

_____ **Помазун Оксана Миколаївна**
« **10** » **березня 2025 р.**

Завдання одержав

здобувач

_____ **Бокей Гліб Вячеславович**
« **10** » **березня 2025 р.**

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

Бокея Гліба Вячеславовича

на тему

**«ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРНЕТ
МАГАЗИНУ КОМП'ЮТЕРНИХ ІГОР»**

- 1. Актуальність теми:** Багато людей купують товари онлайн тому що це швидко та зручно. Геймерська спільнота зростає а з нею і попит на комп'ютерні ігри. Інтернет-магазин комп'ютерних ігор це складний механізм який повинен забезпечити зручну навігацію, безпечні платежі та автоматичну доставку товару. Ринок ігор динамічний, і саме тому новинки з'являються кожен день у великій кількості і зростає попит.
- 2. Позитивні риси кваліфікаційної бакалаврської роботи:** актуальність теми, практична спрямованість, використання сучасних технологій, орієнтація на користувача, перспективність розвитку.
- 3. Наявність самостійних розробок автора:** Діаграми, таблиці, прототип, програмна реалізація інтернет-магазину.
- 4. Цінність теоретичних висновків та практичних рекомендацій:** її результати можуть бути використані під час подальшого навчання і в професійній діяльності у сфері інформаційних технологій.
- 5. Наявність недоліків:** відсутній детальний профіль користувача та платіжна система в програмній реалізації.
- 6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК:** Кваліфікаційна бакалаврська робота Бокея Гліба Вячеславовича відповідає тематиці, структурі та вимогам, передбаченим освітньою програмою. Представлені матеріали свідчать про сформованість фахових компетентностей здобувача. Робота заслуговує на позитивну оцінку та рекомендується до захисту перед екзаменаційною комісією.

Науковий керівник

доцент, проф Помазун О.М.

(підпис)

“ ” _____ 2025 р.

Рецензія

на кваліфікаційну бакалаврську роботу
здобувача вищої освіти

Бокея Гліба Вячеславовича

**Тема ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРНЕТ
МАГАЗИНУ КОМП'ЮТЕРНИХ ІГОР**

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення: Багато людей купують товари онлайн тому що це швидко та зручно. Геймерська спільнота зростає а з нею і попит на комп'ютерні ігри. Інтернет-магазин комп'ютерних ігор це складний механізм який повинен забезпечити зручну навігацію, безпечні платежі та автоматичну доставку товару. Ринок ігор динамічний, і саме тому новинки з'являються кожен день у великій кількості і зростає попит.

Якість проведеного дослідження: робота виконана послідовно, з дотриманням методологій, продемонстровано вміння аналізувати потреби користувачів, формулювання вимог до системи, застосування сучасних засобів моделювання та розробки.

Позитивні риси кваліфікаційної бакалаврської роботи: актуальність теми, практична спрямованість, використання сучасних технологій, орієнтація на користувача, перспективність розвитку.

Зауваження: відсутній детальний профіль користувача та платіжна система в програмній реалізації.

Практична значимість висновків і рекомендацій: результати розробки системи можуть бути використані під час подальшого навчання і в професійній діяльності у сфері інформаційних технологій.

Місце роботи та посада рецензента

ВСП «Фаховий коледж інформаційних систем і технологій КНЕУ імені
Вадима Гетьмана», завідувач відділенням техніко-інформаційних
спеціальностей _____ Правдивий Андрій Миколайович

(підпис, ПІБ)

Підпис засвідчую: завідувач відділенням техніко-інформаційних
спеціальностей _____

(посада, підпис)

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи

здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Проектування інформаційної системи інтернет-магазину
комп'ютерних ігор»

Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена автоматизації інформаційної системи інтернет-магазину комп'ютерних ігор, яку планується вирішувати та покращувати за допомогою сучасних інформаційних технологій.

Кваліфікаційна бакалаврська робота містить три розділи, які між собою логічно пов'язані.

У першому розділі представлено опис предметної області та об'єкта дослідження, проведено аналіз вирішуваних задач, а також міститься список наявних програмних інструментів у цій предметній галузі.

Другий розділ є проектним і має на меті обґрунтування методу проектування системи, створенню її архітектури, виконанню постановки задачі та розробці алгоритму для її розв'язання.

Третій розділ – будівельний. Представлено інформаційне, технічне програмне та організаційне забезпечення для системи інтернет-магазину комп'ютерних ігор. Окреслені аспекти організації інформаційного забезпечення: створена схема інформаційного забезпечення, представлені та описані в додатках формати вхідних та вихідних документів. Обґрунтований набір технічних пристроїв і програмного забезпечення. Зазначені структури інформаційних масивів, що застосовуються при розв'язанні задачі.

За підсумками роботи були сформульовані висновки стосовно створення інформаційної системи для інтернет-магазину комп'ютерних ігор.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 87 сторінок, 10 таблиць, 63 рисунків, список літератури з 30 найменувань, 4 додатків.

ІНФОРМАЦІЙНА СИСТЕМА ІНТЕРНЕТ-МАГАЗИНУ КОМП'ЮТЕРНИХ ІГОР

Перелік ключових слів: інформаційна система, прототип, діаграма, проєктування, методологія, автоматичний інжиніринг.

Предметом дослідження є проєктування архітектури та ключових компонентів системи.

Об'єктом дослідження виступає проєктування інформаційної системи інтернет-магазину комп'ютерних ігор.

Мета кваліфікаційної бакалаврської роботи полягає в проєктуванні інтернет-магазину комп'ютерних ігор.

Завданнями кваліфікаційної бакалаврської роботи є спроектувати та розробити інформаційну систему інтернет-магазину комп'ютерних ігор.

Апаратні та програмні засоби, що використовувались при проєктуванні: Clouinary, Visual Studio Code, MongoDB.

Результати досягнуті в процесі роботи: реалізована інформаційна система інтернет-магазину комп'ютерних ігор з функціоналом реєстрації/авторизації користувача, перегляд та додавання товарів до кошика, зміна/додавання товарів на сайт та оновлення ролей користувачів.

Одержані результати можуть бути використані у запуску повноцінного сайту для комерційної діяльності.

Рік виконання кваліфікаційної бакалаврської роботи 2025.

Рік захисту кваліфікаційної бакалаврської роботи 2025.

ЗМІСТ

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ	
ГАЛУЗІ.....	12
1.1 Характеристика предметної галузі та об'єкта дослідження.....	12
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі	18
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ	
ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	25
2.1. Аналіз і специфікація вимог до інформаційної системи	25
2.2. Постановка задачі	31
2.3. Моделювання інформаційної системи	38
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ	
СИСТЕМИ	52
3.1. Інформаційне забезпечення	52
3.2. Технічне забезпечення	59
3.3. Програмне забезпечення	62
3.4. Результати реалізації інформаційної системи.....	69
ВИСНОВКИ	82
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85
ДОДАТОК А	88
ДОДАТОК Б.....	91
ДОДАТОК В	104
ДОДАТОК Г.....	105

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ.

AMQP - Advanced Message Queuing Protocol

API - Application Programming Interface

CMS - Content Management System

CRM - Customer Relationship Management

DB - Database

DRM - Digital Rights Management

HTTP - Hypertext Transfer Protocol

KPI - Key Performance Indicators

MBSE - Model-Based Systems Engineering

OPM - Object-Process Methodology

UML - Unified Modeling Language

АРМ - Автоматизоване робоче місце

ЕОМ - Електронні обчислювальні машини

ІЗ - Інформаційне забезпечення

ІС - Інформаційна система

ІТ - Інформаційні технології

ООП - Об'єктно-орієнтоване програмування

СКБД - Система управління базами даних

ВСТУП

Сучасна ера інформаційних технологій вирізняється стрімким зростанням електронної комерції, що відкриває нові перспективи для підприємств та споживачів. Це також включає сферу комп'ютерних ігор, яка є однією з найшвидше розвиваючих областей розваг. Інтернет-магазини комп'ютерних ігор систематично покращують свої сервіси, прагнучи задовольнити потреби користувачів, які стають все більш вимогливими до якості обслуговування, функціоналу та зручності взаємодії з платформами.

Для успішного та ефективного функціонування практично будь-якої системи важливо здійснювати контроль, аналіз та впроваджувати автоматизацію для окремих процесів або для системи в цілому.

Метою цієї кваліфікаційної бакалаврської роботи є розробка інформаційної системи для інтернет-магазину комп'ютерних ігор, що суттєво підвищить якість обслуговування клієнтів і спростить процеси продажу комп'ютерних ігор, а також дозволить досягти найбільшої ефективності роботи мережі магазинів комп'ютерних ігор.

Але в значній мірі інтернет-магазин залежить від результативності його інформаційної системи, яка керує всіма основними процесами: від замовлення та сплати до доставки та обслуговування після отримання. Отже, важливим завданням є розробка інформаційної системи, що автоматизує бізнес-процеси, забезпечує персоналізацію сервісів, аналізує поведінку споживачів та прогнозує попит на товари.

Автоматизований інжиніринг інформаційних систем – це інноваційний метод створення та розробки ІС, що включає використання засобів автоматизованого проєктування, штучного інтелекту та машинного навчання для підвищення ефективності й адаптивності системи.

Розробка інтернет-магазину як спосіб підвищення ефективності та спрощення продажу товарів. Розробка такого веб-сайту не становить великої труднощі, оскільки в інтернеті є безліч різноманітних інструментів, які спрощують його створення та публікацію. З появою великої різноманітності

інтернет-магазинів і їхнього розвитку виник узагальнений вигляд і функції, які користувачі сподіваються знайти на сайті. З огляду на незвичайний дизайн і складний інтерфейс, це може відштовхувати користувачів. В інтернеті можна знайти безліч онлайн-магазинів на різні теми. Переважна частина аналогічних ресурсів, крім великих компаній, виглядає досить застаріло. Отже, потрібно шукати компроміс між загальними тенденціями та уникати ключових помилок.

Актуальність теми обумовлена тим, що використання автоматизованого інжинірингу дозволяє не лише пришвидшити процес розробки інформаційної системи, а й гарантує її масштабованість, надійність і ефективність. Це надзвичайно важливо в умовах суворої конкуренції на ринку електронної комерції, де успіх визначається здатністю оперативно реагувати на зміни та потреби покупців.

Для впровадження інформаційної системи застосовуватиметься мова SysML.

Для ілюстрації предметної області та проектованої системи застосовуватимемо UML-діаграми. Мова моделювання UML пропонує потужні інструменти для створення візуальних моделей і має чимало важливих переваг:

- UML є об'єктно-орієнтованим, тому способи опису результатів аналізу та проєктування семантично подібні до методів програмування сучасними об'єктно-орієнтованими мовами;
- UML дає можливість описати систему майже з усіх можливих перспектив та різноманітних аспектів функціонування системи;
- Діаграми UML доволі легкі для сприйняття після короткого знайомства з синтаксисом;
- UML став дуже популярним і активно розвивається.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Область дослідження зосереджена на цифровому ринку відеоігор - секторі світової розважальної індустрії, який стрімко розвивається. Ця галузь включає різноманітну екосистему, в якій переплітаються реальна поведінка користувачів, розподіл цифрового контенту, управління запасами, фінансові угоди та індивідуалізований контакт. В основі цього середовища лежить питання постійної адаптації до споживчих уподобань, нових тенденцій і технологічних стандартів, які розвиваються.

З економічного погляду, це завдання стосується оптимізації комерційних процесів інтернет-магазину, що передбачає пошук балансу між економічною ефективністю та комфортом користувачів. Це охоплює динамічні стратегії ціноутворення, керування обігом запасів, логістикою цифрового ліцензування, цілеспрямовані маркетингові активності та управління рекламними кампаніями. Метою є не лише збільшення прибутку, але й забезпечення утримання клієнтів, удосконалення операційних робочих процесів та зміцнення лояльності до бренду на ринку, що стає дедалі більш насиченим.

Інтернет-магазин комп'ютерних ігор функціонує в більш широких комерційних та технологічних рамках, що охоплюють: промисловість, компанії та бізнес, організації та установи та технологічні об'єкти.

Прийняття рішень в цій цифровій екосистемі є одночасно стратегічним і тактичним. Проблемні ситуації часто з'являються внаслідок змінних ринкових вимог, навантаження на сервер, шахрайських дій, конфліктів з ліцензуванням та вузьких місць у ланцюгах постачання цифрових ключів. Розв'язання цих питань має бути узгоджене з бізнес-цілями, серед яких підвищення конверсії, поліпшення користувацького досвіду та забезпечення цілісності транзакцій.

Внутрішні дані включають: історії покупок, аналітику поведінки користувачів, журнали продуктивності сервера, показники оборотності запасів, показники повернення коштів та відгуки клієнтів.

Зовнішні дані включають: ринкові тенденції, цінову політику та тактику просування конкурентів, настрої в соціальних мережах, агрегатори оглядів ігор, оновлення законодавства щодо цифрових продажів.

До емпіричних типів даних відносяться: спостережувані дії користувачів, показники навантаження на систему, до об'єктивних: журнали транзакцій, звіти платіжних шлюзів, показники активації ліцензій.

Джерелами надходження цієї інформації є CRM-системи, API-канали від паблішерів, платформи веб-аналітики, інструменти соціального прослуховування та зовнішні бази даних маркетингових досліджень.

Підходами для збирання та оброблення є: моніторинг вподобань користувачів та нових тенденцій в режимі реального часу, автоматизація рутинних рішень за допомогою бізнес-правил та штучного інтелекту, застосування досвіду домену та емпіричних стратегій у випадках, коли дані є неповними, запуск прогнозних моделей для оцінки рекламних стратегій або стрибків навантаження, регресійний аналіз, кластеризація, когортний аналіз, сегментація доходів і для рекомендаційних систем виявлення шахрайства та динамічного ціноутворення.

На рисунку 1.1 відображена схема організаційної структури інтернет-магазину комп'ютерних ігор.



Рисунок 1.1 - Схема організаційної структури інтернет-магазину комп'ютерних ігор

Джерело: сформовано автором на основі виконаного дослідження

Директор стратегічний лідер та особа, що приймає остаточні рішення. Він контролює загальний напрямок роботи магазину; встановлює цілі щодо доходів, інновацій та зростання. Технічний директор контролює архітектуру, інновації та операційну діяльність інформаційної системи. Він керує розробкою та безпекою системи, координує команду ІТ та розробників, забезпечує інтеграцію сторонніх інструментів, масштабованість та високу доступність. Команда ІТ та розробки розробляє бекенд/фронтенд функціональність, підтримує структуру та безпеку баз даних, впроваджує автоматизацію, платіжну інтеграцію та логіку CMS. Відділ маркетингу, підвищує видимість платформи, трафік і залученість. Він планує та проводить цифрові кампанії, використовує аналітику для оптимізації залучення та утримання користувачів, контролює виробництво контенту та PR. Команда SEO та PR оптимізує видимість магазину в пошукових системах та керує комунікаціями бренду. Команда аналітиків відстежує KPI, поведінку користувачів, ефективність кампаній та прогнозування. Продуктовий відділ керує цифровим каталогом та користувацьким інтерфейсом платформи. Він визначає асортимент та життєвий цикл ігор, розробляє користувацький досвід та макет магазину, тестує нові функції перед запуском. UX/UI дизайнери забезпечують зручну навігацію, сучасні стандарти інтерфейсу. Команда забезпечення якості проводить тестування, щоб запобігти помилкам у готовому вигляді. Відділ підтримки клієнтів відповідає за всю післяпродажну взаємодію з користувачами. Він вирішує скарги, запити на повернення коштів та проблеми з акаунтами, підтримує канали зв'язку, збирає відгуки користувачів для майбутніх покращень. Команда колл-центру надає допомогу користувачам у режимі реального часу на всіх платформах.

Цифрові ключі в інтернет-магазинах часто дешевші, ніж ігри у звичайних магазинах з кількох причин: Відсутність витрат на виробництво, жодних витрат на розповсюдження, конкуренція на цифровому ринку, пропозиції та знижки.[20]

Відсутність витрат на виробництво. Цифрові ключі не вимагають виробництва, пакування та розповсюдження фізичних носіїв, таких як CD або

DVD. Натомість цифрові ключі поширюються через Інтернет, що усуває витрати, пов'язані з виробництвом і розповсюдженням фізичних носіїв.

Жодних витрат на розповсюдження. Цифрові ключі розповсюджуються через Інтернет, що означає відсутність витрат, пов'язаних із розповсюдженням у звичайних магазинах або роздрібних мережах. Це економить витрати на дистрибуцію.

Конкуренція на цифровому ринку. На цифровому ринку, де розповсюджуються цифрові ключі, існує велика конкуренція між різними платформами, такими як Steam, GOG або Epic Games Store. Введення нижчих цін на ігри може залучити більше клієнтів і збільшити продажі.

Пропозиції та знижки. Цифрові магазини часто пропонують різні типи акцій, пропозицій і знижок на Digital Keys, що приваблює клієнтів і збільшує продажі. Багато цифрових платформ також пропонують програми лояльності, які винагороджують постійних клієнтів і пропонують їм додаткові знижки на наступні покупки.

Є магазини, яким дуже довіряють і які отримали визнання серед геймерів, пропонуючи ключі для таких платформ, як Steam, GOG або Origin. Ці платформи мають добру репутацію, оскільки мають тривалий досвід продажу цифрових ігор і забезпечують безпечну та зручну платформу для клієнтів.

Більшість великих інтернет-магазинів комп'ютерних ігор відрізняється професіоналізмом, керується досвідченою командою та пропонує найвищу якість продажів. Вони також пропонують захист покупців та інші гарантії, щоб ваші покупки завжди були безпечними.

Проте, як і в будь-якій іншій галузі, є магазини, які не заслуговують на довіру та можуть продавати нелегальні чи підроблені копії ігор. Варто звернути увагу на думки інших клієнтів, відгуки про магазин і його репутацію серед ігрової спільноти. Перед покупкою варто перевірити, чи є магазин легально діючою компанією, чи має відповідні сертифікати та ліцензії.

Загалом, більшість авторитетних ігрових магазинів заслуговують на довіру та пропонують якісні продукти та гарне обслуговування клієнтів.

Ціни можуть бути нижчими, ніж у стаціонарних магазинах . У багатьох випадках офіційні продавці пропонують нижчі ціни на цифрові ключі, ніж їхні аналоги у звичайних магазинах. Крім того, у разі періодичних акцій і розпродажів ціни на ігри можуть ще більше впасти, що дозволяє істотно заощадити.

Доступність гри. Купуючи Digital Keys, ви уникаєте проблем, пов'язаних з відсутністю в стаціонарних магазинах. Це означає, що вам не потрібно виходити з дому, щоб отримати улюблену назву. Все, що вам потрібно зробити, це завантажити гру з цифрового магазину та почати грати.

Швидкість і зручність. Купуючи ключ для Digital Keys, ви отримуєте його відразу після оплати. Вам не доведеться чекати доставки або стикатися з проблемами при установці гри. Усе доступне онлайн, тобто ви можете насолоджуватися грою за лічені хвилини.

Захист авторських прав. Купуючи ключ до гри в офіційних продавців, ви можете бути впевнені, що він легальний і не порушує авторські права. Купуючи ігри на сірих ринках, ви ризикуєте отримати незаконний ключ, що може призвести до проблем з піратством.

Цифрові ключі мають кілька переваг, які роблять їх кращими за традиційні коробкові ігри, зокрема наступні: Миттєвий доступ, зручність і економія місця, економія часу і грошей, можливість гри на різних пристроях, носій не зношується.

Миттєвий доступ. Цифрові ключі доступні онлайн, тобто ви можете придбати та завантажити їх будь-коли та будь-де, не виходячи з дому чи шукаючи ігровий магазин.

Зручність і економія місця. Цифрові ключі не займають місце на полиці, їх не потрібно переміщати, немає ризику втрати або пошкодження диска. Це економить місце у вашому домі.

Економія часу і грошей. Коли ви купуєте Digital Keys, вам не потрібно платити за доставку та чекати на доставку. Ви також можете користуватися різноманітними пропозиціями та знижками, доступними лише для цифрових ігор.

Можливість гри на різних пристроях. Цифрові ключі можна грати на різних пристроях, таких як комп'ютери, консолі, смартфони та планшети. Це дозволяє зручно і практично використовувати гру на різних пристроях залежно від ваших потреб.

Носій не зношується. Цифрові ключі не піддаються зношенню на таких носіях, як CD або DVD, тобто немає ризику, що гра стане нечитабельною через подряпини чи пошкодження.

Легальні ігрові ключі кращі з кількох причин: Захист авторських прав, безпека, оновлення та підтримка, гарантія якості, підтримка ігрової індустрії.

Захист авторських прав. Купуючи легальні ключі до гри, ви можете бути впевнені, що не порушуєте авторські права та підтримуєте розробників ігор. Купуючи незаконні ключі, ви ризикуєте отримати фінансові штрафи, а також порушити авторські права.

Безпека. Легітимні ігрові ключі видаються офіційними продавцями, які дбають про безпеку своїх клієнтів. Купуючи законні ключі, ви не ризикуєте отримати заражене програмне забезпечення або ключ, який не працює.

Оновлення та підтримка. Купуючи законні ключі, ви отримуєте доступ до оновлень гри та технічної підтримки, які пропонує виробник гри. Якщо з грою виникнуть проблеми, ви зможете зв'язатися з розробником і отримати допомогу.

Гарантія якості. Виробник гри гарантує якість свого продукту. Купуючи законні ключі, ви можете бути впевнені, що отримаєте якісний продукт, який працює, як описано.

Підтримка ігрової індустрії. Купуючи законні ігрові ключі, ви підтримуєте ігрову індустрію, яка сприяє розвитку культури та розваг у всьому світі. Підтримка розробників ігор означає більше інвестицій у майбутнє, а отже, більше інновацій та розвитку ігрової індустрії.

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

Купівля ключів до комп'ютерних ігор за значно нижчими цінами, ніж роздрібні копії, може викликати кілька питань.[19] Ось розбивка ситуації:

1. Законні ключові торгові посередники.
 - Авторизовані торговельні посередники: деякі веб-сайти мають право продавати ключі до гри за зниженими цінами. Вони часто мають партнерські стосунки з видавцями чи розробниками, що дозволяє їм пропонувати розпродажі або пакетні угоди.
 - Розпродажі та акції: сезонні розпродажі (наприклад, розпродажі в Steam, Чорна п'ятниця тощо) можуть призвести до значних знижок. Деякі посередники використовують ці акції, щоб продавати ключі за нижчими цінами.
2. Ключові посередники сірого ринку.
 - Перепродаж ключів: деякі веб-сайти працюють на сірому ринку, де вони продають ключі, отримані різними способами, як-от розбіжності в регіональних цінах або оптові покупки.
 - Ризики: покупка в цих джерелах може бути ризикованою. Ключі можуть бути отримані сумнівними способами, наприклад за допомогою викрадених кредитних карток, або можуть бути заблокованими в регіоні, що обмежує їх використання.
3. Ризики дешевших ключів.
 - Блокування облікових записів: якщо видавець гри виявить, що ключ було отримано шахрайським шляхом або продається з порушенням його умов, він може заборонити ключ або обліковий запис, який його використовує.
 - Відсутність підтримки: купівля в неавторизованих продавців може означати, що ви не отримуватимете підтримку клієнтів або оновлення від видавця. В таблиці 1.1 наведений аналіз характеристик існуючих інтернет-магазинів комп'ютерних ігор.

*Таблиця 1.1 - Аналіз характеристик існуючих інтернет-магазинів
комп'ютерних ігор*

№з/п	Можливості	Steam	GOG	Epic Games Store	Origin
Основні функції	Способи представлення даних	Так	Так	Так	Так
1	Перегляд каталогу ігор	Так	Так	Так	Так
2	Фільтрація та пошук	Так	Так	Так	Так
3	Опис та рейтинг ігор	Так	Так	Так	Так
4	Підтримка клієнтів	Так	Так	Так	Так
5	Оновлення ігор	Так	Так	Так	Так
6	Відгуки користувачів	Так	Так	Ні	Ні

Продовження Таблиці 1.1

7	Рейтинги	Так	Так	Ні	Так
8	Відеоогляди	Так	Ні	Ні	Так
9	Демо-версії ігор	Так	Ні	Ні	Так
10	Системні вимоги	Так	Так	Так	Так
11	Рекомендації на основі історії покупок	Так	Ні	Ні	Так
12	Історія цін на ігри	Так	Ні	Ні	Ні
13	Знижки та акції	Так	Так	Так	Так
14	Інтеграція з соцмережами	Так	Так	Так	Так
15	Підтримка модифікацій	Так	Так	Ні	Ні

Джерело: сформовано автором на основі виконаного дослідження

Проаналізувавши популярні системи для продажу комп'ютерних ігор, такі як Steam, Epic Games Store, GOG та інші, можна зробити такі висновки:

Steam надає великий функціонал для управління іграми та спілкування з користувачами, включаючи систему досягнень, обговорень і майстерні модів. Водночас, для розробників ігор існує чималий комісійний збір за використання платформи.

Epic Games Store зосереджується на ексклюзивних релізах і забезпечує вигідні умови для розробників, але має обмежений набір функцій для користувачів, порівняно зі Steam, наприклад, відсутність форумів або досягнень.

GOG виділяється тим, що пропонує DRM-free ігри, що є перевагою для користувачів, які не бажають прив'язуватися до платформи, проте їх асортимент значно менший порівняно з конкурентами.

До позитивних характеристик існуючих систем відносяться:

1. Steam
 - Великий асортимент ігор, включаючи інді-проекти та класичні релізи.
 - Система спільноти: форуми, майстерня модів, обговорення ігор, а також досягнення і трейдинг картки.
 - Високий рівень захисту транзакцій і даних користувачів.
 - Регулярні розпродажі та акції, що залучають нових користувачів.
2. Epic Games Store
 - Вигідні умови для розробників (комісія лише 12% проти 30% у Steam).
 - Великий каталог ексклюзивних ігор.
 - Безкоштовні щотижневі ігри для користувачів, що підвищує залучення аудиторії.
3. GOG
 - DRM-free моделі ігор, що дозволяє користувачам володіти продуктами без необхідності підключення до платформи.
 - Зручна система для покупки класичних ігор з налаштованим під сучасні системи програмним забезпеченням.
 - Простий і зрозумілий інтерфейс з акцентом на права користувачів.
4. Origin
 - Origin пропонує доступ до ексклюзивних ігор від Electronic Arts, таких як серії FIFA, The Sims, Battlefield та інші, які можуть бути недоступні в інших магазинах.
 - Origin пропонує підписку EA Play, яка дозволяє користувачам грати у нові ігри, отримувати ранній доступ та ексклюзивні бонуси, що значно економить кошти для тих, хто активно грає у продукти EA.

- Інтерфейс Origin інтуїтивно зрозумілий та зручний для користувачів, що робить процес купівлі та запуску ігор простим і швидким.
- Деякі ігри на Origin мають демо-версії, що дозволяє користувачам протестувати гру перед покупкою.
- Origin дозволяє зберігати дані про прогрес у хмарі, що зручно для доступу з різних пристроїв або для відновлення прогресу у разі перевстановлення гри чи ОС.
- Origin регулярно проводить акції, що дозволяють придбати ігри зі значними знижками, особливо під час свят та розпродажів.

Негативні характеристики існуючих систем:

1. Steam

- Висока комісія для розробників, що може бути недоцільним для інді-студій.
- Перевантаженість інтерфейсу через велику кількість функцій і можливостей, що ускладнює користування для новачків.
- Система повернення грошей обмежена лише двома годинами ігрового часу.

2. Epic Games Store

- Відсутність розширених соціальних функцій, таких як форуми, досягнення та система модів.
- Порівняно невеликий асортимент ігор.
- Користувачі критикують платформу за обмежену кастомізацію інтерфейсу та функціоналу.

3. GOG

- Менший вибір сучасних ігор порівняно зі Steam або Epic Games Store.
- Відсутність значної соціальної складової, як у Steam (досягнення, форуми, майстерня модів).
- Платформа більше орієнтована на класичні та інді-ігри, що може не задовольняти широкий сегмент геймерів.

4. Origin

- Origin орієнтований в основному на ігри від EA, тому його бібліотека значно менша, ніж у таких платформ, як Steam або GOG.
- Порівняно з Steam, Origin має менш розвинені соціальні функції (менше можливостей для комунікації, немає спільнот і форумів, обмежена підтримка модифікацій).
- Origin не має вбудованої підтримки модифікацій для ігор, що може бути недоліком для користувачів, які хочуть кастомізувати свій ігровий досвід.
- Оскільки Origin є платформою від Electronic Arts, користувачі можуть бути обмежені політикою компанії. Наприклад, деякі ігри можуть бути доступні тільки через підписку або мати обмежений доступ у певних країнах.
- Origin не має системи відгуків і оцінок від інших користувачів, що ускладнює можливість дізнатися про якість ігор перед покупкою.
- Хоча Origin надає хмарне збереження та захист від несанкціонованого доступу, використання DRM може викликати незручності для гравців, оскільки деякі ігри вимагають постійного з'єднання з інтернетом для верифікації.

Торговці з власними онлайн-магазинами повністю управляють досвідом своїх клієнтів. Продавці мають можливість використовувати свій веб-ресурс, щоб прямісінько взаємодіяти з клієнтами та спонукати до придбань.

Жоден онлайн-магазин не може функціонувати без фінансових транзакцій. Необхідний широкий вибір розповсюджених методів оплати, щоб споживачі завжди могли без труднощів виконати покупку.

Слід надати можливість клієнтам реєструватися в інтернет-магазині та зберігати свої платіжні дані, щоб спростити майбутні покупки зручнішими.

Інтернет-магазин повинен залучати трафік для здійснення покупок. Основою генерації трафіку є оптимізація для пошукових систем. Враховуючи ключові включення SEO-термінів у назві та описі товарів підвищує ймовірність підняття сайту вищі позиції в результатах пошуку. Збільшена видимість є основним аспектом елементом для підвищення відвідуваності сайту та зростання кількості конверсії.

У зв'язку з підвищенням використанням телефонів для здійснення покупок в Інтернеті, потрібно впевнитися, що оформлення сайту буде комфортним для мобільних гаджетів, щоб споживачі могли здійснювати покупки, не зважаючи на те, який пристрій вони використовують.

Інтернет-магазини забезпечують повну самостійність у досвіді споживачів. Замість того, щоб слідувати ринковій політиці, можна відкрити власний інтернет-магазин на своїх умовах, щоб формувати взаємодію з продуктом своїх користувачів.

З такою свободою можна стимулювати більше конверсій, якщо знати, як розробити кожен компонент магазину. Слідуючи визначеним стратегіям, можна глибше усвідомити, як покупці сприймають різні аспекти магазину, та використати ці знання для його вдосконалення. Створення інтернет-магазину згідно з бажаннями та інтересами основних покупців забезпечує перевагу над менш уважними суперниками.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Аналіз і специфікація вимог до інформаційної системи

Мова моделювання систем (SysML) допомагає командам проєктувати, розробляти, тестувати та розгортати складні фізичні системи. SysML підтримує підхід системної інженерії на основі моделей (MBSE) до збору інформації в системних моделях. MBSE використовує моделі домену для покращення потоку інформації за допомогою різноманітних інструментів, що включають різні аспекти складної системи.

Діаграма вимог представляє системні вимоги. Це допомагає користувачам зрозуміти масштаби системи, визначити відповідних зацікавлених сторін і зв'язати вимоги з різними компонентами чи елементами системи.

Бізнес-вимоги — це ясні твердження, що окреслюють функції, можливості та риси, які програмне забезпечення або система повинна демонструвати для задоволення потреб бізнесу.

Ці потреби можуть включати багато аспектів, зокрема функціональність, ефективність, безпеку та взаємодію з користувачем.

Бізнес-вимоги усувають проміжок між зацікавленими сторонами бізнесу та командою розробників, гарантуючи відповідність кінцевого продукту стратегічним цілям організації.

Бізнес-вимоги виходять за межі програмного забезпечення, містять специфікації фізичних продуктів, виробничих процесів та стандартів якості. Вони охоплюють деталі про матеріали, розміри, допуски й інші важливі чинники, необхідні для успішного створення продукту.

До того ж, вимоги виробничого бізнесу здатні включати масштабованість, економічну ефективність і відповідність галузевим нормам.

За допомогою мови SysML була створена діаграма бізнес-вимог для інформаційної системи інтернет-магазину комп'ютерних ігор (див. рисунок 2.1).

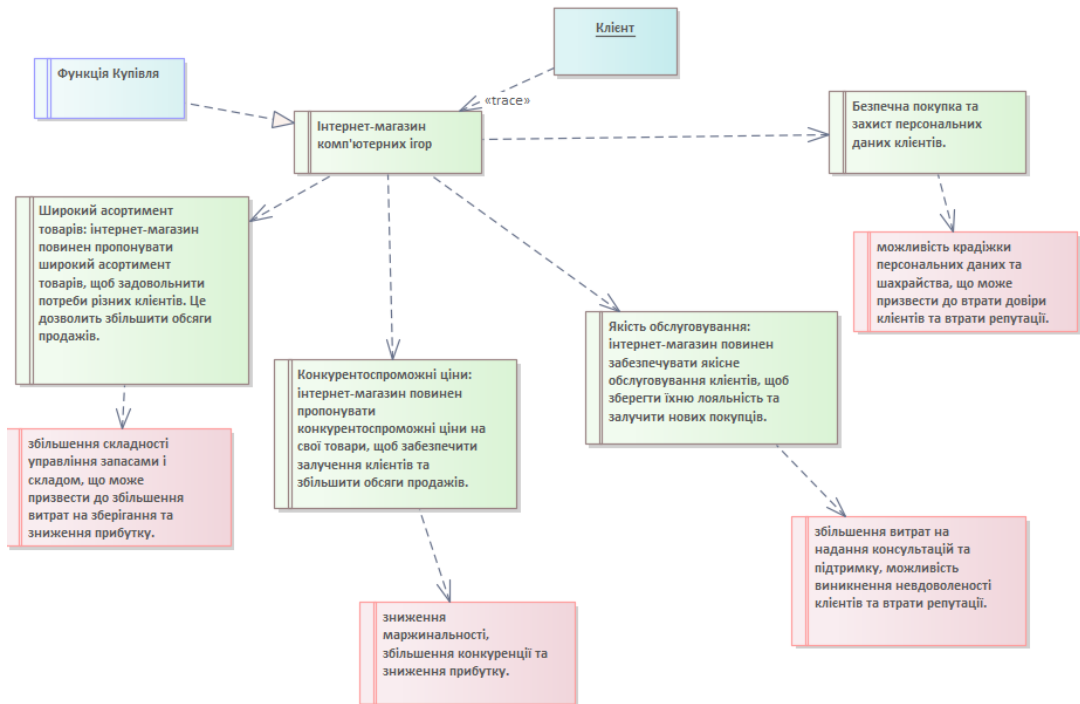


Рисунок 2.1 - Діаграма бізнес-вимог

Джерело: сформовано автором на основі виконаного дослідження

До бізнес вимог інформаційної системи інтернет-магазину комп'ютерних ігор відносяться: безпечна покупка та захист персональних даних, якість обслуговування, конкурентноспроможні ціни, широкий асортимент товарів.

Розробка бізнес-вимог зазвичай передбачає взаємодію між бізнес-аналітиками, зацікавленими особами та експертами в галузі незалежно від сфери. Ця співпраця має на меті врахувати конкретні потреби, обмеження та сподівання бізнесу, забезпечуючи зрозумілу дорожню карту для подальших етапів розробки або виробництва.

Менеджер специфікацій - це особливий і дієвий інструмент, що дає вигляд електронної таблиці або текстового редактора, який можна використовувати для управління будь-яким елементом, хоча він особливо корисний під час роботи з вимогами, які завжди мають описовий текст, щоб детально описати потребу. Нові вимоги можна створювати з назвами і

докладними описами, а такі властивості, як Статус і Пріоритет, можна додавати або змінювати з випадających переліків. Існуючі вимоги можна переглядати і керувати ними у зручному вигляді - наприклад, у вигляді діаграм і вікон - а їх зміна в Диспетчері специфікацій змінить їх в усіх інших місцях репозиторію.

Специфікація бізнес вимог представлена на рисунку 2.2.

☒ Безпечна покупка та захист персональних даних клієнтів.	BusinessRe...	Proposed	Medium	Medium
☐ збільшення витрат на надання консультацій та підтримку, можливість виникнення невдоволеності клієнтів та втрати репутації.	Validate	Proposed	Medium	Medium
☐ збільшення складності управління запасами і складом, що може призвести до збільшення витрат на зберігання та зниження прибутку.	Validate	Proposed	Medium	Medium
☐ зниження маржинальності, збільшення конкуренції та зниження прибутку.	Validate	Proposed	Medium	Medium
☒ Інтернет-магазин комп'ютерних ігор	BusinessRe...	Proposed	Medium	Medium
☐ Клієнт		Proposed		

Рисунок 2.2 - Специфікація бізнес вимог

Джерело: сформовано автором на основі виконаного дослідження

Функціональні вимоги визначають поведінку та функціональність системи. Вони описують дії, які система має виконувати у відповідь на певні вхідні дані або за конкретних умов. Функціональні вимоги зосереджуються на тому, що система мусить робити, аби задовольнити потреби користувача й досягти поставлених цілей.

Ключові характеристики функціональних вимог:

- Поведінкові описи - вони визначають взаємодію між системою та її користувачами або іншими системами.
- Бізнес-правила - вони визначають операційні правила та обмеження, які керують процесами системи.
- Обробка даних - цей розділ описує вимоги до введення, виведення, зберігання та обробки даних.
- Взаємодія з користувачами - тут детально описані можливості та

функції, доступні для користувачів, включаючи дизайн інтерфейсу користувача та робочі процеси.

Моделювання функціональних потреб передбачає створення уявлень про поведінку та взаємодію системи. Цього можна досягти за допомогою різних методів та інструментів, котрі допомагають візуалізувати та роз'яснити вимоги. Діаграма функціональних вимог системи представлена на рисунку 2.3.

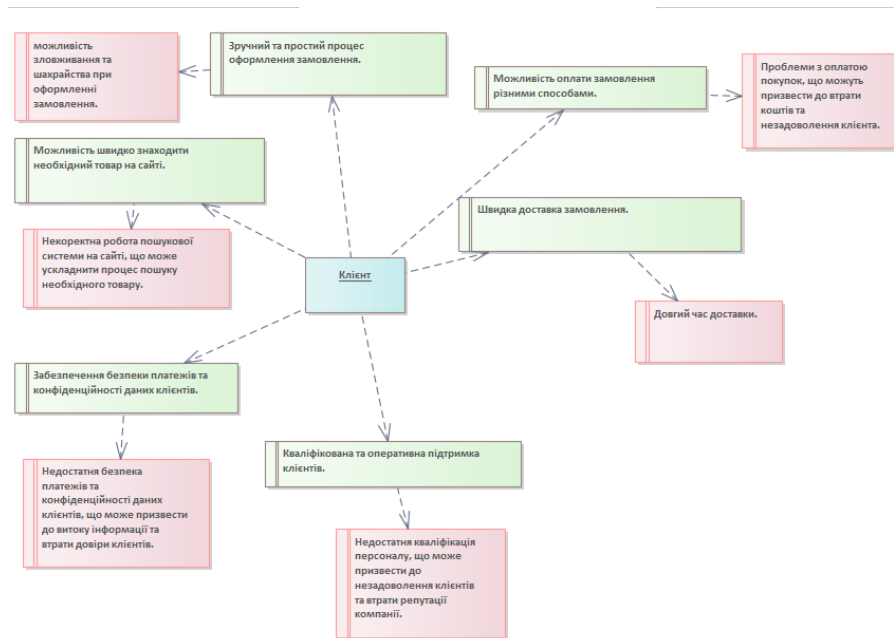


Рисунок 2.3 - Діаграма функціональних вимог

Джерело: сформовано автором на основі виконаного дослідження

До функціональних вимог інформаційної системи інтернет-магазину комп'ютерних ігор відносяться: зручний та простий процес оформлення замовлення, можливість оплати замовлення різними способами, швидка доставка замовлення, кваліфікована та оперативна підтримка клієнтів, забезпечення безпеки платежів та конфіденційності даних клієнтів, можливість знаходити необхідний товар на сайті.

Специфікація функціональних вимог представлена на рисунку 2.4.

Item	Stereotype	Status	Difficulty	Priority
☒ Гнучкість вибору часу та місця доставки замовлення.	Functional...	Proposed	Medium	Medium
☐ Довгий час доставки.	Validate	Proposed	Medium	Medium
☒ Забезпечення безпеки платежів та конфіденційності даних клієнтів.	Functional...	Proposed	Medium	Medium
☒ Зручний та простий процес оформлення замовлення.	Functional...	Proposed	Medium	Medium
☒ Кваліфікована та оперативна підтримка клієнтів.	Functional...	Proposed	Medium	Medium
☐ Клієнт		Proposed		
☐ можливість затримки або втрати товарів під час доставки.	Validate	Proposed	Medium	Medium

Рисунок 2.4 - Специфікація функціональних вимог

Джерело: сформовано автором на основі виконаного дослідження

Нефункціональні вимоги окреслюють атрибути якості системи, нормативи продуктивності та експлуатаційні обмеження. Вони описують, як система має себе вести, а не що вона має робити. Нefункціональні вимоги забезпечують зручність, надійність, продуктивність та безпеку системи, сприяючи загальному досвіду користувачів та підтримці системи.

Ключові характеристики нефункціональних вимог:

- Продуктивність - визначає час відгуку системи, пропускну здатність, введення/виведення та використання ресурсів.
- Безпека - визначає заходи, яких повинна дотримуватися система, включаючи захист даних, автентифікацію та протоколи авторизації.
- Зручність використання - стосується дизайну інтерфейсу користувача, доступності та простоти використання.
- Надійність - це здатність системи правильно функціонувати протягом тривалого часу, включаючи доступність, стійкість до збоїв, ризиків і катастроф.
- Масштабованість - це діапазон, в якому система може витримувати збільшені навантаження або розширюватися для майбутнього зростання та розвитку продукту.

- Відповідність - визначає дотримання нормативних, правових та галузевих стандартів.

Нефункціональні потреби, як звичайно, менш очевидні для візуального представлення в порівнянні з функціональними потребами, але певні діаграми та моделі все ж таки можуть ефективно відображати й передавати ці вимоги. Ці діаграми допомагають зацікавленим особам зрозуміти продуктивність, зручність використання, безпеку та інші атрибути якості системи. Діаграма нефункціональних вимог системи представлена на рисунку 2.5.

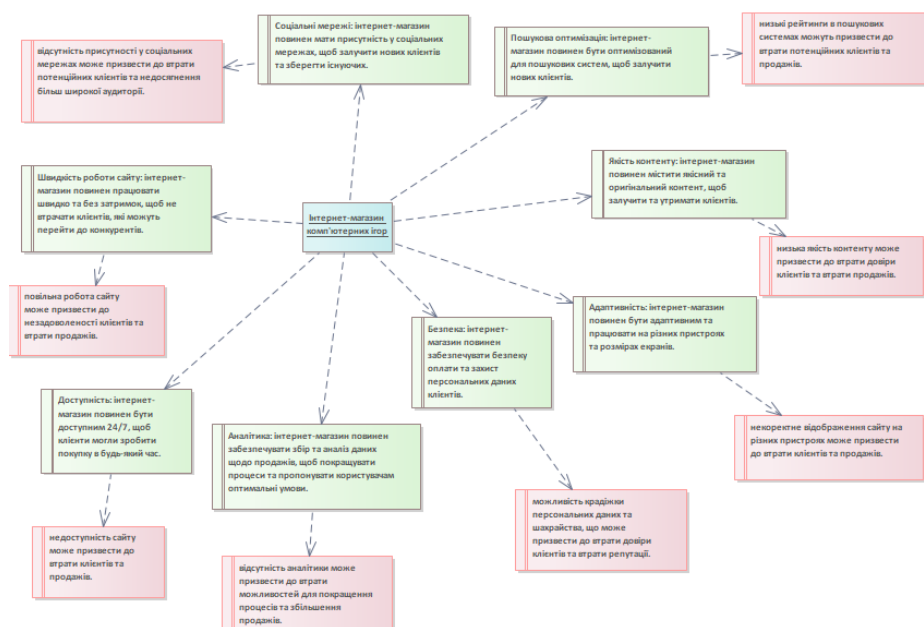


Рисунок 2.5 - Діаграма нефункціональних вимог

Джерело: сформовано автором на основі виконаного дослідження

До нефункціональних вимог інформаційної системи інтернет-магазину комп'ютерних ігор відносяться: соціальні мережі, пошукова оптимізація, якість контенту, адаптивність, безпека, аналітика, доступність, швидкість роботи.

Специфікація нефункціональних вимог представлена на рисунку 2.6.

Item	Stereotype	Status	Difficulty	Priority
☒ Адаптивність: інтернет-магазин повинен бути адаптивним та працювати на різних пристроях та розмірах екранів.	Nonfuncti...	Proposed	Medium	Medium
☒ Аналітика: інтернет-магазин повинен забезпечувати збір та аналіз даних щодо продажів, щоб покращувати процеси та пропонувати користувачам оптимальні умови.	Nonfuncti...	Proposed	Medium	Medium
☒ Безпека: інтернет-магазин повинен забезпечувати безпеку оплати та захист персональних даних клієнтів.	Nonfuncti...	Proposed	Medium	Medium
☐ відсутність аналітики може призвести до втрати можливостей для покращення процесів та збільшення продажів.	Validate	Proposed	Medium	Medium
☐ відсутність присутності у соціальних мережах	Validate	Proposed	Medium	Medium

Рисунок 2.6 - Специфікація нефункціональних вимог

Джерело: сформовано автором на основі виконаного дослідження

2.2. Постановка задачі

Інформаційна система "Інтернет-магазин комп'ютерних ігор" призначена для організації процесу продажу комп'ютерних ігор через онлайн-платформу. Вона забезпечує зручний доступ до ігор для споживачів, автоматизує управління каталогом товарів, обробку замовлень, платіжні транзакції та взаємодію з клієнтами.

Сутність задачі полягає в оптимізації процесів продажу, управління запасами та взаємодії з користувачами. Використання електронних обчислювальних машин (ЕОМ) дозволяє зменшити час обробки замовлень, покращити точність даних, знизити витрати на обслуговування клієнтів і підвищити ефективність маркетингових кампаній.

Розв'язання задачі за допомогою ЕОМ є необхідним для:

- Швидкого оброблення великої кількості замовлень.
- Автоматизації обліку товарів та управління запасами.
- Проведення аналітики для оптимізації асортименту та цінової політики.

- Забезпечення безпеки фінансових транзакцій.

Об'єктами управління в цій задачі є:

- Комп'ютерні ігри (каталог товарів).
- Клієнти (користувачі інтернет-магазину).
- Замовлення (історія покупок).
- Платіжні системи (обробка платежів).

Вихідна інформація включає:

- Звіти про продажі (для аналізу ефективності).
- Статистика замовлень (для управління запасами).
- Інформація про клієнтів (для маркетингових кампаній).
- Аналітика ринку (для прийняття стратегічних рішень).

Періодичність: Щоденна обробка замовлень; щомісячні звіти про продажі та аналіз ринку.

Термін видачі: Інформація про замовлення видається миттєво; аналітичні звіти - в межах 24 годин після завершення звітного періоду.

Автоматизоване розв'язання задачі може бути припинено у випадках:

- Технічних збоїв у системі.
- Вимкнення електропостачання або проблем з інтернет-з'єднанням.

- Зупинки підтримки програмного забезпечення.

Задача "Інтернет-магазин комп'ютерних ігор" пов'язана з:

- Задачами управління запасами на складі.
- Задачами фінансового обліку та аналізу.
- Задачами маркетингового аналізу та планування.
- Задачами обслуговування клієнтів (підтримка та консультації).

Персонал:

- Обробка запитів клієнтів (служба підтримки).
- Аналіз звітів та прийняття рішень (менеджери з продажу).
- Упровадження нових ігор у каталоги (контент-менеджери).

Технічні засоби:

- Автоматизація обробки замовлень та платежів.
- Генерація звітів та аналітики.
- Управління базою даних товарів та клієнтів.

Схема інформаційної моделі задачі показана на рисунку 2.7.

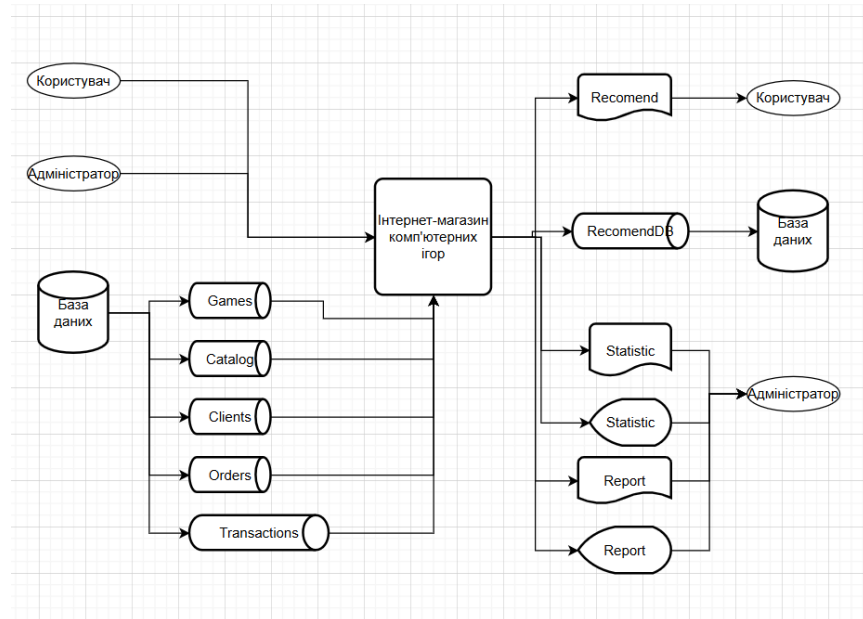


Рисунок 2.7 - Інформаційна модель задачі

Джерело: сформовано автором на основі виконаного дослідження

Персоналізовані рекомендації допомагають користувачеві знайти ігри, що можуть його зацікавити, підвищуючи ймовірність покупки.

Система аналізує історію покупок користувача, перегляди та інші дії на сайті для створення рекомендацій. Алгоритми машинного навчання або прості фільтри за категоріями допомагають видати релевантні ігри в режимі реального часу за запитом користувача.

Звіт надає адміністратору повний список замовлень за день, що дозволяє контролювати кількість продажів, аналізувати популярність продуктів та планувати оновлення запасів.

Дані про замовлення автоматично накопичуються в базі даних після завершення кожної транзакції. Щодня в кінці робочого дня формується автоматичний звіт, який експортується у файл і надається адміністратору. В таблиці 2.1 наведений перелік вихідних повідомлень.

Таблиця 2.1 - Перелік вихідних повідомлень

№з/п	Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувачі інформації
1	Рекомендації ігор для користувача	Recomend_games	Екранна форма	За запитом користувача	Миттєво	Користувачі
2	Звіт про замовлення	Order_report	Файл	Щодня	Протягом 5 хв після завершення дня	Адміністратор
3	Статистика продажів	Sell_stat	Файл	Щотижня	Кожного понеділка	Адміністратор
4	Звіт про наявність товарів		Файл	Щодня	На початку дня	Адміністратор
5	Рекомендації щодо популярних ігор	Pop_games	Екранна форма	За вимогою	Миттєво	Користувач
6	Звіт про знижки та акції	Discount_report	Файл	Щотижня	Кожної п'ятниці	Адміністратор

Джерело: сформовано автором на основі виконаного дослідження

Призначення вхідної інформації для інтернет-магазину комп'ютерних ігор полягає в отриманні даних про користувачів, ігри та замовлення. Ця інформація може бути отримана через реєстрацію та оформлення замовлення.

Способи одержання вхідної інформації включають текстові запити, голосові команди, вибір з меню або кнопок у інтерфейсі та інтеграцію з базою даних.

Перелік і опис вхідних повідомлень показаний в таблиці 2.2.

Таблиця 2.2 - Перелік вхідних повідомлень

№з /п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Періодичність видання	Термін і частота надходження	Джерело
1	Дані про користувачів	UserData	Реєстраційні дані в базі даних	При реєстрації нового клієнта	Постійно, в режимі реального часу	База даних
2	Каталог ігор	Games_catalog	Електронний каталог в форматі XML/JSON	Щоденне оновлення	Щоденно	Адміністратор
3	Інформація про нові ігри	New_Games	Електронний каталог в форматі XML/JSON	Щотижневе оновлення	Щотижня	Адміністратор
4	Інформація про замовлення	Orders	Електронні замовлення в форматі XML/JSON	При оформленні нового замовлення	Постійно, в режимі реального часу	База даних
5	Дані про платежі	Payments	Електронні платіжні транзакції в форматі XML/JSON	При оплаті замовлення	Постійно, в режимі реального часу	База даних

Джерело: сформовано автором на основі виконаного дослідження

Інтернет-магазин комп'ютерних ігор для забезпечення ефективної роботи використовує різноманітні вхідні дані, які надходять з внутрішніх та зовнішніх джерел. Головна мета цієї інформації полягає в: ідентифікації користувачів та персоналізації взаємодії, створенні структури каталогу товарів

і демонстрації доступного контенту, слідкуванні за новинами та сучасним асортиментом, обробці замовлень і платіжних операцій у реальному часі, гарантуванні цілісності баз даних і аналізу поведінки користувачів. В таблиці 2.3 наведений перелік масивів використовуваної інформації.

Таблиця 2.3 - Перелік масивів використовуваної інформації

Масив	Ідентифікатор	Максимальна кількість записів
1	UserData	Необмежено
2	Games_catalog	До 10000 записів
3	New_games	До 1000 записів
4	Orders	Необмежено
5	Payments	Необмежено

Джерело: сформовано автором на основі виконаного дослідження

Виходячи з аналізу вхідних даних, система створює ряд результатів, що мають вирішальне значення для процесу прийняття рішень, управління магазином у реальному часі та взаємодії з споживачами. Ці дані розраховані на: пропозиції сервісом індивідуальних рекомендацій для користувачів, створення щоденних і щотижневих звітів для адміністратора, аналітику з продажів і прогнозування на основі статистики, інформування споживачів про знижки, акції та затребувані товари, контроль за наявністю та кількістю товару. В таблиці 2.4 наведений перелік масивів результатної інформації.

Таблиця 2.4 - Перелік масивів результатної інформації

Масив	Ідентифікатор	Максимальна кількість записів
1	Recommend_games	Діамічно
2	Order_report	1 запис на день
3	Sell_stat	1 запис на тиждень
4	Pop_games	До 100 записів
5	Discount_report	1 запис щоп'ятниці

Джерело: сформовано автором на основі виконаного дослідження

При розв'язанні задачі можуть бути використані наступні показники:

1. Розрахунок загальної вартості замовлення

Формула для розрахунку загальної вартості замовлення:

$$C = \sum_{i=1}^n P_i * Q_i - D$$

де:

C — загальна вартість замовлення,

P_i — ціна i -ї гри,

Q_i — кількість i -ї гри,

D — знижка (якщо є).

2. Розрахунок доходу від продажів

Формула для розрахунку загального доходу від продажів за певний період:

$$R = \sum_{j=1}^m C_j$$

де:

R — загальний дохід,

C_j — вартість j -го замовлення,

m — кількість замовлень за період.

3. Розрахунок середньої вартості замовлення

Формула для розрахунку середньої вартості замовлення:

$$AOV = \frac{R}{N}$$

де:

AOV — середня вартість замовлення,

R — загальний дохід,

N — кількість замовлень.

4. Розрахунок коефіцієнта конверсії

Конверсія — це відношення числа відвідувачів сайту, які виконали на ньому якісь цільові дії, до загального числа відвідувачів сайту, виражене у відсотках.

Формула для розрахунку коефіцієнта конверсії:

$$CR = \frac{O}{V} * 100\%$$

де:

CR — коефіцієнт конверсії,

O — кількість завершених замовлень,

V — кількість відвідувачів сайту.

5. Розрахунок прибутку

Формула для розрахунку прибутку:

$$P = R - C_i$$

де:

P — прибуток,

R — загальний дохід,

C_i — загальні витрати

2.3. Моделювання інформаційної системи

Найважливішим кроком на шляху до вдосконалення мов програмування стала поява об'єктно-орієнтованого підходу до програмування (або, скорочено, ООП) та відповідного класу мов. Саме дослідження теорії та практики проєктування і реалізації програмних систем за принципами ООП і є основною метою другої частини даного курсу.[28]

При об'єктно-орієнтованому підході програма являє собою опис об'єктів, їх властивостей (або атрибутів), сукупностей (або класів), відносин між ними, способи їх взаємодії та операцій надоб'єктами (або методи).

Безперечною перевагою даного підходу є концептуальна близькість до предметної області довільної структури та призначення. Механізм

спадкоємства атрибутів і методів дозволяє будувати похідні поняття на основі базових і таким чином створювати модель як завгодно складної предметної області з заданими властивостями.

Фундаментальними поняттями об'єктного підходу є поняття класу і об'єкту. Об'єкт – відчутна сутність, що чітко виявляє свою поведінку. Кожний об'єкт характеризується властивостями (розмір, колір, смак і таке інше) та діями (ходити їсть, сидить і таке інше). За збігом основних ознак ми можемо виділити групи об'єктів – класи. Клас – це сукупність об'єктів одного типу, тобто схожих за своїми властивостями та призначенням. Іншими словами, кожен об'єкт є екземпляром певного класу.

З точки зору програмної реалізації об'єкт – це сукупність полів даних та методів (процедур або функцій) їх обробки. Таким чином, при описі програмного об'єкту ми задаємо перелік полів та описуємо заголовки методів. Полями класу являються змінні оголошені усередині класу, вони призначені для зберігання даних під час роботи екземпляра класу (об'єкту). Обмежень на типи полів в класі не передбачено.

На рисунку 2.8 зображена діаграма прецедентів, що в узагальненому вигляді документує функціональні вимоги у вигляді варіантів використання як типової взаємодії системи з акторами.

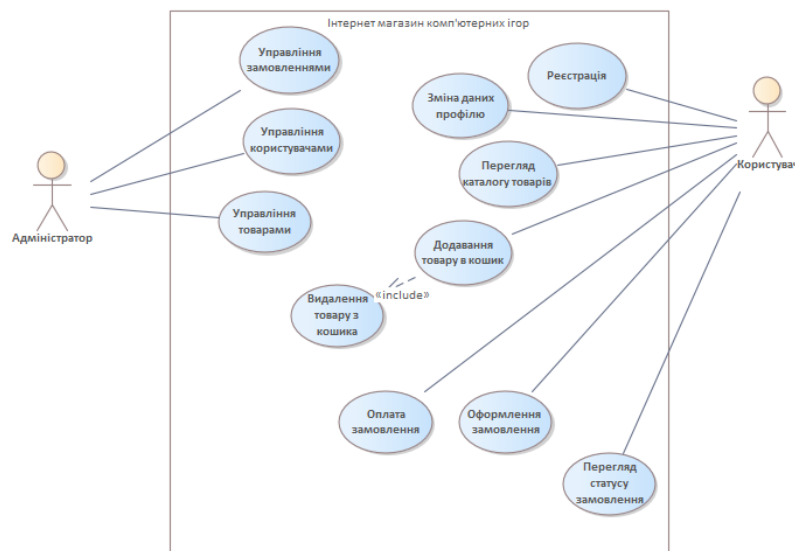


Рисунок 2.8 – Діаграма прецедентів

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності - застосовуються для моделювання логіки варіантів використання, показуючи відомості, що передаються між об'єктами в системі під час реалізації сценарію.

Діаграма послідовності показує, як процеси або об'єкти взаємодіють під час реалізації сценарію. На діаграмі відображаються класи, необхідні для виконання сценарію, та повідомлення, які вони пересилають один одному (спричинені кроками у варіанті використання).

На рисунку 2.9 діаграма послідовності ілюструє процес взаємодії між користувачем, системою та кошиком під час додавання товару до кошика в інтернет-магазині.

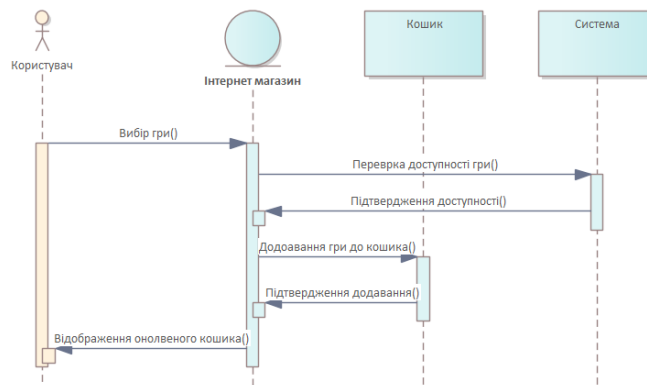


Рисунок 2.9 - Діаграма послідовності додавання до кошика

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.10 діаграма послідовності описує процес взаємодії адміністратора із системою при оновленні каталогу товарів в інтернет-магазині.

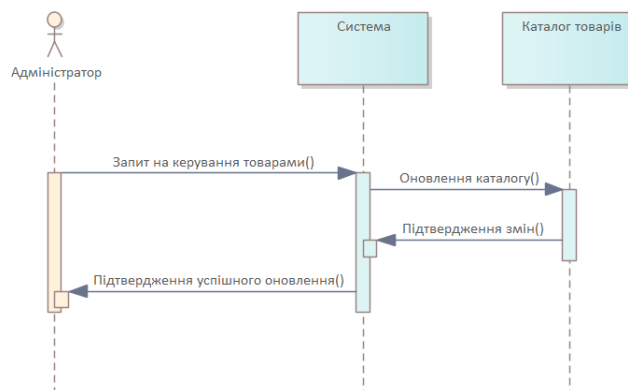


Рисунок 2.10 - Діаграма послідовності управління товарами

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.11 діаграма послідовності описує процес взаємодії користувача із системою при оплаті замовлення в інтернет-магазині.

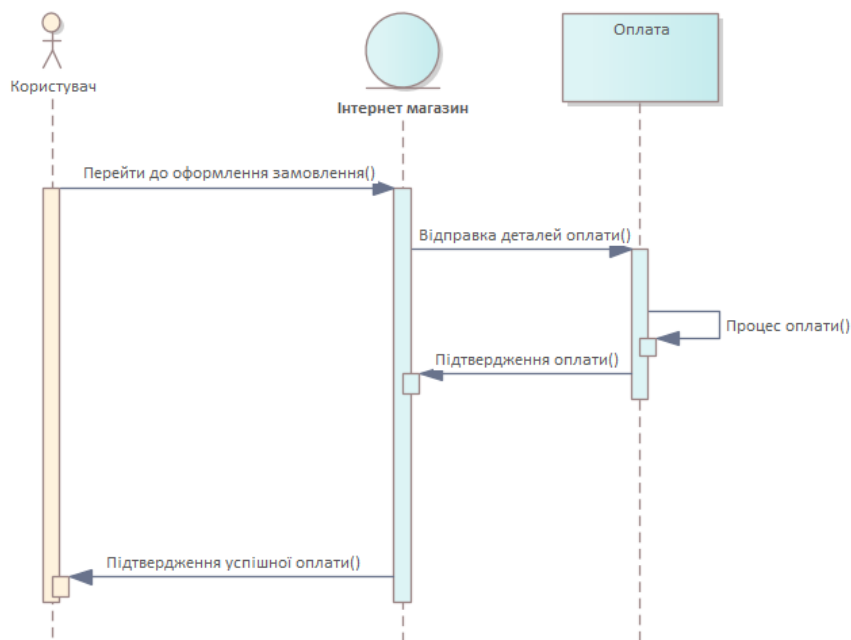


Рисунок 2.11 - Діаграма послідовності оплати

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.12 діаграма послідовності описує процес взаємодії користувача із кошиком при створенні замовлення в інтернет-магазині.

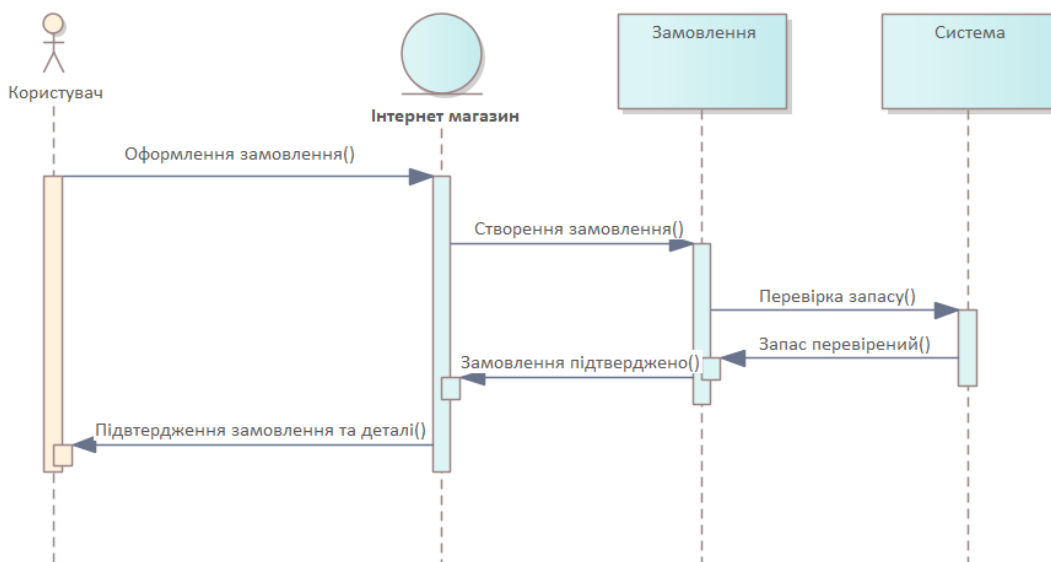


Рисунок 2.12 - Діаграма послідовності замовлення

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.13 діаграма послідовності описує процес взаємодії користувача із системою при реєстрації в інтернет-магазині.

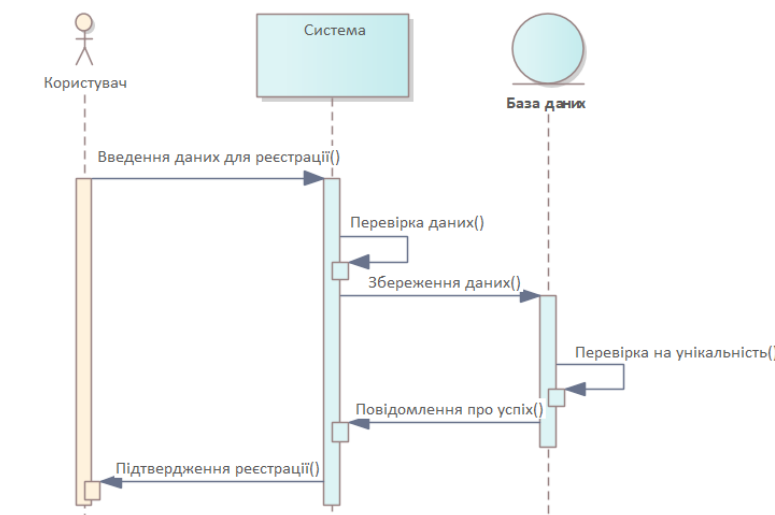


Рисунок 2.13 - Діаграма послідовності реєстрації

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.14 діаграма послідовності описує процес взаємодії користувача із системою при авторизації в інтернет-магазині.

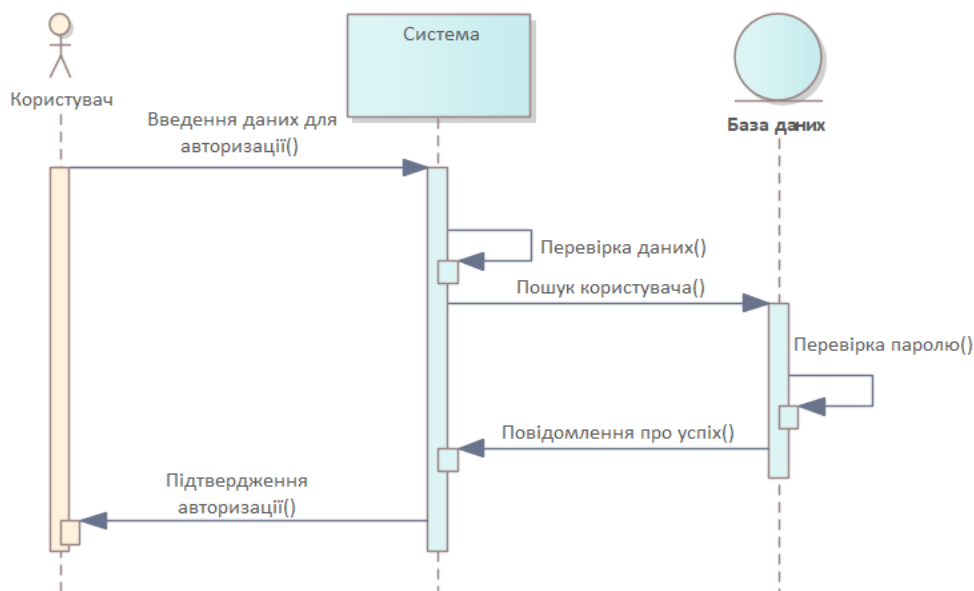


Рисунок 2.14 - Діаграма послідовності авторизації

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.15 діаграма послідовності описує процес взаємодії користувача із системою при перегляді каталогу товарів в інтернет-магазині.

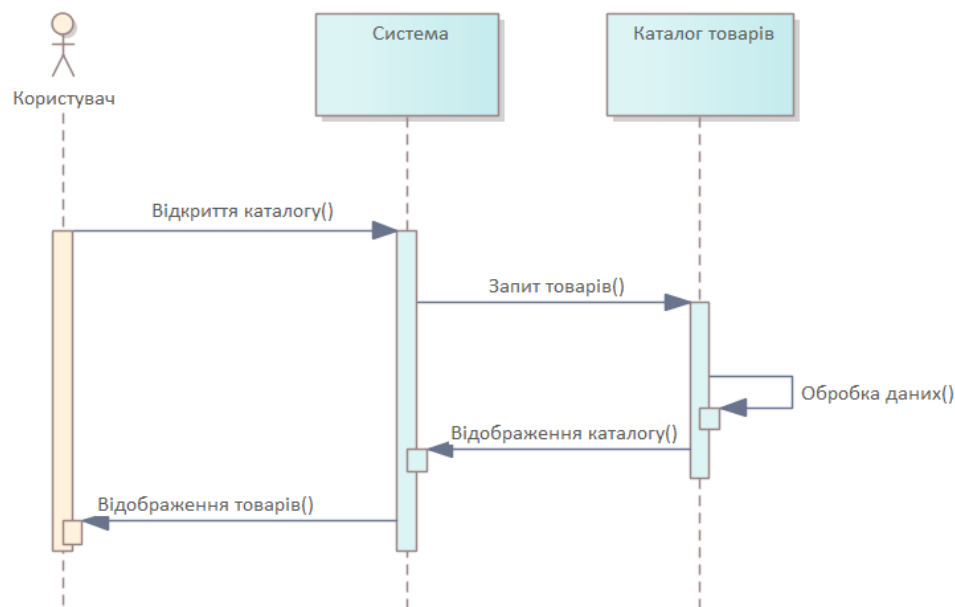


Рисунок 2.15 - Діаграма послідовності перегляду каталогу товарів

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.16 діаграма послідовності описує процес взаємодії адміністратора із системою при управлінні користувачами в інтернет-магазині.

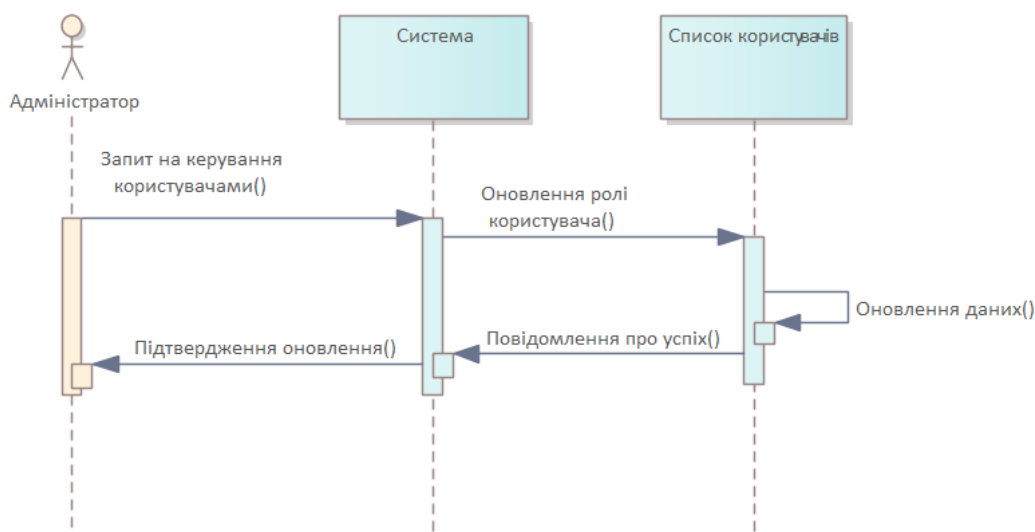


Рисунок 2.16 - Діаграма послідовності управління користувачами

Джерело: сформовано автором на основі виконаного дослідження

Діаграми класів - це вид UML-діаграм, який використовується в програмній інженерії для візуалізації структури та зв'язків між класами в системі, тобто для створення та уявлення об'єктно-орієнтованих систем.

Діаграми класів пропонують загальний вигляд на дизайн системи, сприяючи комунікації та документуванню архітектури програмного забезпечення. Вони є основним засобом об'єктно-орієнтованого проєктування і мають ключове значення у життєвому циклі розробки програмного забезпечення.

В об'єктно-орієнтованому програмуванні (ООП) клас виступає як схема або модель для генерації об'єктів. Об'єкти є реалізаціями класів, і кожен клас задає набір атрибутів (даних) та методів (функцій чи процедур), які матимуть об'єкти, виготовлені на основі цього класу. Атрибути описують риси або характеристики об'єкта, тоді як методи визначають дії або поведінку, яку об'єкт може здійснювати.

На рисунку 2.17 зображена діаграма класів інформаційної системи інтернет-магазину комп'ютерних ігор.

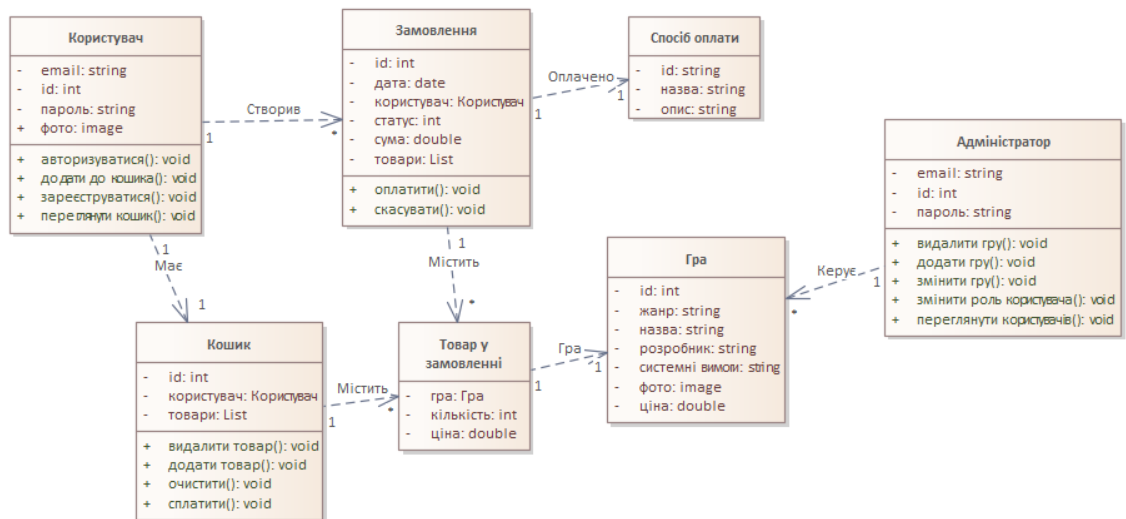


Рисунок 2.17 - Діаграма класів

Джерело: сформовано автором на основі виконаного дослідження

Основні елементи діаграми визначення блоків:

- 1) Блоки. Вони представляють основні компоненти системи. Блоки можуть бути фізичними об'єктами, програмними модулями або процесами.
- 2) Частини. Це підкомпоненти блоків, які становлять його внутрішню структуру. Частини можуть бути іншими блоками або простими об'єктами.
- 3) Порти. Вони представляють точки взаємодії між блоками. Порти показують, через які частини блоки пов'язані між собою.

4) Спільність. Вказує, що деякі блоки мають спільні частини або порти. На діаграмі це показується стрілками між блоками.

На рисунку 2.18 показана діаграма визначення блоків та структура системи інтернет-магазину комп'ютерних ігор.

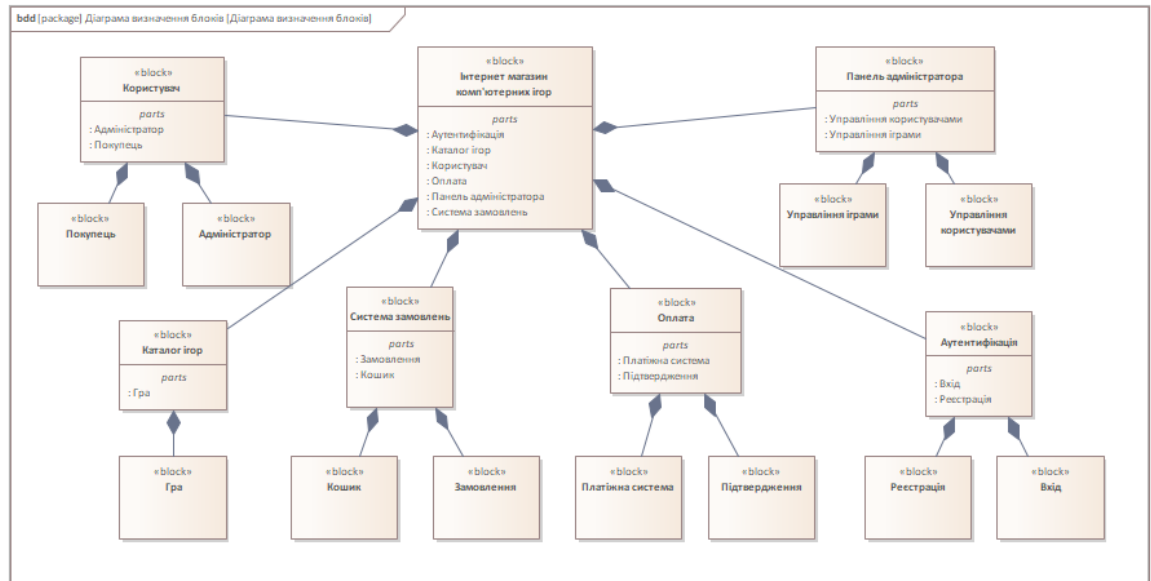


Рисунок 2.18 - Діаграма визначення блоків

Джерело: сформовано автором на основі виконаного дослідження

1) Головний блок «Інтернет-магазин комп'ютерних ігор». Це центральний блок, який представляє повну систему інтернет-магазину. Він складається з частин: «Аутентифікація», «Каталог ігор», «Користувач», «Оплата», «Панель адміністратора», «Система замовлень». Цей блок слугує основою, що об'єднує всі функціональні модулі системи.

2) Блок «Користувач». Цей блок описує суб'єктів, які взаємодіють із системою. Частини: «Адміністратор» і «Покупець». Вони мають різні ролі: покупець здійснює перегляд і покупку ігор, а адміністратор керує наповненням та користувачами.

3) Блок «Каталог ігор». Цей блок відповідає за відображення та зберігання даних про ігри. Містить частину: «Гра», яка включає характеристики товару, такі як назва, жанр, платформа, ціна тощо.

4) Блок «Система замовлень». Описує процеси, пов'язані з оформленням покупок. Містить частини: «Замовлення» та «Кошик». Цей блок

дозволяє додавати товари, переглядати вміст кошика та оформлювати замовлення.

5) Блок «Оплата». Відповідає за обробку транзакцій у системі. Містить частини: «Платіжна система» та «Підтвердження». Реалізує обробку платежів, взаємодію з платіжними шлюзами та підтвердження успішних операцій.

6) Блок «Панель адміністратора». Забезпечує інтерфейс для керування ресурсами системи. Включає частини: «Управління користувачами» та «Управління іграми». Цей блок доступний лише адміністраторам.

7) Блок «Аутентифікація». Відповідає за вхід користувачів до системи. Містить частини: «Вхід» та «Реєстрація». Цей блок забезпечує безпечний доступ до персонального кабінету користувача або адміністратора.

8) Спільність між блоками. Блоки взаємодіють один з одним через вбудовані частини. Наприклад, блок «Користувач» пов'язаний з блоками «Аутентифікація» (через «Вхід» і «Реєстрацію»), «Система замовлень» (через «Кошик» і «Замовлення»), а також з «Оплатою». Блок «Панель адміністратора» включає частини, що перетинаються з іншими, наприклад, «Управління іграми» пов'язане з «Каталогом ігор».

9) Загальна структура. Діаграма демонструє модульну архітектуру інтернет-магазину, де головний блок агрегує декілька функціональних систем. Кожен блок виконує певну роль у загальному функціонуванні системи — від реєстрації до управління товарами, замовленнями та оплатою. Взаємозв'язки між частинами забезпечують цілісну роботу інтернет-магазину.

Діаграма внутрішніх блоків є частиною мови моделювання ОРМ (Object-Process Methodology). Ця діаграма використовується для детального опису внутрішньої структури конкретного блоку та процесів, які в ньому відбуваються. Вона показує взаємодію частин блоку через порти і процеси, які реалізують функціональність блоку.

Основні елементи діаграми:

- 1) Блок. Представляє об'єкт або систему, яку моделюють.
- 2) Частини. Це підкомпоненти блоку, які входять до його складу.

3) **Порти.** Представляють точки взаємодії між частинами блоку. Порти показують, через які частини блоку відбуваються процеси.

4) **Процеси.** Це дії або операції, які відбуваються між частинами блоку через порти.

5) **Залежності.** Вказують, які частини залежать одна від одної через порти.

На рисунку 2.19 показана діаграма внутрішніх блоків та структура системи інтернет-магазину комп'ютерних ігор.

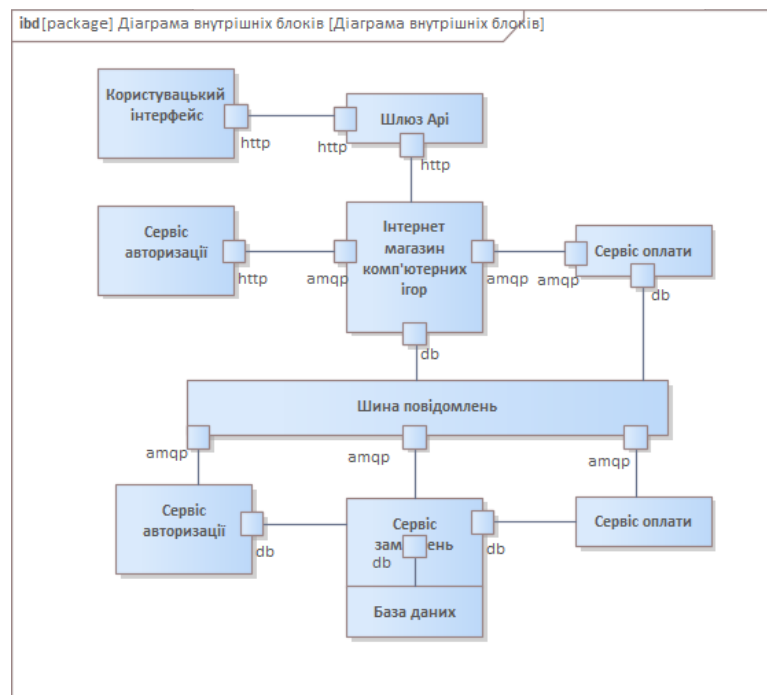


Рисунок 2.19 - Діаграма внутрішніх блоків

Джерело: сформовано автором на основі виконаного дослідження

1) **Головний блок:** «Інтернет-магазин комп'ютерних ігор». Це головний блок, який представляє централізовану систему інтернет-магазину. Він пов'язує в єдину систему всі сервіси, включаючи користувацький інтерфейс, API-шлюз, сервіси авторизації, оплати, обробки замовлень та шину повідомлень. У ньому реалізовано обробку запитів, маршрутизацію, передачу повідомлень та роботу з базами даних.

2) **Частини блоку.**

- **Користувацький інтерфейс** — відповідає за взаємодію користувача з системою. Порти: http до Шлюзу API.

- Шлюз API приймає HTTP-запити та маршрутизує їх до відповідних сервісів. Порти: http до користувацького інтерфейсу та до центральної системи.
- Сервіс авторизації відповідає за автентифікацію та перевірку доступу користувачів. Порти: amqp (використовується для обміну повідомленнями), db (для з'єднання з базою даних).
- Сервіс оплати обробляє платіжні транзакції та підтвердження. Порти: amqp, db.
- Сервіс замовлень управляє створенням і збереженням замовлень. Порти: db.
- Шина повідомлень забезпечує обмін повідомленнями між сервісами за допомогою протоколу amqp. Порти: з'єднання з авторизацією, оплатою та інтернет-магазином.
- База даних — забезпечує зберігання даних усіх систем (авторизація, замовлення, оплата). Порти: db до відповідних сервісів.

3) Процеси між частинами. Процес авторизації: починається з Користувацького інтерфейсу, проходить через Шлюз API, передається до Сервісу авторизації, а потім дані обробляються та записуються у базу через порт db. Процес замовлення: дані з інтернет-магазину або інтерфейсу передаються в Сервіс замовлень через Шину повідомлень (amqp), після чого зберігаються у базі (db). Процес оплати: транзакції передаються з Інтернет-магазину комп'ютерних ігор до Сервісу оплати через amqp, який, у свою чергу, взаємодіє з базою (db). Процес маршрутизації: запити HTTP проходять через Шлюз API, далі направляються в головний блок або відповідний сервіс залежно від типу запиту.

4) Залежності між частинами. Користувацький інтерфейс до Шлюзу API (через порт http). Шлюз API до Інтернет-магазину комп'ютерних ігор (через http). Інтернет-магазин від/до Сервісу авторизації, Сервісу оплати (через amqp). Усі сервіси (авторизація, оплата, замовлення) до Баз даних (через db). Всі взаємодії сервісів координуються через Шину повідомлень.

5) Діаграма відображає модульну мікросервісну архітектуру системи. Головний блок складається з кількох незалежних сервісів, що взаємодіють між собою через шину повідомлень (amqp) або безпосередньо через HTTP та бази даних. Кожен сервіс має окремі порти для зв'язку, що дозволяє масштабувати, розширювати та відмовостійко реалізовувати систему.

Діаграма трасування — це вид діаграми, який застосовується в системній інженерії, аналізі вимог і управлінні проєктами, щоб відслідковувати зв'язки між елементами на різних рівнях системи.

Вона демонструє, як вимоги користувача пов'язані із системними специфікаціями, архітектурою, тестуванням, елементами реалізації тощо, дозволяє відстежити генезис кожного елемента системи — від початкових цілей до впровадження та валідації, підтримує зворотнє трасування - від виконання до вимоги і є інструментом перевірки повноти, особливо у великих системах.

Діаграма трасування є графічним засобом для демонстрації логічних взаємозв'язків між окремими елементами системи.

На рисунку 2.20 зображена діаграма трасування Інтернет-магазину комп'ютерних ігор.

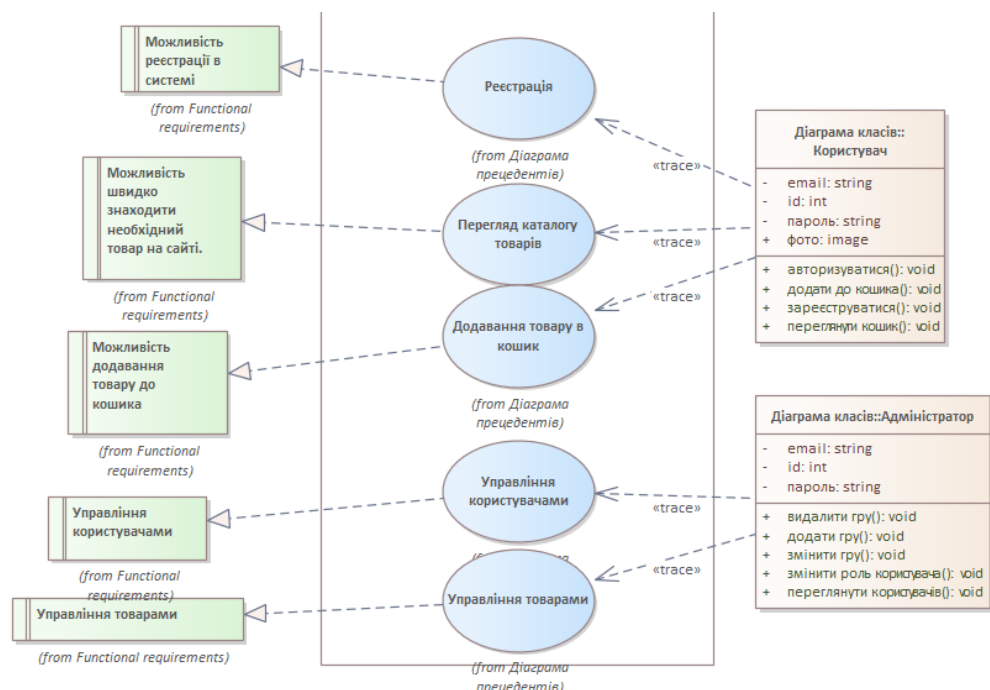


Рисунок 2.20 - Діаграма трасування

Джерело: сформовано автором на основі виконаного дослідження

Матриця взаємозв’язків є табличним відображенням діаграми трасування, що показує, як вимоги (рядки) пов’язані з іншими об’єктами (наприклад, компонентами, тестами, системами тощо, які розташовані в стовпцях).

Вона надає прозорість відповідності між вимогами та їх реалізацією, сприяє виявленню пропущених вимог (немає реалізованих частин), нереалізованих аспектів (без вимог) і відсутніх тестів для конкретних функцій. Вона також часто застосовується при верифікації та валідації систем.

Матриця трасування (матриця взаємозв’язків) — це її табличний формат, що дозволяє оперативно оцінити повноту та точність виконання вимог.

На рисунках 2.21 та 2.22 зображені матриці взаємозв’язків, що доповнюють діаграму трасування.

Target +	Functional requirements::Адмін	Functional requirements::Гучні	Functional requirements::Довги	Functional requirements::Забезп	Functional requirements::Зручн	Functional requirements::Кваліт	Functional requirements::Клієнт	Functional requirements::Можл	Functional requirements::можл	Functional requirements::можл	Functional requirements::Можл	Functional requirements::Можл	Functional requirements::Можл	Functional requirements::Недос	Functional requirements::Недос	Functional requirements::Некор	Functional requirements::Обміт	Functional requirements::Пробл	Functional requirements::Пробл	Functional requirements::Управ	Functional requirements::Управ	Functional requirements::Управ	Functional requirements::Управ	Functional requirements::Швид
+ Source																								
Діаграма прецедентів::Зм...																								
Діаграма прецедентів::Інт...																								
Діаграма прецедентів::Ко...																								
Діаграма прецедентів::Оп...																								
Діаграма прецедентів::О...																								
Діаграма прецедентів::Пе...																								
Діаграма прецедентів::Пе...																								
Діаграма прецедентів::Ре...																								
Діаграма прецедентів::Уп...																								
Діаграма прецедентів::Уп...																								
Діаграма прецедентів::Уп...																								

Рисунок 2.21 - Матриця взаємозв'язків

Джерело: сформовано автором на основі виконаного дослідження

Target +	Діаграма класів::Адміністратор	Діаграма класів::Гра	Діаграма класів::Діаграма клас	Діаграма класів::Замовлення	Діаграма класів::Користувач	Діаграма класів::Кошик	Діаграма класів::Спосіб оплати	Діаграма класів::Товар у замове
+ Source								
Діаграма прецедентів::Зм...								
Діаграма прецедентів::Інт...								
Діаграма прецедентів::Ко...								
Діаграма прецедентів::Оп...								
Діаграма прецедентів::О...								
Діаграма прецедентів::Пе...					↑			
Діаграма прецедентів::Пе...								
Діаграма прецедентів::Ре...					↑			
Діаграма прецедентів::Уп...								
Діаграма прецедентів::Уп...	↑							
Діаграма прецедентів::Уп...	↑							

Рисунок 2.22 - Матриця взаємозв'язків

Джерело: сформовано автором на основі виконаного дослідження

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1. Інформаційне забезпечення

Інформаційне забезпечення (ІЗ) інтернет-магазину комп'ютерних ігор охоплює набір даних, структур, форматів і методів обробки, потрібних для забезпечення функціонування ІС. Головними елементами ІЗ є: вхідні дані, вихідні дані, масиви інформації, канали передачі даних і база даних, яка структурована за допомогою сучасних СКБД.

Структура ІЗ включає масиви користувачів (UserData), каталог ігор (Games_catalog), замовлення (Orders), платежі (Payments) тощо. Дані представлені у вигляді структурованих записів та зберігаються в MongoDB — сучасній документоорієнтованій NoSQL СКБД, яка була обрана за її гнучкість, масштабованість і високу продуктивність для роботи з JSON-подібними структурами.

MongoDB - це потужна, адаптивна та масштабована NoSQL-система бази даних, що гарантує високу продуктивність і обробку даних в реальному часі. На відміну від звичайних реляційних баз даних, MongoDB застосовує документно-орієнтовану модель, що дає можливість розробникам ефективно зберігати і управляти великими обсягами нестандартизованих або напівструктурованих даних.

MongoDB - це система управління базами даних, а інформація зберігається в цих системах. Іншими словами, середовище MongoDB пропонує нам сервер, який ми можемо запустити, а потім на ньому створити кілька баз даних за допомогою MongoDB. Оскільки це база даних NoSQL, інформація зберігається в документах і колекціях.

MongoDB пропонує простоту інтеграції з фронтом на React і має вбудовану підтримку агрегаційних запитів, індексації та горизонтального масштабування.

Носіями інформації є: сервери баз даних, електронні носії (SSD), а також мережеві сховища для резервного копіювання. Передача даних відбувається через HTTP/HTTPS-протоколи з використанням REST API.

На рисунку 3.1 зображена загальна схема інформаційного забезпечення при роботі з інтернет магазином комп'ютерних ігор.

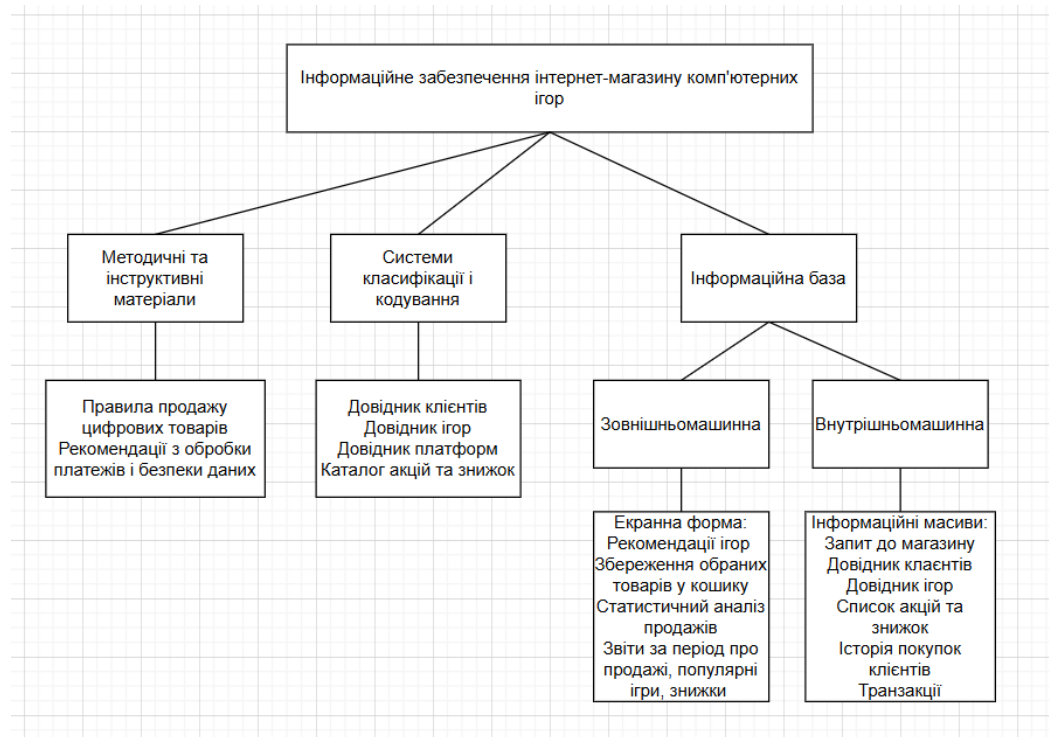


Рисунок 3.1- Схема інформаційного забезпечення

Джерело: сформовано автором на основі виконаного дослідження

Основними джерелами інформації є внутрішні у вигляді бази даних користувачів, замовлень, ігор та зовнішні у вигляді інтернету, API від партнерів та постачальників ігор.

Первинна інформація збирається за допомогою форм реєстрації, авторизації, оформлення замовлень, додавання нових ігор адміністратором. Інформація подається у форматі JSON/XML та надходить у режимі реального часу (наприклад, при авторизації чи оплаті).

Носіями інформації є: веб-інтерфейс, API запити через інтернет-браузер, системна база даних MongoDB.

Частота отримання: інформація про користувачів та замовлення відбувається в режимі реального часу, оновлення ігрового каталогу відбувається раз на день або раз на тиждень, інформація про акції/знижки

надається кожної п'ятниці, інформація про платежі надається відразу після операції.

За кожним вхідним повідомленням зазначено його джерело, а сама: UserData створюється під час реєстрації, джерелом є користувач, Games_catalog та New_games подаються адміністратором через інтерфейс, замовлення та платежі система створює під час обробки замовлення.

Класифікація у системі здійснена для: ідентифікації користувачів, категоризації ігор, маркування замовлень та угод.

Застосовуються такі класифікатори: жанр ігор (екшн, рольові, симулятори, стратегії, пригодницькі тощо), платформи (WINDOWS, MACOS, LINUX, PLAYSTATION, XBOX), замовлення (унікальний UUID), платежі (PAY_VISA, PAY_MC, PAY_PAYPAL).

Методи кодування — це символно-цифрові коди фіксованої чи змінної довжини. Наприклад: код гри (GAME_000123), код жанру (аббревіатура з 3 літер), код замовлення (UUID (36 знаків)).

Документи розподіляються на: екранні форми (поради, підтвердження заявки), файли (щоденні звіти, статистичні дані, звіт зі знижками).

Способи подання документів: JSON/XML (для інтерфейсів і баз даних), HTML/JSX (для інтерфейсів на React), CSV/Excel (для аналітичних звітів).

Вимоги до документів: унікальний ідентифікаційний номер, актуальність та точність, захист особистих даних.

Приклади документів: Recomend_games, Order_report, Discount_report.

На електронну пошту надсилаються: щотижневий продажний звіт, щоденний звіт щодо замовлень, звіт щодо акцій.

В таблиці 3.1 представлено опис структури масиву «Користувачі».

Таблиця 3.1 – Опис структури масиву «Користувачі»

Найменування	Ідентифікатор у програмі	Умове позначення у файлах	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	

Продовження таблиці 3.1 - Опис структури масиву «Користувачі»

Ідентифікатор	_id	ID	UUID	PK	Автоматичний	Так	Так	-
Електронна пошта	email	E	String	FK	Email-формат	Так	Так	Авторизація
Пароль	Password	P	String	FK	Мінімум 8 символів	Так	Ні	Авторизація
Ім'я	Username	U	String	FK	Не порожнє	Так	Ні	Профіль
Роль	Role	R	String	FK	Користувач/Адміністратор	Так	Ні	Доступ/Адміністрування
Дата реєстрації	createdAt	D	Date	FK		Так	Ні	Аналітика

В таблиці 3.2 представлено опис структури масиву «Ігри».

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.2 - Опис структури масиву «Ігри»

Найменування	Ідентифікатор у програмі	Умове	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/	Умова на значення	Обов'язкове	Індексне	
Ідентифікатор	_id	ID	UUID	PK	Автоматичний	Так	Так	-
Назва	Title	T	String	FK	Унікальна	Так	Так	Каталог
Жанр	Genre	G	String	FK	ACT,RPG,SIM,etc..	Так	Ні	Фільтрація
Ціна	Price	C	Decimal	FK	>0	Так	Ні	Розрахунок
Платформа	platform	P	String	FK	WIN,MAC,PS,XBOX	Так	Ні	Фільтр

Джерело: сформовано автором на основі виконаного дослідження

В таблиці 3.3 представлено опис структури масиву «Замовлення».

Таблиця 3.3 - Опис структури масиву «Замовлення»

Найменування	Ідентифікатор у програмі	Умове	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/	Умова на значення	Обов'язкове	Індексне	
Ідентифікатор	_id	ID	UUID	PK	Автоматичний	Так	Так	-
Користувач	User_id	U	UUID	FK	Автоматичний	Так	Так	Посилання на users
Перелік ігор	Items	I	List	FK	Не порожній	Так	Ні	Продукти
Дата замовлення	createdAt	D	DateTime	FK	Автоматичний	Так	Так	Звітність
Сума	total	T	Decimal	FK	≥ 0	Так	Ні	Оплата

Джерело: сформовано автором на основі виконаного дослідження

В таблиці 3.4 представлено опис структури масиву «Платежі».

Таблиця 3.4 - Опис структури масиву «Платежі».

Найменування	Ідентифікатор у програмі	Умове	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/	Умова на значення	Обов'язкове	Індексне	
Ідентифікатор	_id	ID	UUID	PK	Автоматичний	Так	Так	-
Замовлення	Order_id	O	UUID	FK	Автоматичний	Так	Так	Посилання на orders
Сума	amount	A	Decimal	FK	> 0	Так	Ні	Розрахунок
Дата оплати	Date	D	DateTime	FK	Автоматичний	Так	Так	Звітність
Метод оплати	method	M	String	FK	Visa, MasterCard	Так	Ні	Аналітика

Джерело: сформовано автором на основі виконаного дослідження

Для реалізації системи інтернет-магазину комп'ютерних ігор в межах цього проєкту вибрано MongoDB як систему управління базами даних (СУБД). Цей вибір обумовлений як архітектурними характеристиками інформаційної

системи, так і вимогами до гнучкості зберігання, масштабованості та обробки даних в реальному часі.

MongoDB – це документо-орієнтована NoSQL СКБД, що дозволяє зберігати об'єкти у форматі JSON. Це корисно при взаємодії з різноманітними структурами даних, такими як продукти, замовлення, користувачі тощо.

MongoDB забезпечує горизонтальне масштабування через шардінг, що дає можливість обробляти великі обсяги даних і запитів — це особливо важливо для систем з інтенсивним трафіком.

Завдяки індексації та агрегованим запитам MongoDB гарантує високу ефективність під час пошуку, фільтрації, сортування й аналізу даних.

СКБД надає зручні бібліотеки для Node.js, які полегшують розробку бекенду та спілкування з фронтендом React в режимі реального часу.

Відсутність обов'язкових схем дає можливість змінювати структуру документів без ускладнених міграцій, що є важливим на стадіях розробки ітераційного MVP-продукту.

Вимоги та обмеження, що впливають на проєктування: відсутність чіткої схеми; потрібно уважно розробити структуру документів, необхідність ручної реалізації зовнішніх зв'язків (виконуються на рівні коду), треба проєктувати індекси та агрегаційні запити для уникнення перевантаження, обов'язкове впровадження механізмів авторизації, обмежень доступу до колекцій, слід реалізувати регулярне резервування (бекапи) при високій частоті змін, СКБД вибрана з урахуванням майбутнього горизонтального масштабування.

На рисунку 3.2 показана інфологічна модель бази даних інтернет-магазину комп'ютерних ігор.

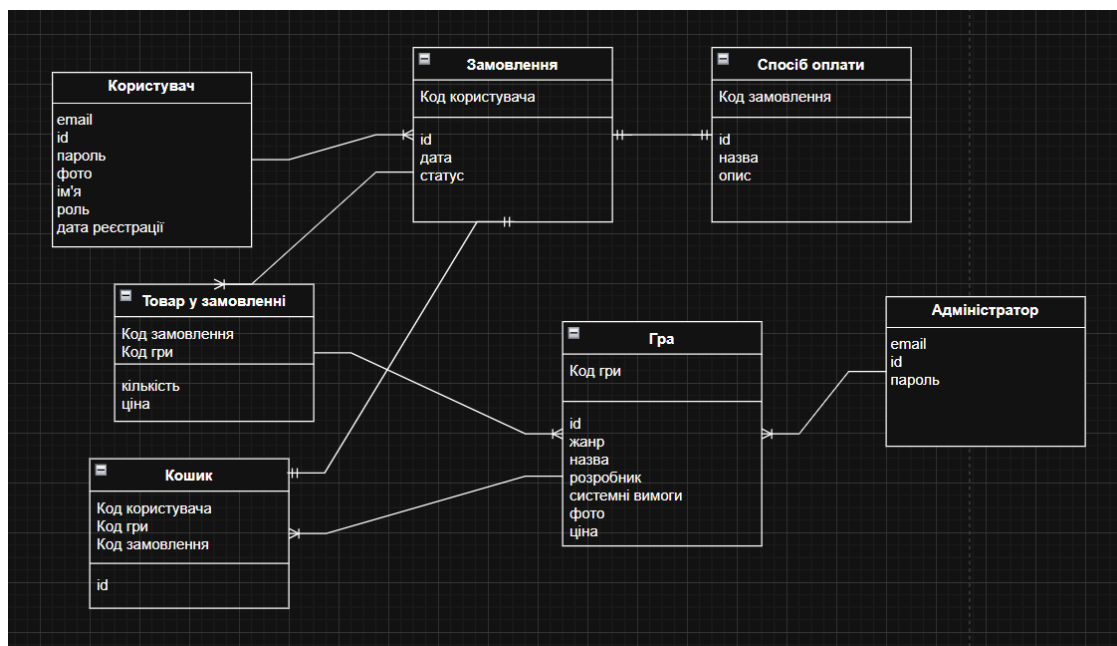


Рисунок 3.2 - Інфологічна модель бази даних

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 3.3 показана даталогічна модель бази даних інтернет-магазину комп'ютерних ігор.

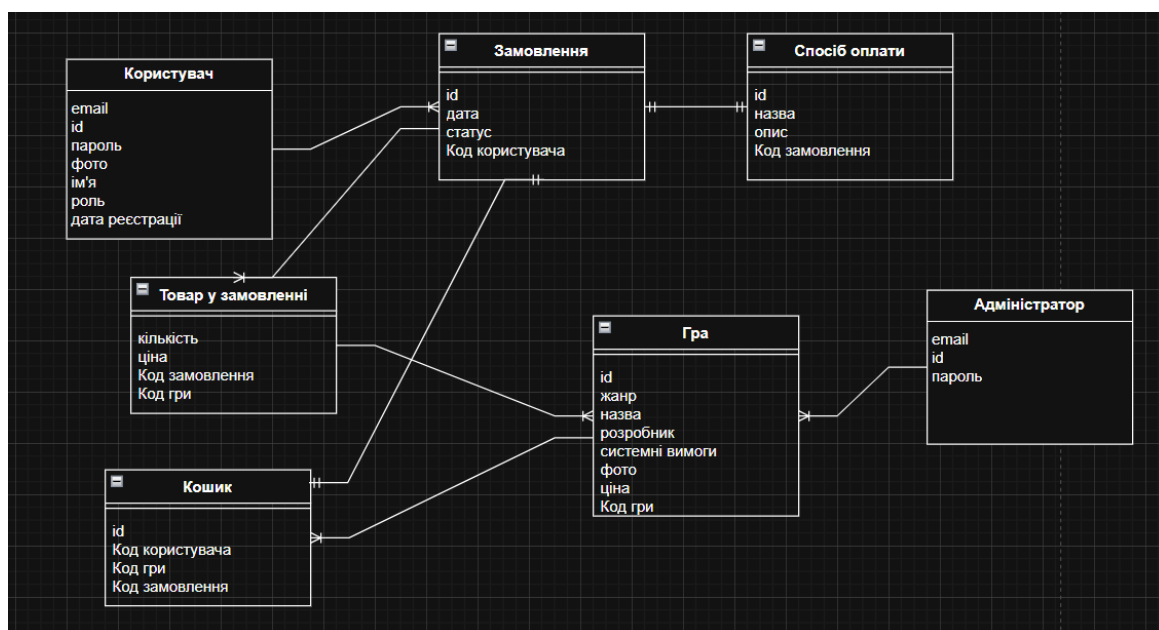


Рисунок 3.3 - Даталогічна модель бази даних

Джерело: сформовано автором на основі виконаного дослідження

3.2. Технічне забезпечення

Технічне забезпечення опишемо у вигляді діаграми розгортання, яка відображає фізичну архітектуру системи (див. рисунок 3.4). Користувачі це насамперед клієнти, які взаємодіють із сайтом. Їм потрібен надійний, швидкий і зручний інтерфейс для здійснення покупок. Використання декількох користувачів на схемі вказує на можливість обробки численних одночасних запитів. До веб-серверу відносимо кошик, який дозволяє зберігати вибрані товари користувачем до моменту оформлення замовлення. Розміщення кошика на веб-сервері забезпечує швидкий доступ і зручність у використанні, що сприяє покращенню досвіду користувача. Також сюди відноситься платіжний шлюз, який забезпечує безпечне з'єднання з платіжним порталом, що необхідно для обробки онлайн-оплат. Наявність платіжного шлюзу на веб-сервері підвищує безпеку й мінімізує ризик зловживань. Платіжний портал це окремий сервіс, що дозволяє користувачам здійснювати оплату за допомогою кредитних карток або інших платіжних засобів. Використання зовнішнього платіжного порталу забезпечує високу безпеку та відповідність стандартам обробки платежів. До серверу бази даних відносяться:

- Виробнича база даних — містить важливі дані для забезпечення внутрішніх бізнес-процесів, таких як управління запасами, звітність і аналіз продажів. Її ізоляція від бази даних інтернет-магазину забезпечує додатковий рівень безпеки та захисту даних.
- База даних інтернет-магазину — зберігає інформацію про товари, клієнтів і транзакції, необхідну для безперебійної роботи сайту. Вона забезпечує швидкий доступ до даних під час запитів користувачів, що підвищує ефективність і швидкодію веб-сайту.

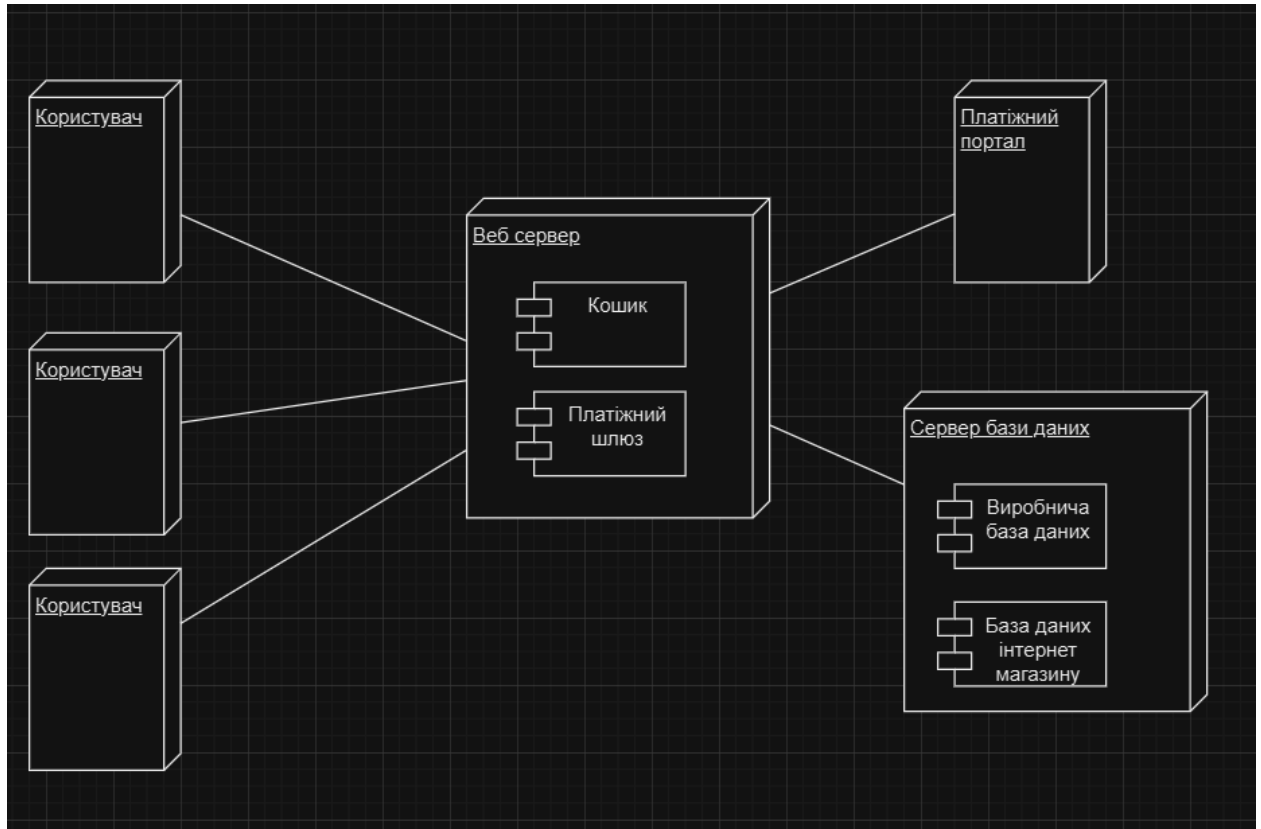


Рисунок 3.4 - Діаграма розгортання

Джерело: сформовано автором на основі виконаного дослідження

Інформаційна система онлайн-магазину комп'ютерних ігор працюватиме в умовах локальної мережі компанії з можливістю віддаленого доступу через Інтернет. Система забезпечує централізоване зберігання та обробку даних на сервері з підтримкою веб-середовища, до якого доступуються адміністратори та користувачі через браузер. Впровадження системи може відбуватися на основі хмарного сервісу або на власному серверному устаткуванні.

Комплекс технічних засобів містить такі ключові елементи: веб-сервер для розміщення бекенд-складової системи, система бази даних для зберігання інформації про користувачів, ігри, замовлення та платежі, робочі місця користувачів для входу на сайт, реєстрації, перегляду продукції, здійснення покупок, адміністративне АРМ для керування контентом, аналізу, обробки замовлень.

АРМ адміністратора встановлюється на комп'ютері з операційною системою Windows або Linux. За допомогою веб-інтерфейсу адміністратор може: додавати/змінювати ігри, створювати статистичні звіти, керувати користувачами та їхніми ролями. Програмне забезпечення: браузер, доступ до

інтерфейсу адміністратора через HTTPS SSH-з'єднання з сервером у разі необхідності адміністративного втручання.

Вся комунікація відбувається за допомогою протоколу HTTPS. Взаємодія між клієнтською частиною і серверною частиною реалізується за допомогою REST-запитів. База даних MongoDB з'єднана з сервером, на якому є автентифікація та обмеження доступу.

Для гарантування безперервної роботи застосовується балансувальник навантаження разом із автоматичним масштабуванням у хмарному середовищі.

На рисунку 3.5 зображена загальна схема автоматизації.

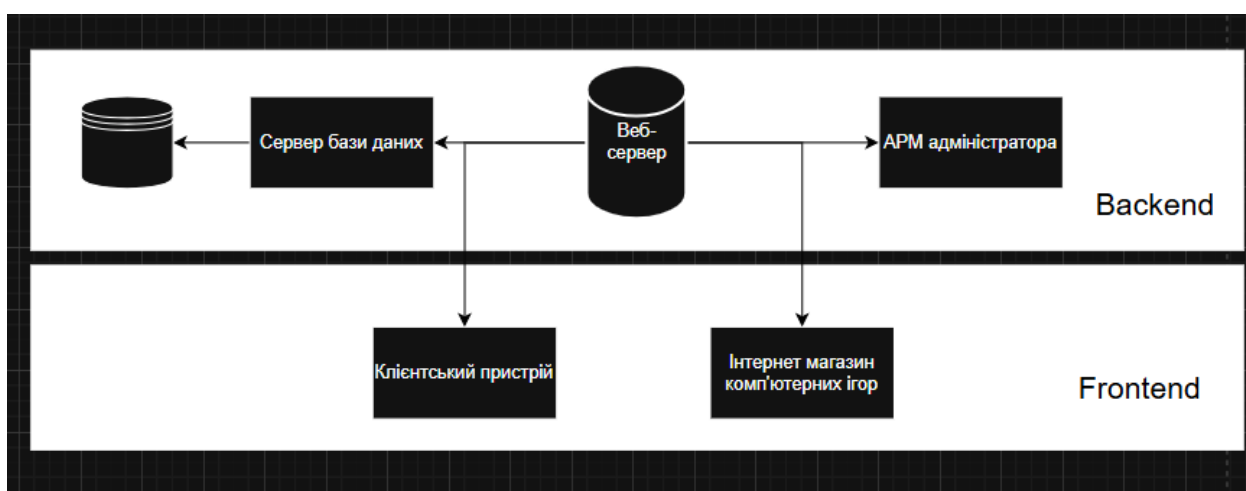


Рисунок 3.5 - Загальна схема автоматизації

Джерело: сформовано автором на основі виконаного дослідження

В таблиці 3.5 показана характеристика комплексу технічних засобів.

Таблиця 3.5 – Характеристика комплексу технічних засобів

Компонент	Характеристика
Сервер бази даних	2 vCPU, 4 GB RAM, 50 GB SSD, MongoDB
Веб-сервер	2 vCPU, 2 GB RAM, Node.js
APM адміністратора	Пк з браузером, 8 GB RAM, Windows/Linux
Клієнтський пристрій	Мобільний/Пк браузер, мін. 4 GB RAM

Продовження таблиці 3.5 - Характеристика комплексу технічних засобів

Інтернет з'єднання	Не менше 20 Мбіт/с
--------------------	--------------------

Джерело: сформовано автором на основі виконаного дослідження

3.3. Програмне забезпечення

Програмне забезпечення опишемо у вигляді діаграми компонентів, яка показує як компоненти програмної системи організовані і як вони взаємодіють між собою (див. рисунок 3.6).

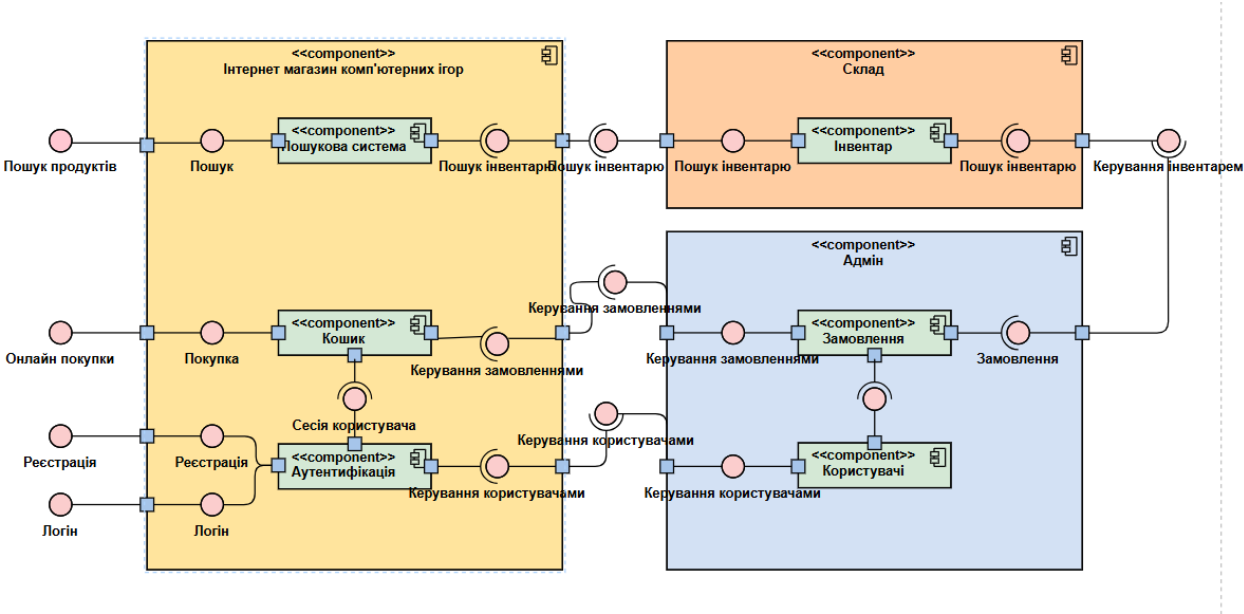


Рисунок 3.6 - Діаграма компонентів інформаційної системи інтернет-магазину комп'ютерних ігор

Джерело: сформовано автором на основі виконаного дослідження

Test Case — це структурований опис випробування, який використовується для перевірки функціональності чи характеристик системи. Він містить докладну інформацію про те, як має бути випробувана конкретна частина програмного забезпечення, включаючи послідовність дій, передбачувані результати та фактичні спостереження після виконання тесту. Test Case можна пов'язати з елементами моделі, як класи, компоненти або сценарії користувача, що дозволяє відстежувати, які частини системи були протестовані.

Стислий опис тест кейсів інформаційної системи інтернет-магазину комп'ютерних ігор:

1. Реєстрація нового користувача. Перевіряється процес реєстрації нового користувача в системі. Користувач вводить свої дані (ім'я, email, пароль) і натискає кнопку «Зареєструватися». Очікуваний результат: користувач успішно реєструється, і система перенаправляє його на головну сторінку.

2. Авторизація користувача. Перевіряється можливість авторизації зареєстрованого користувача. Користувач вводить email та пароль і натискає кнопку «Увійти». Очікуваний результат: користувач успішно увійшов у систему, і відбувається перенаправлення на головну сторінку.

3. Додавання товару до кошика. Перевіряється функціональність додавання товару до кошика. Авторизований користувач обирає товар і натискає кнопку «Додати до кошика». Очікуваний результат: товар додається до кошика, і кількість товарів у кошику оновлюється.

4. Видалення товару з кошика. Перевіряється можливість видалення товару з кошика. Користувач відкриває кошик і натискає кнопку «Видалити» біля товару. Очікуваний результат: товар видаляється з кошика, а загальна сума замовлення оновлюється.

5. Пошук товару. Перевіряється робота пошукової системи. Користувач вводить ключове слово у поле пошуку і натискає кнопку «Пошук». Очікуваний результат: система відображає список товарів, що відповідають пошуковому запиту.

6. Додавання нового товару. Перевіряється можливість адміністратора додавати новий товар до каталогу. Адміністратор переходить до панелі управління, вводить дані про товар і натискає кнопку «Додати». Очікуваний результат: товар додається до каталогу і стає доступним для перегляду.

7. Оновлення ролі користувача. Перевіряється можливість адміністратора змінювати роль користувача. Адміністратор переходить до списку користувачів, обирає користувача і змінює його роль через випадаючий список. Очікуваний результат: роль користувача успішно оновлюється.

Вся тестова документація наведена в Додатку А.

Для розробки прототипу був використаний застосунок Figma.

Місія Figma - забезпечити доступність дизайну для всіх. Її товари сприяють людям візуально передавати свої ідеї та розробляти предмети спільно.

Figma Design дає змогу користувачам розробляти, обмінюватися й перевіряти дизайни для веб-сайтів, мобільних програм та різноманітних цифрових продуктів і послуг. Цей інструмент широко використовується дизайнерами, продакт-менеджерами, авторами та розробниками, сприяючи всім, хто залучений до процесу проєктування, вносити свої пропозиції, давати коментарі та швидше ухвалювати оптимальні рішення.

Режим розробника активує ряд функцій у Figma Design, які сприяють перетворенню проєктів на код. З режимом Dev Mode дизайнери і програмісти можуть бути в одному руслі, гарантуючи, що важливі аспекти не втраяться під час переходу.

Figma Slides - це засіб для виготовлення слайдів, що дозволяє користувачу розробляти стильні презентації. Додавати дизайни та прототипи з Figma Design, спільно розробляти та впевнено демонструвати презентації за допомогою інструментів для співпраці і презентацій, а також використовувати інтерактивні віджети для отримання відгуків від зацікавлених сторін ще до завершення демонстрації.

FigJam надає можливість створювати онлайн-дошки, на яких кожен може долучитися. Люди часто застосовують FigJam для сесій, генерації ідей, створення схем, організації та аналізу. У FigJam є можливість використовувати текст, форми, графіки, картинки, наклейки та інші елементи для візуалізації ідей та спільного обговорення.

Основним програмним забезпеченням для створення інтернет-магазину був обраний React.

React - це JavaScript бібліотека, створена компанією Facebook, яка, зокрема, була застосована для розробки Instagram. Вона дає змогу розробникам швидко розробляти користувацькі інтерфейси для веб-сайтів, настільних і

мобільних додатків. React-додатки зазвичай функціонують швидко та продуктивно.

Як розширення мови програмування JavaScript, React пропонує безліч переваг. Продукти, розроблені за допомогою React, просто масштабуються, і, що дуже зручно, одна й та сама мова використовується на сервері, клієнті та мобільних пристроях, що забезпечує вражаючу продуктивність. Також є шаблони робочих процесів для комфортної командної співпраці, а код інтерфейсу користувача є зрозумілим та простим у підтримці.

Однією з ключових причин застосування React.js у веб-розробці є максимально оптимізований інтерфейс для розробників і мова програмування бібліотеки. Отже, простий API React підтримується швидкою продуктивністю для забезпечення безперешкодного та швидкого процесу розробки.

Застосування React.js у веб-розробці може бути надзвичайно зручним, адже React - це саме той випадок, коли ви вивчаєте одну технологію і легко здатні використовувати її на різних платформах. Це зумовлено тим, що ця бібліотека є природною, головною метою якої є виготовлення індивідуальних елементів і компонентів для веб-дизайну.

React-компонент є основним елементом React-додатку. Він дає можливість розробникам створювати повторно застосовуваний, ізольований компонент інтерфейсу користувача, яким можна легко управляти та оновлювати. Саме з цієї причини створені компоненти можна без труднощів використовувати повторно. Створивши компонент веб-додатку на React.js, ви отримуєте унікальний елемент, який можна інтегрувати в будь-який інший проект, що підтримує код на основі React.

Віртуалізуючи та зберігаючи DOM в пам'яті, React забезпечує швидкий процес рендерингу, при цьому всі зміни зовнішнього вигляду легко відображаються у віртуальному DOM. Алгоритм diff спеціалізується на порівнянні попереднього і поточного стану віртуального DOM, визначаючи найкращий спосіб реалізації нових змін без зайвих оновлень. Потім здійснюється найменша кількість оновлень для забезпечення максимально

швидкого часу читання/запису, що в результаті підвищує загальну продуктивність.

React можна застосовувати для розробки веб-, мобільних і десктопних програм, що робить його універсальним середовищем для кросплатформеного програмування. React Native, фреймворк, що базується на React, призначений для створення мобільних додатків, тоді як React Desktop дає можливість розробки десктопних додатків за допомогою веб-технологій.

Для зберігання даних та їх використання обрана база даних MongoDB.

MongoDB базується на архітектурі масштабування, яка здобула популярність серед розробників різних типів для створення масштабованих додатків з динамічними схемами даних.

Як документна база даних, MongoDB надає можливість розробникам без труднощів зберігати як структуровані, так і неструктуровані дані. Для архівування документів застосовують формат, подібний до JSON. Цей формат прямо представляється у власних об'єктах більшості сучасних мов програмування, що робить його логічним вибором для розробників, адже їм не потрібно турбуватися про нормалізацію даних. MongoDB здатен обробляти великі обсяги даних і може бути масштабованим як у вертикальному, так і горизонтальному напрямках, щоб впоратися з інтенсивними навантаженнями даних.

MongoDB була розроблена для осіб, які розробляють веб-додатки та бізнес-рішення, які повинні швидко адаптуватися та ефективно зростати. Фірми та групи розробників різних масштабів застосовують MongoDB з безлічі причин.

MongoDB є новатором NoSQL-баз даних, що виникли внаслідок обмежень SQL-СУБД у підтримці масштабування та швидких циклів розробки, необхідних для розробки сучасних програм.

NoSQL - це узагальнюючий термін; він охоплює документно-орієнтовані бази даних, як-от MongoDB, стовпчикові бази даних, бази даних у пам'яті та інші.

MongoDB завжди вкладає багато зусиль і часу, щоб розробники мали відмінний досвід використання. Розробники цінують, що MongoDB забезпечила використання бази даних з різними мовами програмування, такими як C, C#, .NET, C++, Go, Java, JavaScript, PHP, Python, Ruby, Rust, Scala та Swift.

При створенні моделей даних у MongoDB часто застосовують вбудовування об'єктів один в інший. Те, що в класичних реляційних базах даних вимагало кількох транзакцій для оновлення, в MongoDB може бути здійснено за одну транзакцію.

Для зручного зберігання та відображення медіафайлів на сайті використовується система Cloudinary.

Cloudinary – це комплексне рішення для керування зображеннями для веб-сайтів та мобільних додатків.

Програма підтримує завантаження зображень, хмарне сховище, маніпуляції із зображеннями, оптимізацію зображень для Інтернету та доставку. Cloudinary пропонує API-інтерфейси та гнучкі можливості адміністратора для інтеграції з новими та існуючими веб-додатками та мобільними додатками.

Cloudinary дозволяє завантажувати та зберігати необмежену кількість зображень конфіденційно та безпечно та включає автоматичне резервне копіювання та історичні зміни. Ви можете керувати зображеннями, використовуючи широкий спектр опцій, таких як застосування ефектів, зміна розміру, обрізка, виявлення облич, водяні знаки та багато іншого.

Зображення надсилаються через всесвітню мережу CDN Akamai для швидкої доставки, оптимізованої для будь-якого пристрою. Cloudinary виступає як рішення для керування цифровими активами з медіа-бібліотекою, яка використовує RESTful API та SDK для автоматизації керування зображеннями. Також можна отримати доступ до розширеної аналітики та звітів, щоб оптимізувати та зрозуміти продуктивність зображень.

Сервіс хостингу зображень дає можливість завантажувати фото на віддалений сервер для перегляду в мережі. Далі можна демонструвати

зображення користувачам через свій веб-сайт або веб-додаток, не витрачаючи місця для зберігання зображень на своєму веб-сервері.

Сервіси надає користувацький інтерфейс, який дозволяє завантажувати зображення з локального пристрою або з хмарних сховищ. Ключовим моментом сервісів хостингу зображень є те, що вони здатні підвищити ефективність веб-сайту, забезпечуючи контент користувачам, які знаходяться близько до їхнього реального місця розташування.

Численні сервіси хостингу зображень також надають інтерфейс програмування додатків (API) для програмного завантаження та отримання зображень. Декілька сервісів хостингу зображень також пропонують інтерфейс для завантаження файлів, який можна інтегрувати на свій веб-сайт, щоб дати можливість користувачам безпосередньо завантажувати свої файли.

Для написання коду був обраний зручний програмний застосунок Visual Studio Code.

Основою Visual Studio Code є швидкий редактор коду, який чудово підходить для щоденного використання. Підтримуючи безліч мов, VS Code сприяє швидкій продуктивності завдяки синтаксичному підсвічуванню, парному узгодженню дужок, автоматичним відступам, акцентуванню блоків, фрагментам тощо. Зрозумілі комбінації клавіш, просте налаштування та спільно створені представлення клавіатурних скорочень допомагають легко навігувати у коді.

Visual Studio Code містить інтегровану підтримку завершення коду IntelliSense, розширене семантичне сприйняття коду та навігаційні можливості, а також рефакторинг коду.

Visual Studio Code має інтерактивний дебагер, котрий дозволяє переглядати вихідний код, перевіряти змінні, досліджувати стеки викликів і виконувати команди в консолі.

VS Code також поєднується з інструментами для збору та написання скриптів, що дозволяє ефективно виконувати спільні завдання, прискорюючи щоденні робочі процеси. VS Code має інтеграцію з Git, що дозволяє виконувати

операції з контролем версій безпосередньо у редакторі, зокрема переглядати зміни, які чекають на перевірку.

3.4. Результати реалізації інформаційної системи

Результатом впровадження інформаційної системи став розроблений повнофункціональний прототип та сайт онлайн-магазину комп'ютерних ігор, який враховує всі стадії життєвого циклу обробки дій користувачів — від реєстрації до здійснення платежу та отримання аналітичних даних.

Здійснений функціонал в ході розробки: модуль автентифікації та реєстрації який забезпечує перевірку користувачів за допомогою актуальних технологій захисту особистих даних (хешування паролів, JWT, HTTPS), каталог ігор який створено з можливостями фільтрації, сортування, пошуку, показу характеристик товарів та формування докладних карток товарів, система замовлень і кошик які дають змогу додавати продукти до кошика, коригувати їхню кількість, формувати та підтверджувати замовлення, адмінпанель яка дозволяє управляти товарами, переглядати статистику продажів, створювати звіти, контролювати користувачів і встановлювати знижки.

Для технологічна реалізації були використані: React.js, Node.js + Express.js, REST API з аутентифікацією, MongoDB, використання Mongoose ODM, Cloudinary.

На сторінці реєстрації користувач може створити свій профіль та перейти на головну сторінку сайту. В разі якщо користувач уже має профіль, він може перейти на сторінку авторизації. Прототип та розроблений варіант сторінки реєстрації зображений на рисунках 3.7 та 3.8.

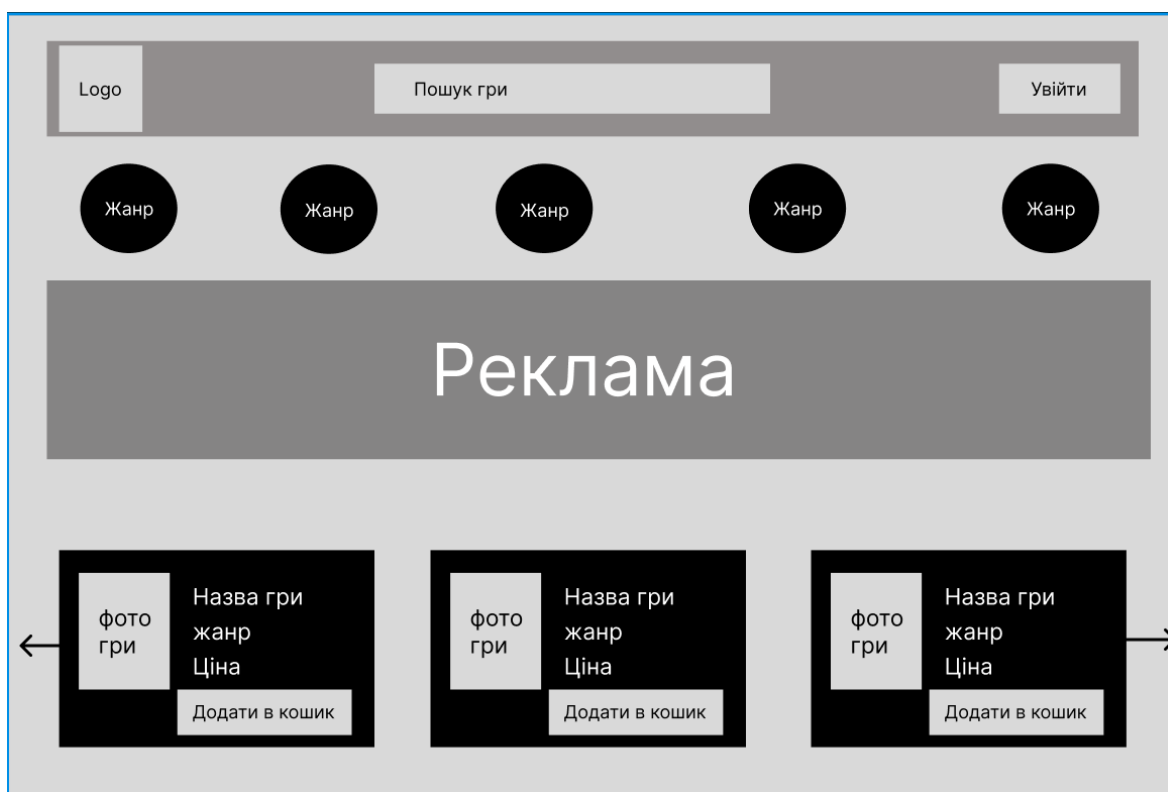


Рисунок 3.7 - Прототип сторінки реєстрації

Джерело: сформовано автором на основі виконаного дослідження

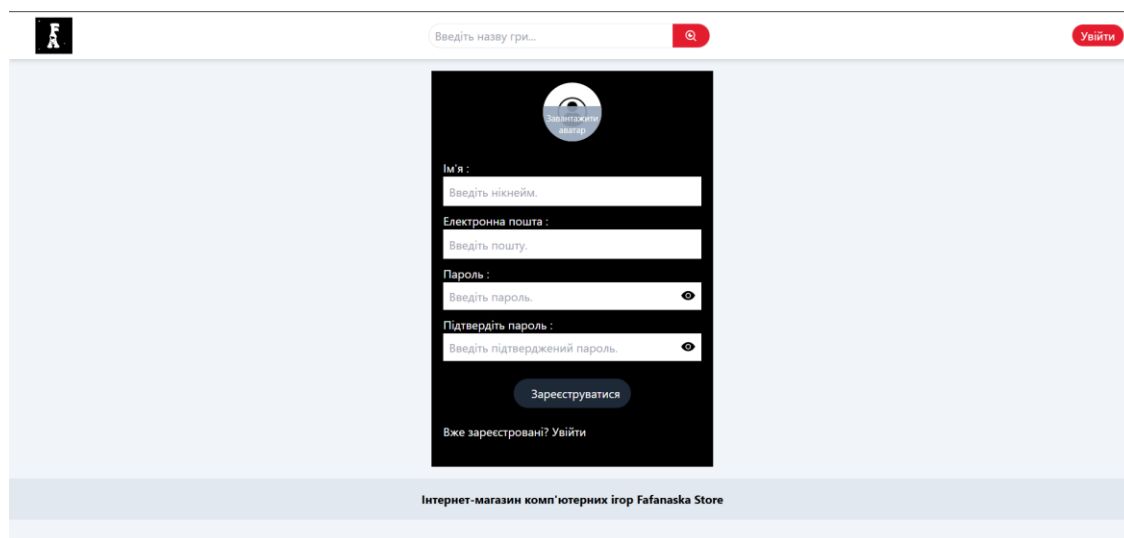


Рисунок 3.8 - Сторінка реєстрації

Джерело: сформовано автором на основі виконаного дослідження

На сторінці авторизації користувач вводить свої дані та переходить на головну сторінку сайту. В разі якщо користувач не має профіль, він може

перейти на сторінку реєстрації. Прототип та розроблений варіант сторінки авторизації зображений на рисунках 3.9 та 3.10.

Прототип сторінки авторизації. Верхня панель містить логотип (Logo), поле пошуку гри (Пошук гри) та кнопку вийти (Увійти). Центральна форма на темному фоні включає: поле для фото (Фото), поле для електронної пошти (Електронна пошта) з підписом 'Ввести пошту', поле для пароля (Пароль) з підписом 'Ввести пароль', кнопку вийти (Увійти), посилання 'Забув пароль', та посилання 'Ще не зареєстровані?' та 'Зареєструватися'.

Рисунок 3.9 - Прототип сторінки авторизації

Джерело: сформовано автором на основі виконаного дослідження

Сторінка авторизації. Верхня панель містить логотип (F), поле пошуку гри (Введіть назву гри...) та кнопку вийти (Увійти). Центральна форма на темному фоні включає: поле для фото (Фото), поле для електронної пошти (Електронна пошта) з підписом 'Введіть пошту', поле для пароля (Пароль) з підписом 'Введіть пароль' та іконку ока, кнопку вийти (Увійти), посилання 'Забув пароль', та посилання 'Ще не зареєстровані?' та 'Зареєструватися'. Нижня панель містить текст 'Інтернет-магазин комп'ютерних ігор Fafanaska Store'.

Рисунок 3.10 - Сторінка авторизації

Джерело: сформовано автором на основі виконаного дослідження

На головній сторінці сайту у користувача відкривається багато можливостей. Він може переглядати всі ігри, сортувати їх за жанром, додати їх до кошику та використати панель пошуку для швидкого доступу до товару. Також можна перейти в кошик. Якщо хочеться повернутись на сторінку відображення всіх ігор, є можливість перейти на головну сторінку сайту.

Прототип та розроблений варіант головної сторінки зображений на рисунках 3.11 та 3.12.



Рисунок 3.11 - Прототип головної сторінки

Джерело: сформовано автором на основі виконаного дослідження

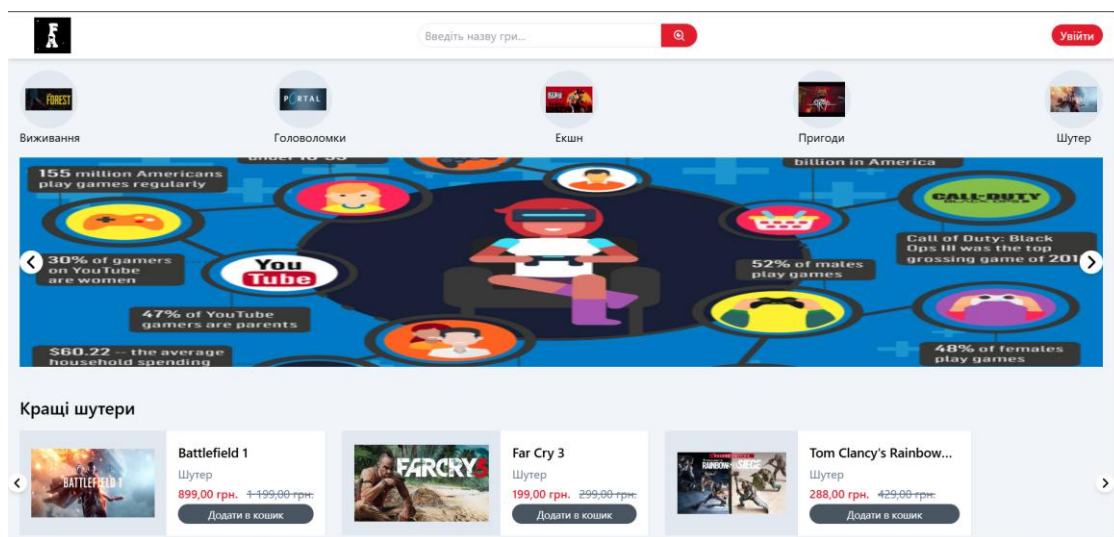


Рисунок 3.12 - Головна сторінка

Джерело: сформовано автором на основі виконаного дослідження

Якщо зверху обрати жанр гри, користувач автоматично буде перенаправлений на сторінку на якій будуть відображатись всі ігри які відносяться до обраного жанру. Прототип та розроблений варіант сторінки сортування зображений на рисунках 3.13 та 3.14.

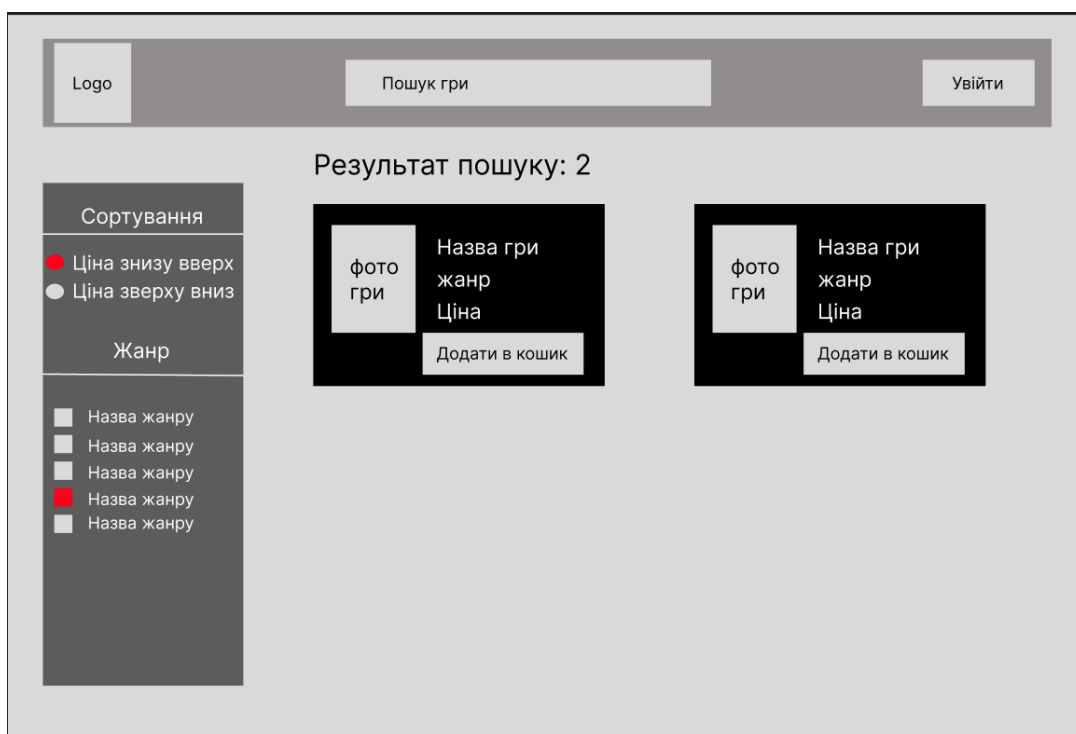


Рисунок 3.13 - Прототип сторінки сортування

Джерело: сформовано автором на основі виконаного дослідження

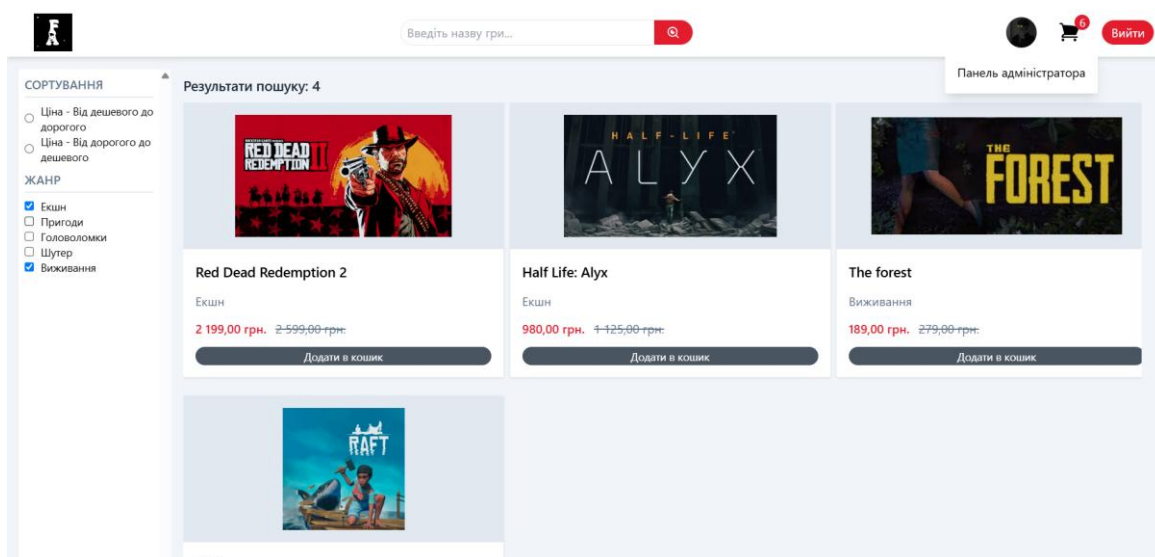


Рисунок 3.14 - Сторінка сортування

Джерело: сформовано автором на основі виконаного дослідження

Після входу в акаунт у адміністратора з'являється кнопка переходу до панелі адміністратора, а у користувача додається можливість купувати товари.

Прототип та розроблений варіант головної сторінки після входу в акаунт зображений на рисунках 3.15 та 3.16.

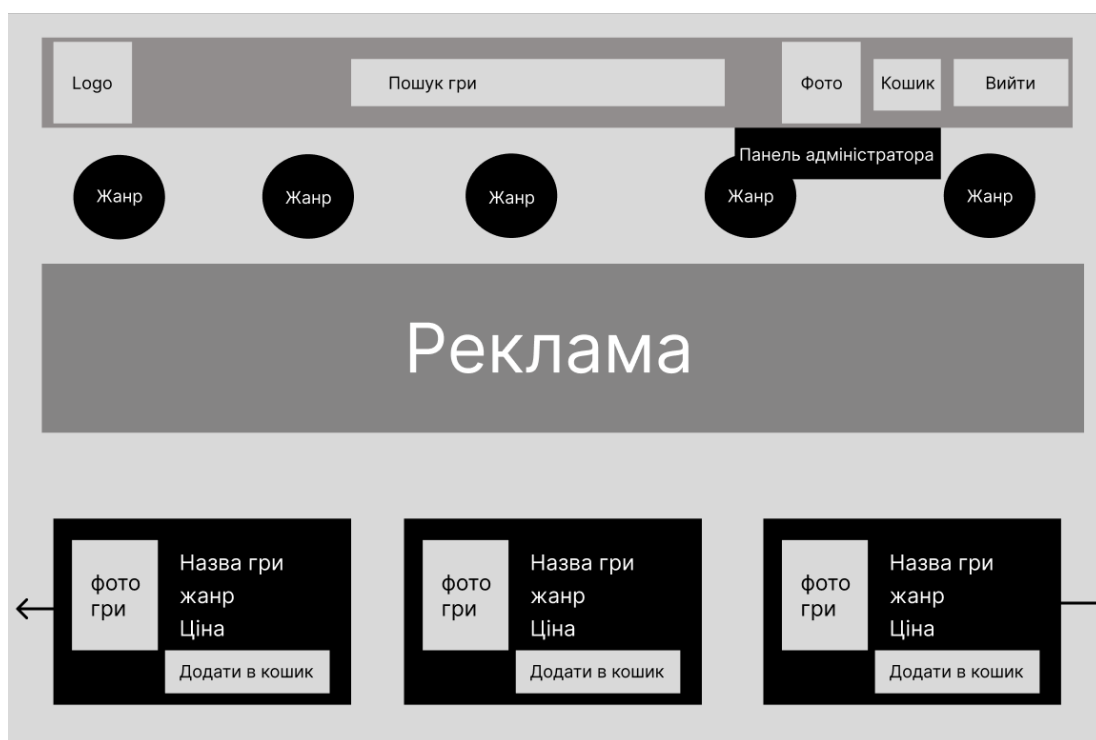


Рисунок 3.15 - Прототип головної сторінки після входу в акаунт

Джерело: сформовано автором на основі виконаного дослідження

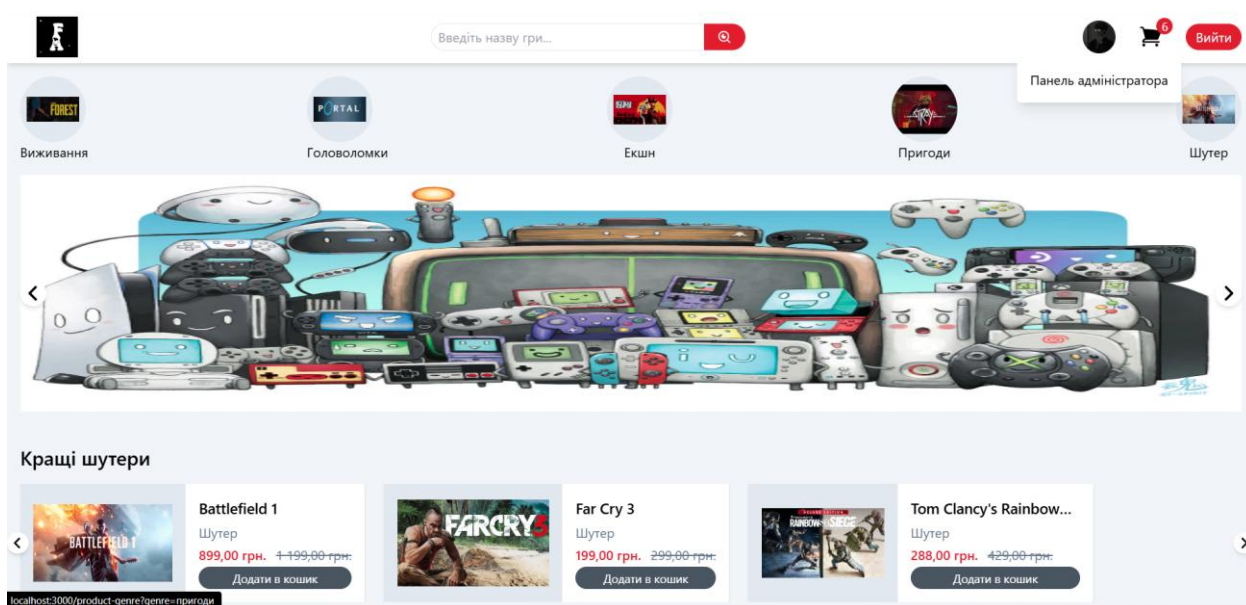


Рисунок 3.16 - Головна сторінка після входу в акаунт

Джерело: сформовано автором на основі виконаного дослідження

На сторінці кошику користувач може переглянути всі ігри які він додав до кошику, видалити зайві або ж очистити весь кошик. Також тут відображається сума до сплати всіх товарів які знаходяться в кошику. Прототип та розроблений варіант сторінки кошику зображений на рисунках 3.17 та 3.18.

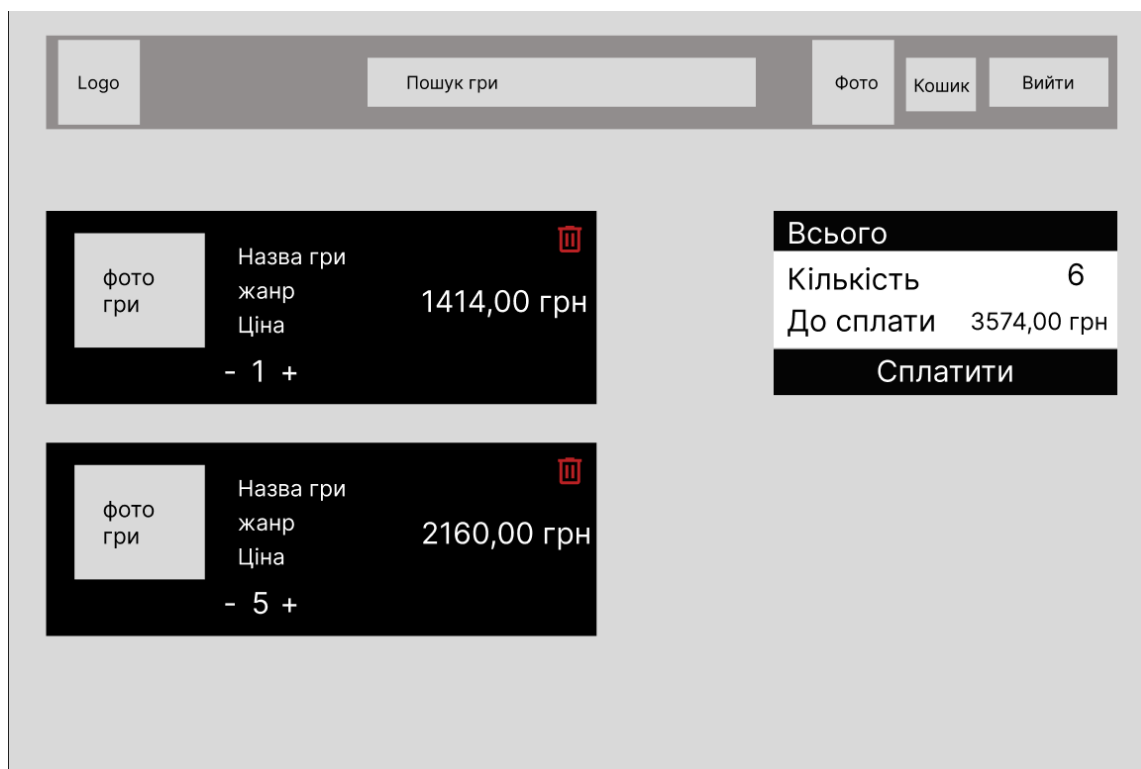


Рисунок 3.17 - Прототип сторінки кошику

Джерело: сформовано автором на основі виконаного дослідження

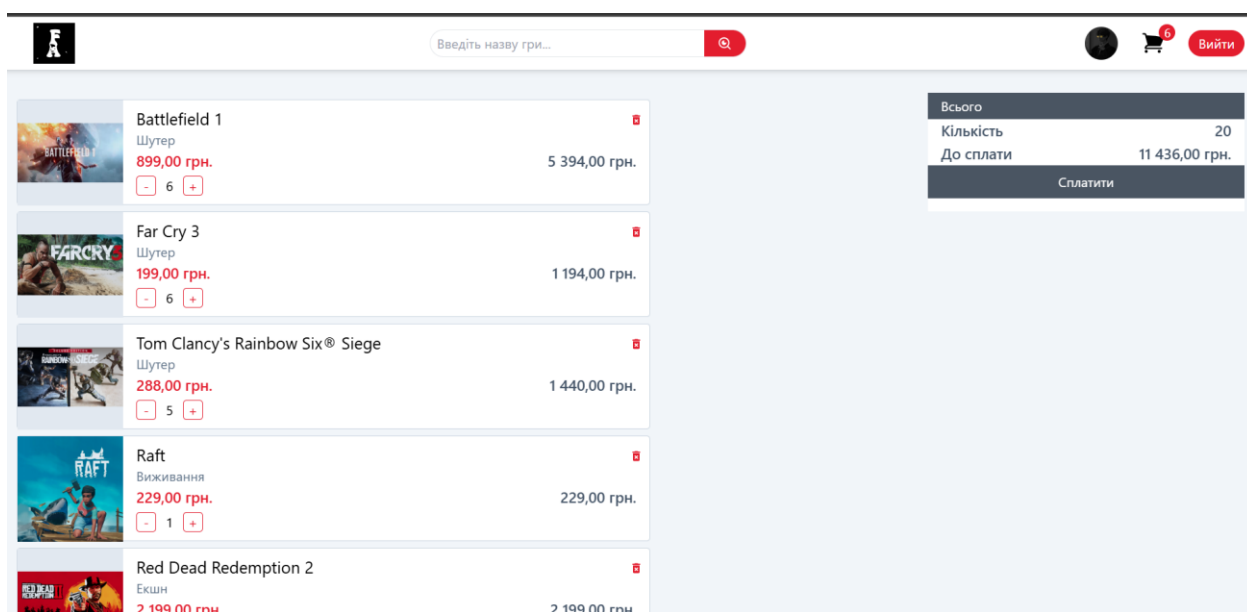


Рисунок 3.18 - Сторінка кошику

Джерело: сформовано автором на основі виконаного дослідження

Коли адміністратор переходить в панель адміністратора у нього відображається список всіх товарів які знаходяться на сайті. Прототип та розроблений варіант сторінки панелі адміністратора зображений на рисунках 3.19 та 3.20.

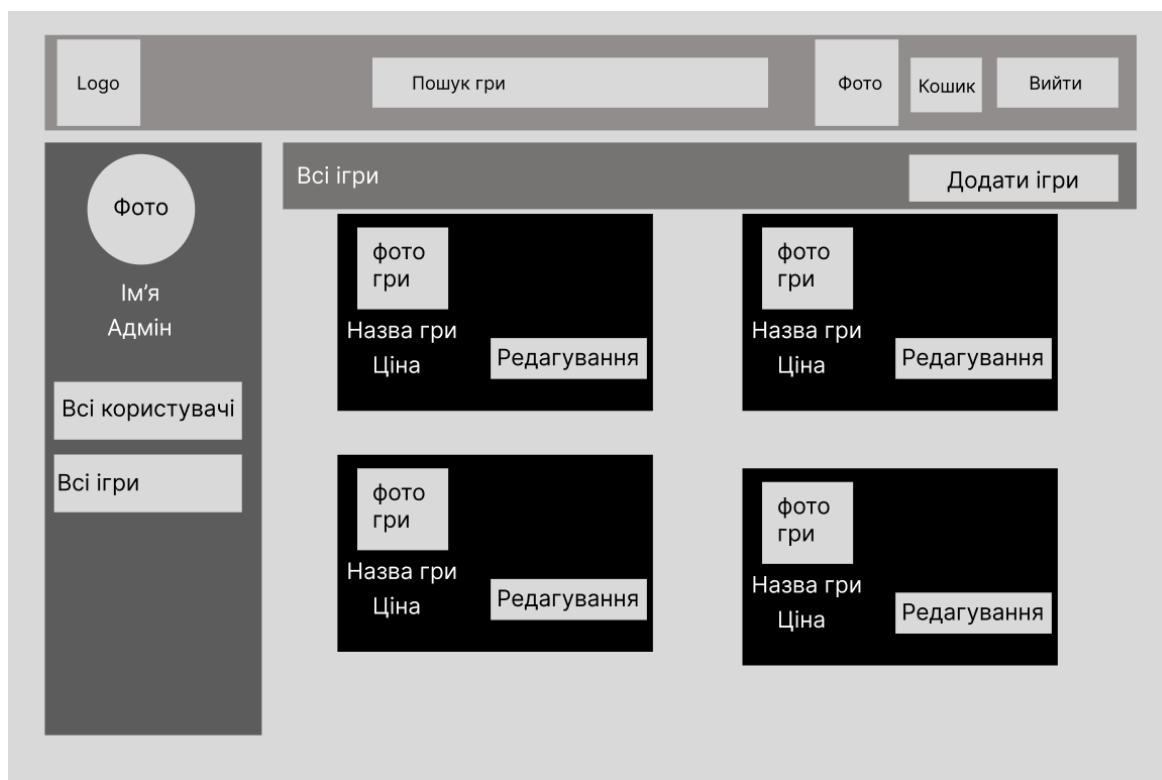


Рисунок 3.19 - Прототип сторінки панелі адміністратора

Джерело: сформовано автором на основі виконаного дослідження

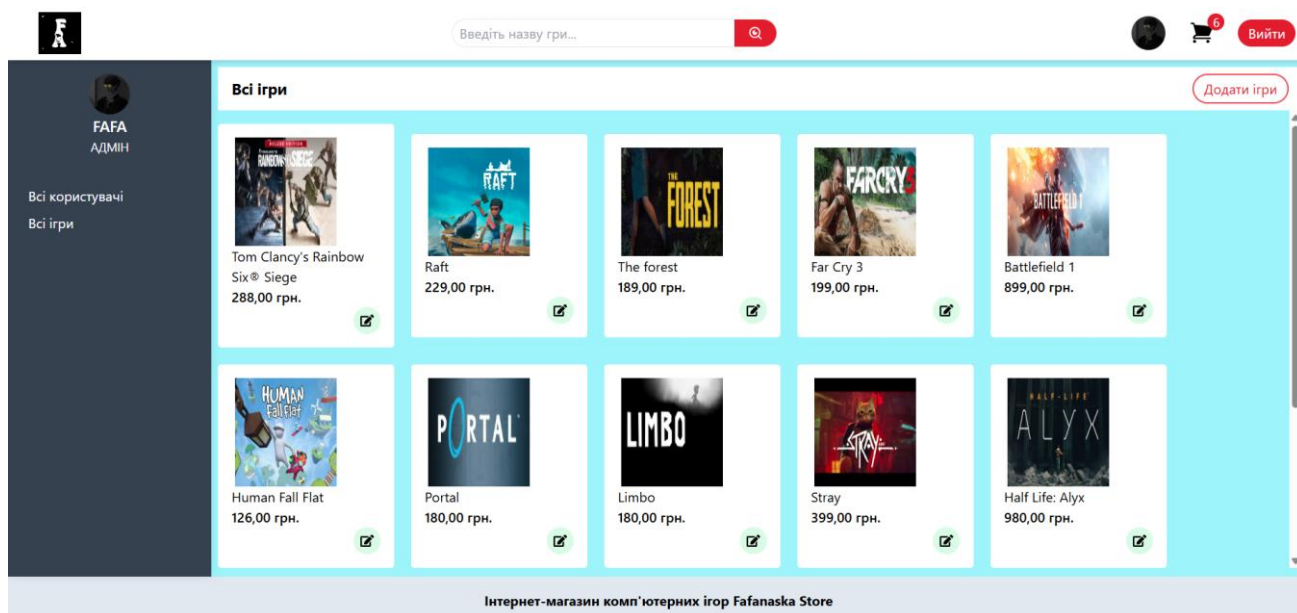


Рисунок 3.20 - Сторінка панелі адміністратора

Джерело: сформовано автором на основі виконаного дослідження

Далі у адміністратора є можливість додати товар на сайт. Прототип та розроблений варіант сторінки додавання товару на сайт зображений на рисунках 3.21 та 3.22.

Рисунок 3.21 - Прототип сторінки додавання товару на сайт

Джерело: сформовано автором на основі виконаного дослідження

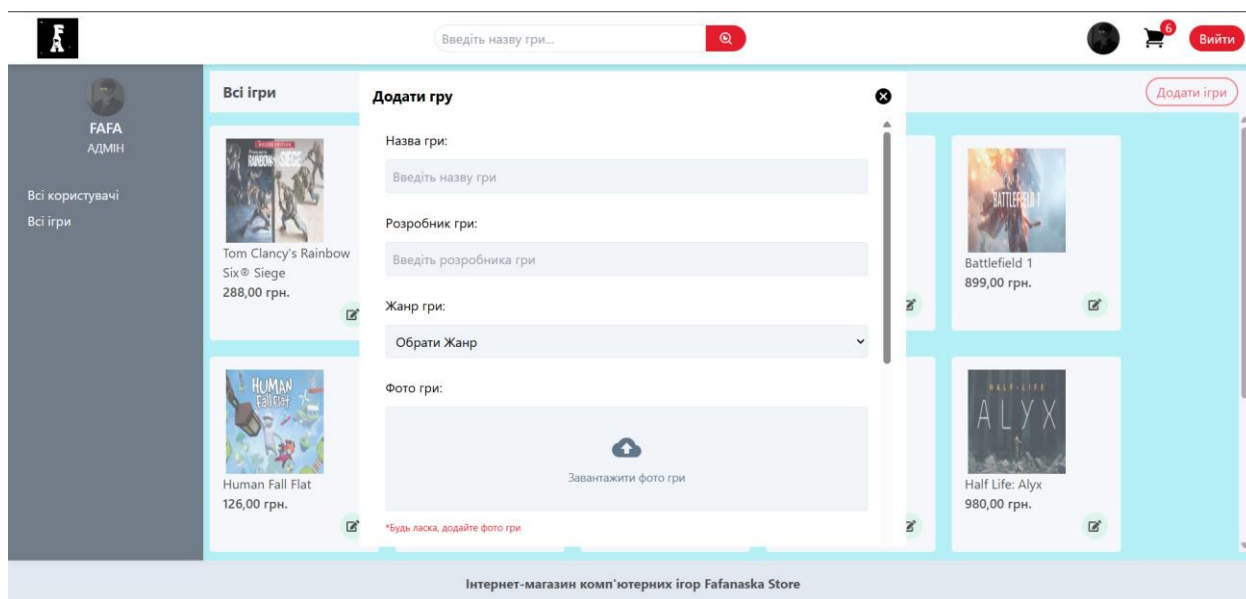


Рисунок 3.22 - Сторінка додавання товару на сайт

Джерело: сформовано автором на основі виконаного дослідження

Також є можливість редагувати товар на сайті якщо була допущена якась помилка при написанні або змінити ціну. Прототип та розроблений варіант сторінки редагування товару на сайті зображений на рисунках 3.23 та 3.24.

Рисунок 3.23 - Прототип сторінки редагування товару

Джерело: сформовано автором на основі виконаного дослідження

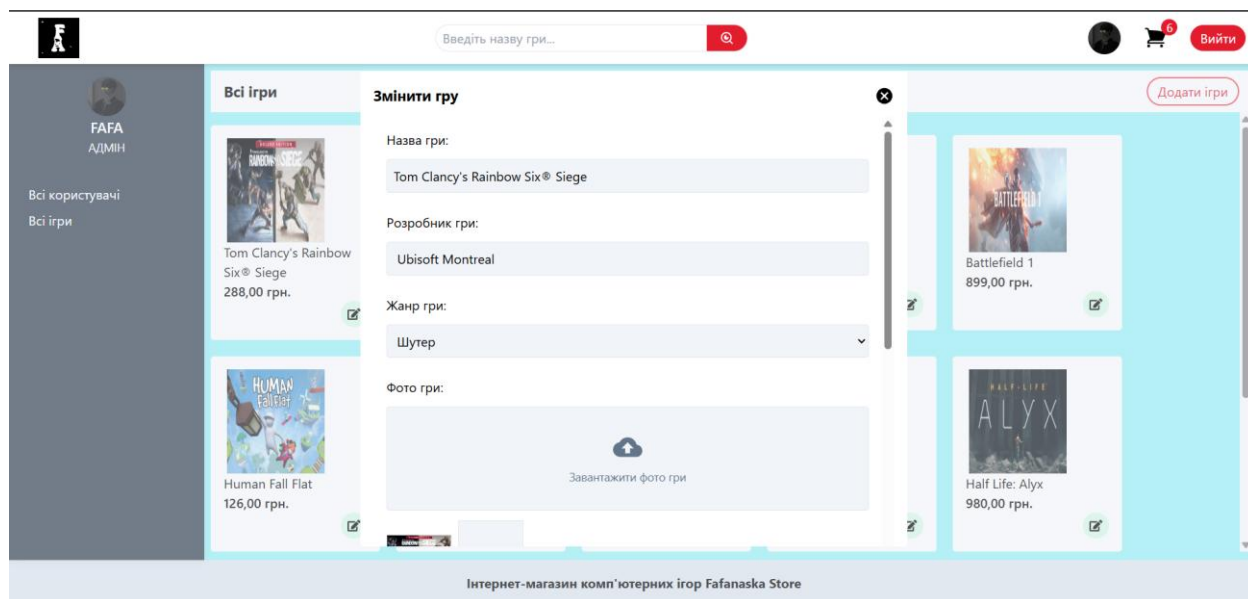


Рисунок 3.24 - Сторінка редагування товару

Джерело: сформовано автором на основі виконаного дослідження

Ще одна можливість адміністраторської панелі – переглядання всіх зареєстрованих користувачів. Прототип та розроблений варіант сторінки перегляду всіх користувачів зображений на рисунках 3.25 та 3.26.

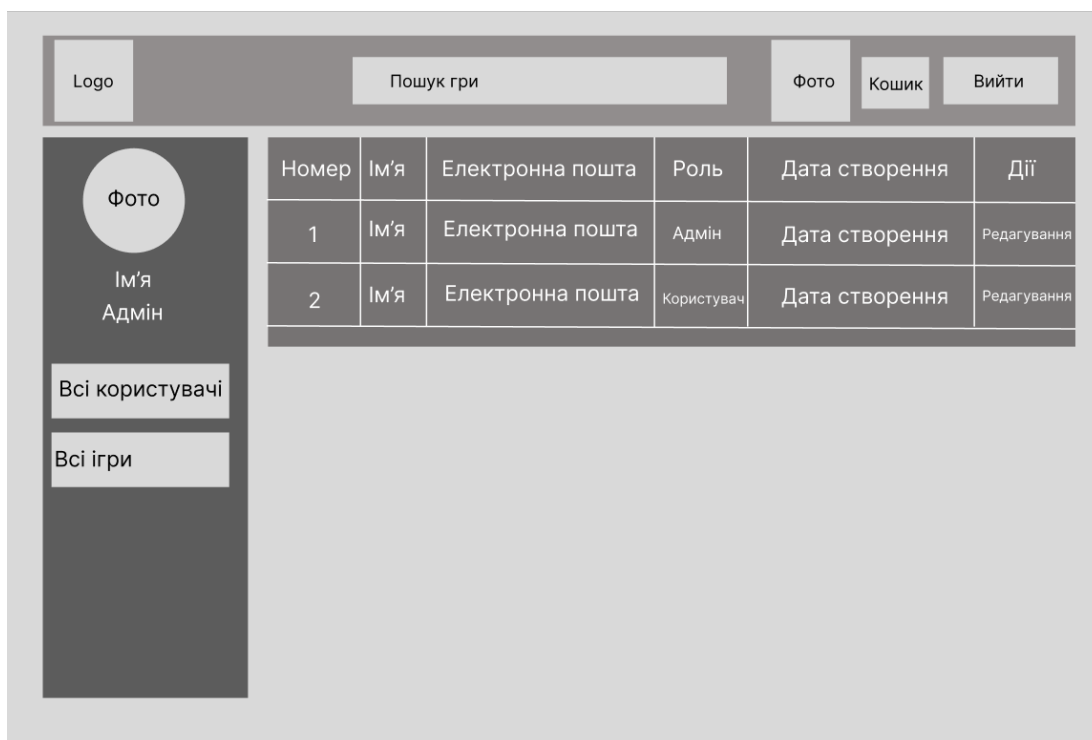


Рисунок 3.25 - Прототип сторінки перегляду всіх користувачів

Джерело: сформовано автором на основі виконаного дослідження

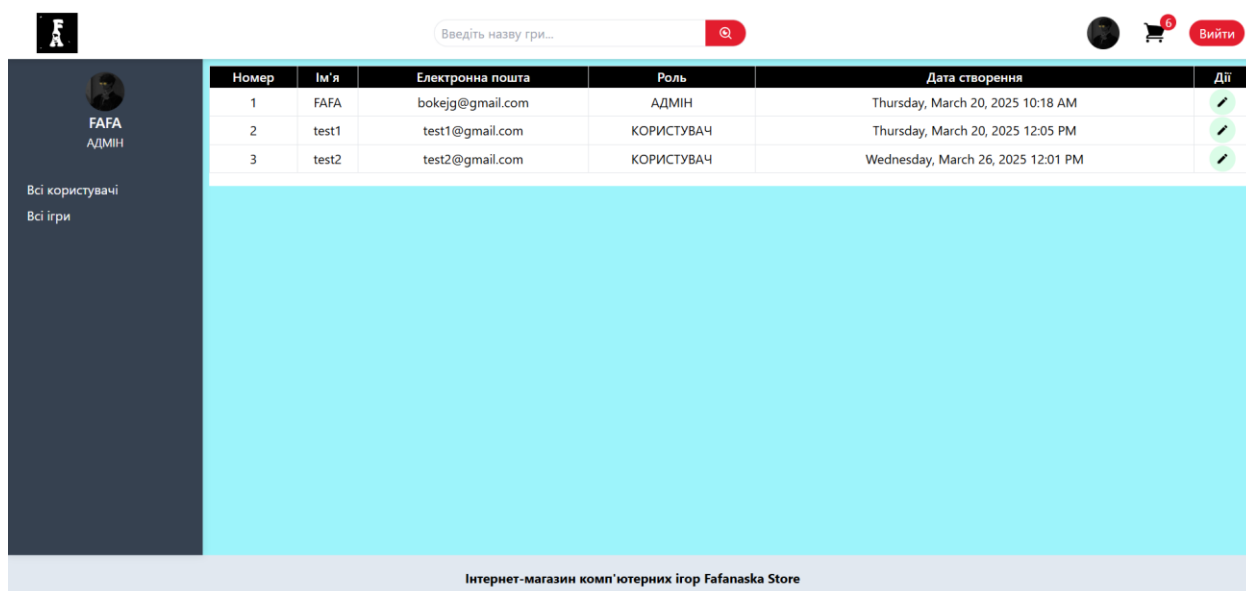


Рисунок 3.26 - Сторінка перегляду всіх користувачів

Джерело: сформовано автором на основі виконаного дослідження

Окрім перегляду всіх користувачів, адміністратор має можливість змінювати роль на адміністратора або на користувача. Прототип та розроблений варіант сторінки редагування ролі користувача зображений на рисунках 3.27 та 3.28.

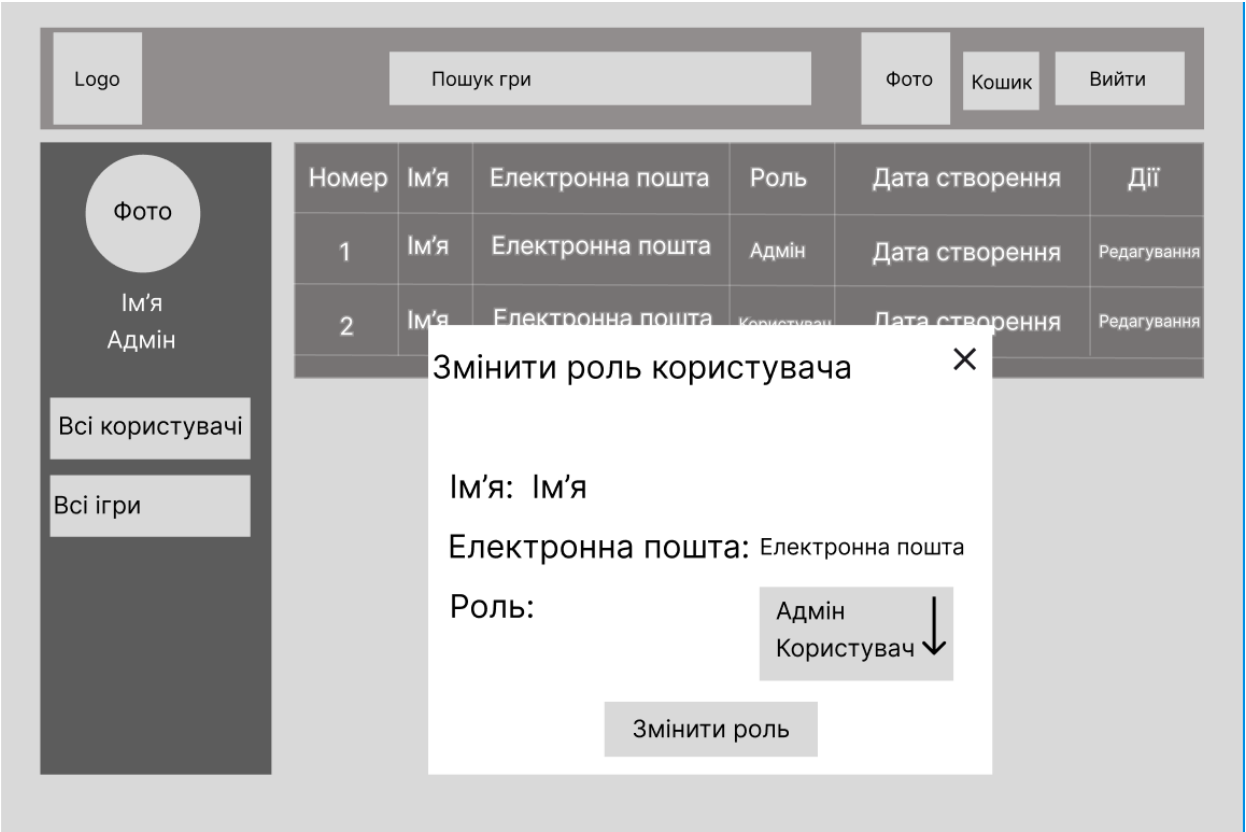


Рисунок 3.27 - Прототип сторінки редагування ролі користувача

Джерело: сформовано автором на основі виконаного дослідження

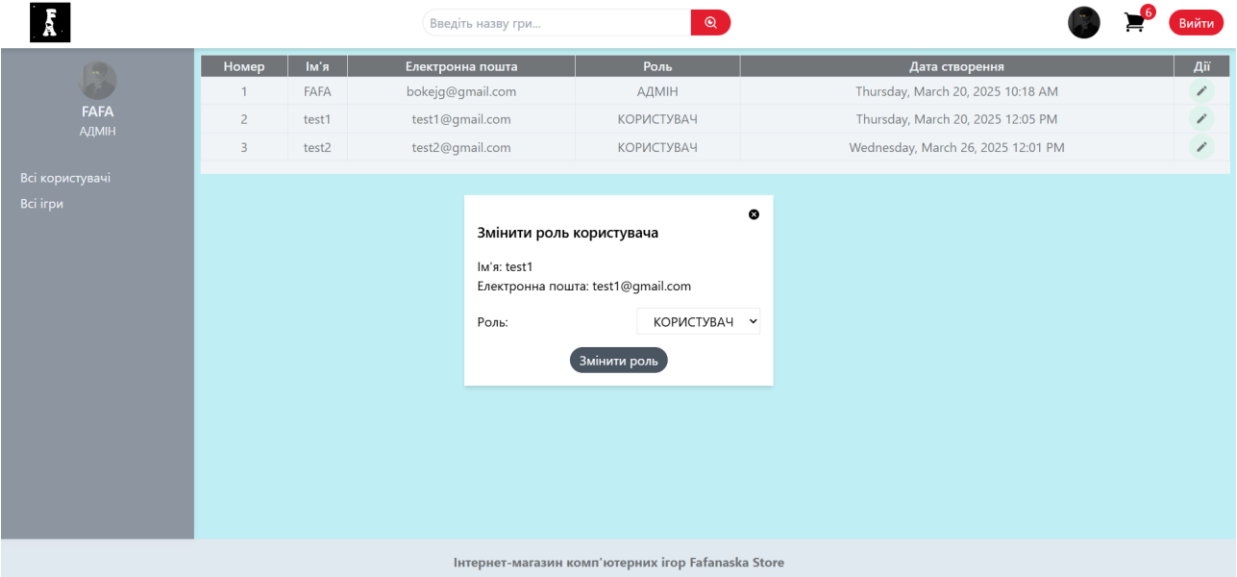


Рисунок 3.28 - Сторінка редагування ролі користувача

Джерело: сформовано автором на основі виконаного дослідження

І для зручної навігації по сайту та для його комфортної розробки був створений клікабельний прототип для розуміння переходів між сторінками. Клікабельний прототип зображений на рисунку 3.29.

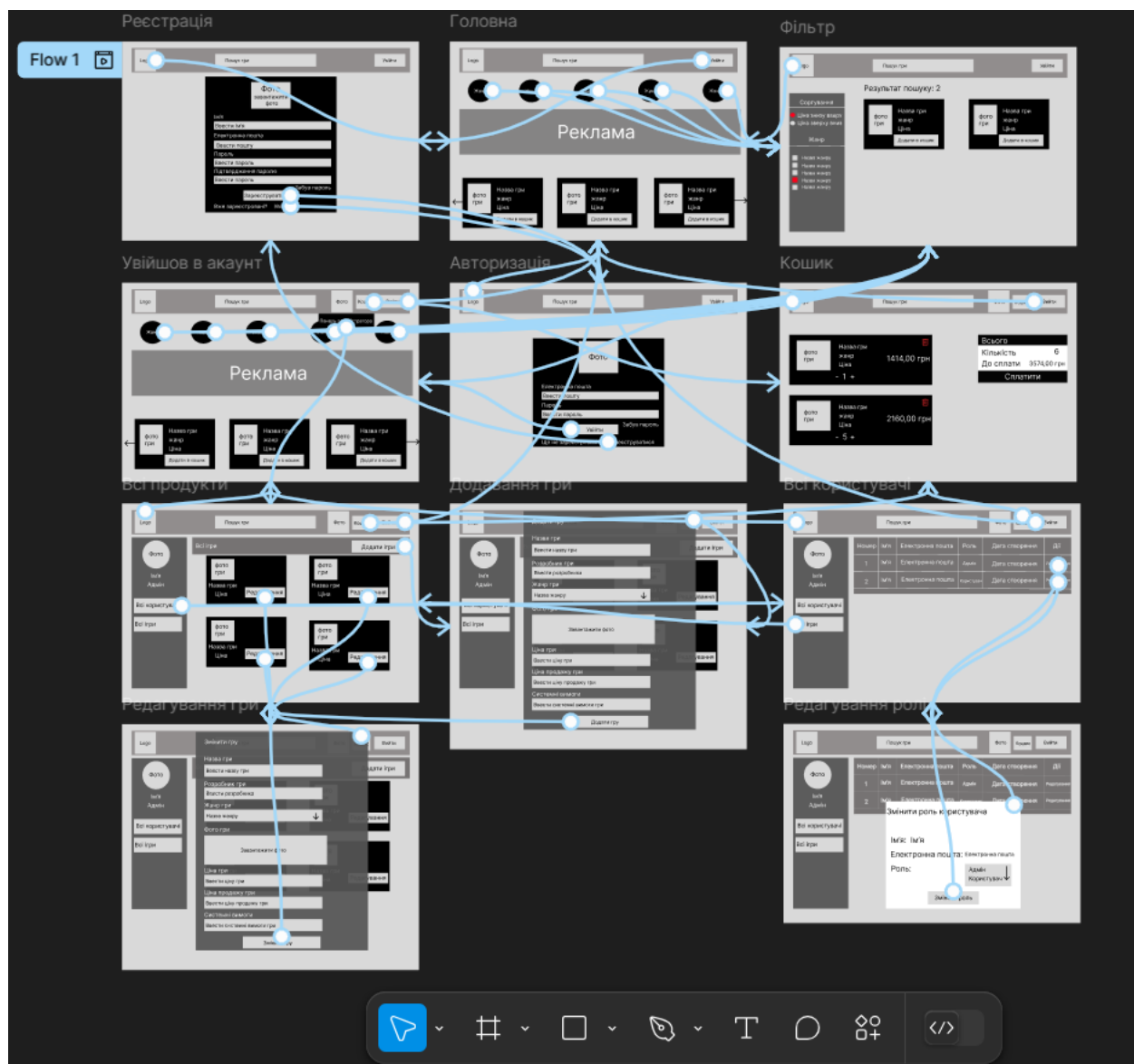


Рисунок 3.29 - Клікабельний прототип у застосунку Figma

Джерело: сформовано автором на основі виконаного дослідження

В додатку Б зазначені фрагменти коду, використані для створення сайту. В додатку В показане зберігання даних про користувачів, товари та кошик в базі даних.

ВИСНОВКИ

Основною метою кваліфікаційної бакалаврської роботи було проєктування інформаційної системи інтернет-магазину комп'ютерних ігор що можемо вважати виконаним в ході роботи.

Було розроблено клікабельний прототип системи у середовищі Figma, а за допомогою онлайн-утиліт DrawIO та LucidChart було описано варіанти використання роботи користувача із системою.

Діаграми визначення блоків та внутрішніх блоків надали можливість відобразити структуру системи, а діаграма трасування та матриця взаємозв'язків забезпечили покриття вимог до системи. Розроблені тест-кейси дозволили перевірити сценарії користувача, такі як реєстрація, авторизація, пошук товарів, додавання товарів до кошика та адміністративне управління. Модульна структура системи дозволяє легко розширювати її функціональність. Інтуїтивно зрозумілий користувацький інтерфейс спрощує взаємодію користувачів з системою, підвищуючи задоволеність клієнтів. Впровадження цієї системи може супроводжуватися навчальними матеріалами для адміністраторів та користувачів, щоб максимально спростити перехід на нову платформу. Розроблена система демонструє практичне значення автоматичного інжинірингу та підтверджує використання сучасних методологій для створення інформаційних систем.

В першому розділі ми охарактеризували предметну область, було проведено порівняння та аналіз існуючих інформаційних систем.

В другому розділі були проаналізовані вимоги до інформаційної системи і змодельована інформаційна система.

В третьому розділі було описано інформаційне забезпечення задачі, описано проєктні рішення, було проілюстровано структуру прикладного програмного забезпечення у діаграмі компонентів, а також відображено фізичну архітектуру системи в діаграмі розгортання. Також були представлені результати розробки інформаційної системи інтернет-магазину комп'ютерних ігор.

Інформаційна система інтернет-магазину комп'ютерних ігор забезпечує низку переваг, які роблять роботу магазину ефективнішою, покращують користувацький досвід і підтримують бізнес-процеси. Автоматизація обробки замовлень, управління товарними запасами та фінансових операцій дозволяє зменшити витрати на ручну працю та знижує ризик помилок. Це дає змогу обробляти великий обсяг замовлень навіть під час пікових навантажень. Користувачі отримують можливість зручно купувати ігри будь-коли і з будь-якого місця, а такі функції, як кошик і рекомендації, роблять процес покупки приємнішим та стимулюють до додаткових покупок.

Інформаційна система інтернет-магазину комп'ютерних ігор забезпечує автоматизацію ключових бізнес-процесів: від реєстрації користувачів до обробки замовлень і проведення платежів. В процесі виконання роботи також були створені інфологічна та даталогічна моделі бази даних, які відображають логічну структуру предметної області і відповідають нормалізованим вимогам щодо зберігання інформації. Модель розроблена в середовищі MongoDB дозволила створити адаптивну та розширювальну структуру даних. Створений вебінтерфейс на платформі React.js гарантує простоту використання системи як для користувачів, так і для адміністратора. Інформаційна система включає функціональні модулі для авторизації, пошуку, управління товарами, замовленнями та аналізу. Гарантовано захист приватної інформації, контроль доступу, шифрування та резервне копіювання, що відповідає сучасним стандартам безпеки. Система перевірена на відповідність функціональним і технічним стандартам, показує стабільність роботи та здатність до масштабування. Реалізоване рішення відзначається швидкістю виконання, гнучкістю структури, проте потребує доопрацювання в аспекті інтеграції з платіжними і логістичними сервісами. Описано можливості подальшого розвитку системи, зокрема реалізацію рекомендаційних механізмів, підтримку кількох мов та мобільний формат. Проєкт був розроблений самостійно за допомогою сучасних інформаційних технологій та хмарної інфраструктури, з урахуванням сучасних вимог до електронної комерції. Отримані результати мають практичну значущість і можуть слугувати основою для запуску

повноцінного онлайн-магазину або для подальшого наукового вивчення в сфері інформаційних технологій.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Class Diagram. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/>
2. Cloudinary. [Електронний ресурс] – Режим доступу до ресурсу:
<https://hellip.com/ua/product/cloudinary.html>
3. Dotenv. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.npmjs.com/package/dotenv>
4. Drawio. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.drawio.com/>
5. Epic Games. [Електронний ресурс] – Режим доступу до ресурсу:
<https://store.epicgames.com/en-US/>
6. Expressjs. [Електронний ресурс] – Режим доступу до ресурсу:
<https://expressjs.com/>
7. Figma. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.figma.com/>
8. Jsonwebtoken. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.npmjs.com/package/jsonwebtoken>
9. Lucidchart. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.lucidchart.com/pages/>
10. MongoDB. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/>
11. MongoDB. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.mongodb.com/>
12. Mongoose. [Електронний ресурс] – Режим доступу до ресурсу:
<https://mongoosejs.com/>
13. Nodejs. [Електронний ресурс] – Режим доступу до ресурсу:
<https://nodejs.org/en>
14. Origin. [Електронний ресурс] – Режим доступу до ресурсу:
<https://get-origin.ru/>

15. React. [Електронний ресурс] – Режим доступу до ресурсу:
<https://react.dev/>
16. Redux. [Електронний ресурс] – Режим доступу до ресурсу:
<https://react-redux.js.org/introduction/getting-started>
17. Steam. [Електронний ресурс] – Режим доступу до ресурсу:
<https://store.steampowered.com/?l=en>
18. Visual Studio. [Електронний ресурс] – Режим доступу до ресурсу:
<https://code.visualstudio.com/>
19. What's up with buying PC game keys. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.quora.com/Whats-up-with-buying-PC-game-keys-way-cheaper-than-an-actual-copy>
20. Why are digital keys cheaper? [Електронний ресурс] – Режим доступу до ресурсу: <https://k4g.com/blog/games/why-are-digital-keys-cheaper>
21. Why Use MongoDB. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb>
22. Why Visual Studio Code. [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs/editor/whyvscode>
23. Why You Should Use React. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development>
24. Бізнес вимоги до системи [Електронний ресурс] – Режим доступу до ресурсу: <https://softwaredominos.com/home/software-design-development-articles/business-requirements-an-essential-guide-to-definition-and-application-in-it-projects/#:~:text=Business%20requirements%20are%20explicit%20statements,%2C%20security%2C%20and%20user%20experience.>
25. Етапи процесу розробки програмного забезпечення. [Електронний ресурс] – Режим доступу до ресурсу:
<https://studfile.net/preview/9404039/>

26. Концептуальне моделювання з використанням об'єктно-процесної методології [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/234827702_Conceptual_modeling_with_the_object-process_methodology_in_software_architecture

27. Мова моделювання систем [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/searchcio/definition/Systems-Modeling-Language-SysML>

28. Об'єктно-орієнтований підхід до програмування. [Електронний ресурс] – Режим доступу до ресурсу: <http://www.znannya.org/?view=csharp-oop>

29. Основні етапи і задачі об'єктно-орієнтованого проєктування. [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/9404039/page:7/>

30. Функціональні та нефункціональні вимоги до системи [Електронний ресурс] – Режим доступу до ресурсу: <https://specinnovations.com/blog/non-functional-vs.-functional-requirements-when-to-use-each-type>

ДОДАТОК А

Тестова документація

TESTING
 Basic Path Scenario test. Реєстрація користувача
Description: Відкрити сторінку реєстрації. Ввести дані користувача (ім'я, email, пароль). Натиснути кнопку "Зареєструватися".
Results: Користувач успішно зареєстрований, система перенаправляє на головну сторінку.

Рисунок А.1 - Тест кейс реєстрація нового користувача

TESTING
 Standard Unit test. Авторизація
Description Відкрити сторінку входу. Ввести email та пароль. Натиснути кнопку "Увійти".
Input
Acceptance Criteria
Results Користувач успішно авторизований, система перенаправляє на головну сторінку.

Рисунок А.2 - Тест кейс авторизація користувача

TESTING
 Standard Unit test. Додавання товару до кошика
Description Вибрати товар у каталозі. Натиснути кнопку "Додати до кошика".
Input
Acceptance Criteria
Results Товар додано до кошика, його кількість відображена.

Рисунок А.3 - Тест кейс додавання товару до кошика

TESTING

Standard Unit test. Видалення товару з кошика

Description

Відкрити кошик.
Натиснути кнопку "Видалити" біля товару.

Input

Acceptance Criteria

Results

Товар видалено з кошика, загальна сума оновлена.

Рисунок А.4 - Тест кейс видалення товару з кошика

TESTING

Standard Unit test. Пошук товару

Description

Ввести ключове слово у поле пошуку (наприклад, "Raft").
Натиснути кнопку "Пошук".

Input

Acceptance Criteria

Results

Система відображає список товарів, що відповідають пошуковому запиту.
[Last run Not

Рисунок А.5 - Тест кейс пошук товару

TESTING

Standard Unit test. Додавання нового товару

Description

Перейти в панель адміністратора
Перейти до розділу "Всі ігри".
Ввести дані товару.
Натиснути кнопку "Додати".

Input

Acceptance Criteria

Results

Товар додано до каталогу, він доступний для перегляду.

Рисунок А.6 - Тест кейс додавання нового товару

TESTING



Standard Unit test. Оновлення користувача

Description

Перейти в панель адміністратора
Перейти до списку користувачів.
Вибрати користувача.
Натиснути кнопку "Змінити роль".
Обрати роль з випадаючого списку.

Input

Acceptance Criteria

Results

Роль користувача оновлено.

Рисунок А.7 - Тест кейс оновлення користувача

ДОДАТОК Б

Код програми

Код Б.1 – Код головної сторінки сайту.

```
import React from 'react'
import GenreList from '../components/GenreList'
import BannerProduct from '../components/BannerProduct'
import HorizontalCardProduct from '../components/HorizontalCardProduct'
import VerticalCardProduct from '../components/VerticalCardProduct'

const Home = () => {
  return (
    <div>
      <GenreList/>
      <BannerProduct/>

      <HorizontalCardProduct genre={"шутер"} heading={"Кращі шутери"}/>
      <HorizontalCardProduct genre={"головоломки"} heading={"Популярні
головоломки"}/>

      <VerticalCardProduct genre={"виживання"} heading={"Виживання"}/>
      <VerticalCardProduct genre={"екшн"} heading={"Екшн"}/>
      <VerticalCardProduct genre={"пригоди"} heading={"Пригоди"}/>
    </div>
  )
}

export default Home
```

Код Б.2 – Код сторінки реєстрації.

```
import React from 'react'
import loginIcons from '../assest/signin.png'
import { useState } from 'react'
import { IoEye } from "react-icons/io5";
import { IoEyeOff } from "react-icons/io5";
import { Link, useNavigate } from 'react-router-dom';
import imageTobase64 from '../helpers/imageTobase64';
import SummaryApi from '../common';
import { toast } from 'react-toastify';

const SignUp = () => {
  const [showPassword, setShowPassword] = useState(false)
  const [showConfirmPassword, setConfirmPassword] = useState(false)
  const [data, setData] = useState({
    email: "",
    password: "",
    name: "",
    confirmPassword: "",
    profilePic: "",
  })
  const navigate = useNavigate()

  const handleChange = (e) =>{
    const {name, value } = e.target

    setData((preve) =>{
      return{
        ...preve,
        [name] : value
      }
    })
  }

  const handleUploadPic = async(e) =>{
    const file = e.target.files[0]

    const imagePic = await imageTobase64(file)
    setData((preve) =>{
      return{
        ...preve,
        profilePic : imagePic
      }
    })
  }
}
```

```

    }
  })

}

const handleSubmit = async(e) =>{
  e.preventDefault()

  if(data.password === data.confirmPassword){
    const dataResponse = await fetch(SummaryApi.signUP.url,{
      method : SummaryApi.signUP.method,
      headers : {
        "content-type" : "application/json"
      },
      body : JSON.stringify(data)
    })

    const dataApi = await dataResponse.json()

    if(dataApi.success){
      toast.success(dataApi.message)
      navigate("/login")
    }

    if (dataApi.error){
      toast.error(dataApi.message)
    }

    }else{
      toast.error("Перевірте пароль та підтвердіть його.")
    }
  }

  console.log("data login", data)
  return (
    <section id ='sign-up'>
      <div className='mx-auto container p-4'>

        <div className='bg-black p-4 w-full max-w-sm mx-auto'>

          <div className='w-20 h-20 mx-auto relative overflow-
hidden rounded-full'>

            <div>

```

```

                                <img src={data.profilePic || loginIcons}
alt='login icons' />

                                </div>
                                <form>
                                    <label>
                                        <div className='text-white text-xs bg-
opacity-90 bg-slate-400 cursor-pointer py-2 text-center absolute bottom-0 w-
full'>
                                            Завантажити аватар
                                        </div>
                                        <input type='file' className='hidden'
onChange={handleUploadPic} />
                                    </label>
                                </form>
                            </div>

                            <form className='pt-6 flex flex-col gap-2'
onSubmit={handleSubmit}>
                                <div className='grid'>
                                    <label className='text-white'>Ім'я : </label>
                                    <div className='bg-white text-black p-2'>
                                        <input
                                            type='text'
                                            placeholder='Введіть нікнейм.'
                                            name='name'
                                            value={data.name}
                                            onChange={handleOnChange}
                                            required
                                            className='w-full h-full outline-none bg-
transparent' />
                                    </div>
                                </div>
                                <div className='grid'>
                                    <label className='text-white'>Електронна пошта :
</label>
                                    <div className='bg-white text-black p-2'>
                                        <input
                                            type='email'
                                            placeholder='Введіть пошту.'
                                            name='email'
                                            value={data.email}
                                            onChange={handleOnChange}
                                            required
                                            className='w-full h-full outline-none bg-
transparent' />

```

```

        </div>
    </div>

    <div>
        <label className='text-white'> Пароль : </label>
        <div className='bg-white text-black p-2 flex'>
            <input
                type={showPassword ? "text" : "password"}
                placeholder='Введіть пароль.'
                name='password'
                value={data.password}
                onChange={handleOnChange}
                required
                className='w-full h-full outline-none bg-
transparent' />
                <div className='cursor-pointer text-xl'
onClick={ () => setShowPassword ( (preve) => !preve) }>
                    <span>
                        {
                            showPassword ? (
                                <IoEyeOff />
                            )
                            :
                            (
                                <IoEye />
                            )
                        }
                    </span>
                </div>
            </div>
        </div>

    <div>
        <label className='text-white'> Підтвердіть пароль
: </label>

        <div className='bg-white text-black p-2 flex'>
            <input
                type={showConfirmPassword ? "text" :
"password"}

                placeholder='Введіть підтверджений пароль.'
                name='confirmPassword'
                value={data.confirmPassword}
                onChange={handleOnChange}
                required

```

```

        className='w-full h-full outline-none bg-
transparent' />

        <div className='cursor-pointer text-xl'
onClick={ ()=>setConfirmPassword( (preve)=>!preve) }>
            <span>
                {
                    showConfirmPassword ? (
                        <IoEyeOff />
                    )
                    :
                    (
                        <IoEye />
                    )
                }
            </span>
        </div>
    </div>
</div>

    <button className='bg-gray-800 hover:bg-gray-700
text-white px-6 py-2 w-full max-w-[160px] rounded-full hover:scale-95
transition-all mx-auto block mt-4 text-center'> Зареєструватися </button>

</form>

    <p className='text-white my-5'> Вже зареєстровані? <Link
to={"/login"} className='hover:text-red-500 hover:underline'>Увійти</Link></p>
    </div>

</div>
</section>
)
}

export default SignUp

```

Код Б.3 – Код сторінки деталей про товар.

```
import React, { useCallback, useContext, useEffect, useState } from
'react'

import { useNavigate, useParams } from 'react-router-dom'
import SummaryApi from '../common'
import { FaStar } from "react-icons/fa";
import { FaStarHalf } from "react-icons/fa";
import displayUAHCurrency from '../helpers/displayCurrency';
import VerticalCardProduct from '../components/VerticalCardProduct';
import
                                GenreWiseProductDisplay                                from
'../components/GenreWiseProductDisplay';
import addToCart from '../helpers/addToCart';
import Context from '../context';

const ProductDetails = () => {
  const [data, setData] = useState({
    productName : "",
    developer : "",
    genre : "",
    productImage : [],
    systemRequirements : "",
    price : "",
    sellingPrice : ""
  })
  const params = useParams()
  const [loading, setLoading] = useState(true)
  const productImageListLoading = new Array(4).fill(null)
  const [activeImage, setActiveImage] = useState("")
  const [zoomImageCoordinate, setZoomImageCoordinate] = useState({
    x : 0,
    y : 0
  })
  const [zoomImage, setZoomImage] = useState(false)

  const { fetchUserAddToCart } = useContext(Context)

  const navigate = useNavigate()

  const fetchProductDetails = async()=>{
    setLoading(true)
    const response = await fetch(SummaryApi.productDetails.url,{
      method : SummaryApi.productDetails.method,
      headers : {
        "content-type" : "application/json"
```

```

    },
    body : JSON.stringify({
      productId : params?.id
    })
  })
  setLoading(false)
  const dataResponse = await response.json()

  setData(dataResponse?.data)
  setActiveImage(dataResponse?.data?.productImage[0])
}
console.log("data",data)

useEffect(()=>{
  fetchProductDetails()
},[params])

const handleMouseEnterProduct = (imageUrl)=>{
  setActiveImage(imageURL)
}

const handleZoomImage = useCallback((e) =>{
  setZoomImage(true)
  const { left , top , width , height} =
e.target.getBoundingClientRect()
  console.log("coordinate", left, top, width, height)

  const x = (e.clientX - left) / width
  const y = (e.clientY - top) / height

  setZoomImageCoordinate({
    x,
    y
  })
},[zoomImageCoordinate])

const handleLeaveImageZoom = () =>{
  setZoomImage(false)
}

const handleAddToCart = async(e,id)=>{
  await addToCart(e,id)
  fetchUserAddToCart()
}

```

```

const handleBuyProduct = async (e, id) => {
  await addToCart (e, id)
  fetchUserAddToCart ()
  navigate ("/cart")
}

return (
  <div className='container mx-auto p-4'>

    <div className='min-h-[200px] flex flex-col lg:flex-row gap-4'>
      {/**product image */}
      <div className='h-96 flex flex-col lg:flex-row-reverse gap-4'>

        <div className='h-[300px] w-[300px] lg:h-96 lg:w-96 bg-slate-200 relative p-2'>
          <img src={activeImage} className='h-full w-full object-scale-down mix-blend-multiply' onMouseMove={handleZoomImage} onMouseLeave={handleLeaveImageZoom}/>

          {/**product zoom */}
          {
            zoomImage && (
              <div className='hidden lg:block absolute min-w-[500px] overflow-hidden min-h-[400px] bg-slate-200 p-1 -right-[510px] top-0'>
                <div
                  className='w-full h-full min-h-[400px] min-w-[500px] mix-blend-multiply scale-150'
                  style={{
                    backgroundImage : `url(${activeImage})`,
                    backgroundRepeat : 'no-repeat',
                    backgroundPosition :
                      `${zoomImageCoordinate.x * 100}% ${zoomImageCoordinate.y * 100}%`

                  }}
                >

                </div>
              </div>
            )
          }

        </div>
      </div>
    </div>
  )
}

```

```

        <div className='h-full'>
          {
            loading ? (
              <div className='flex gap-3 lg:flex-col overflow-
scroll scrollbar-none h-full'>
                {
                  productImageListLoading.map((el,index) =>{
                    return(
                      <div className='h-20 w-20 bg-slate-200
rounded animate-pulse' key={"loadingImage"+index}>

                        </div>
                      )
                    })
                  }
                </div>
              ) : (
                <div className='flex gap-3 lg:flex-col overflow-
scroll scrollbar-none h-full'>
                  {
                    data?.productImage?.map((imgURL,index) =>{
                      return(
                        <div className='h-20 w-20 bg-slate-200
rounded p-1' key={imgURL}>

                          <img src={imgURL} className='w-full h-
full object-scale-down mix-blend-multiply cursor-pointer' onMouseEnter={()=>
handleMouseEnterProduct(imgURL)}                                onClick={()=>
handleMouseEnterProduct(imgURL)} />

                        </div>
                      )
                    })
                  }
                </div>
              )
            }
          </div>
        </div>

        {/**product details */}
        <div>
          {
            loading ? (
              <div className='grid gap-1 w-full'>

```

```

        <p className='bg-slate-200 animate-pulse h-6 lg:h-8 w-full
rounded-full inline-block'></p>
        <h2 className='text-2xl lg:text-4xl font-medium h-6 lg:h-8
bg-slate-200 animate-pulse w-full'></h2>
        <p className='capitalize text-slate-400 bg-slate-200 animate-
pulse min-w-[100px] h-6 lg:h-8 w-full'></p>

        <div className='text-red-600 bg-slate-200 h-6 animate-pulse
flex items-center gap-1 w-full'>
        </div>

        <div className='flex items-center gap-2 text-2xl lg:text-3xl
font-medium my-1 h-6 lg:h-8 animate-pulse w-full'>
        <p className='text-red-600 bg-slate-200 w-full'></p>
        <p className='text-slate-400 line-through bg-slate-200 w-
full'></p>
        </div>

        <div className='flex items-center gap-3 my-2 w-full'>
        <button className='h-6 lg:h-8 bg-slate-200 rounded
animate-pulse w-full'></button>
        <button className='h-6 lg:h-8 bg-slate-200 rounded
animate-pulse w-full'></button>
        </div>

        <div className='w-full'>
        <p className='text-slate-600 font-medium my-1 h-6 lg:h-8
bg-slate-200 rounded animate-pulse w-full'></p>
        <p className='bg-slate-200 rounded animate-pulse h-10 lg:h-
12 w-full'></p>
        </div>
    </div>
    ) : (
        <div className='flex flex-col gap-1'>
        <p className='bg-red-200 text-red-600 px-2 rounded-full
inline-block w-fit '>{data?.developer}</p>
        <h2 className='text-2xl lg:text-4xl font-
medium'>{data?.productName}</h2>
        <p className='capitalize text-slate-400'>{data?.genre}</p>

        <div className='text-red-600 flex items-center gap-1'>
        <FaStar/>
        <FaStar/>
        <FaStar/>
        <FaStar/>
    
```

```

        <FaStarHalf/>
      </div>

      <div className='flex items-center gap-2 text-2xl lg:text-3xl
font-medium my-1'>
        <p
                                className='text-red-
600'>{displayUAHCurrency(data?.sellingPrice)}</p>
        <p
                                className='text-slate-400
                                line-
through'>{displayUAHCurrency(data?.price)}</p>
      </div>

      <div className='flex items-center gap-3 my-2'>
        <button className='border-2 border-red-600 rounded px-3
py-1 min-w-[120px] text-red-600 font-medium hover:bg-red-600 hover:text-
white'onClick={ (e)=>handleBuyProduct(e,data?._id)} >Придбати</button>
        <button className='border-2 border-red-600 rounded px-3
py-1 min-w-[120px] font-medium text-white bg-red-600 hover:text-red-600
hover:bg-white'onClick={ (e)=>handleAddToCart(e,data?._id)}>Додати
        в
        кошик</button>
      </div>

      <div>
        <p className='text-slate-600 font-medium my-1'>Системні
вимоги: </p>
        <p>{data?.systemRequirements}</p>
      </div>
    </div>
  )
}
</div>

</div>

{
  data.genre && (
    <GenreWiseProductDisplay
                                genre={data?.genre}
heading={"Рекомендовані ігри"}/>
  )
}

</div>
)
}

```

```
export default ProductDetails
```

ДОДАТОК В

Зберігання даних в базі даних

QUERY RESULTS: 1-3 OF 3

<pre>{ "_id": ObjectId("67dbcf39d9b0fd583b5fbd08"), "name": "FAFA", "email": "bokej@gmail.com", "password": "\$2b\$10\$CMi0HcE2UfeM6FY3J3iKusTxxAqJIsTMV0gpPv3XbyA1Fwbvcg4e", "profilePic": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAQABIAAD/2wBDAAIEBQYFBAYGBQY_", "role": "ADMIN", "createdAt": 2025-03-20T08:18:01.857+00:00, "updatedAt": 2025-03-20T10:14:43.603+00:00, "__v": 0 }</pre>
<pre>{ "_id": ObjectId("67dbe84d31bcac1b26c5bb7"), "name": "test1", "email": "test1@gmail.com", "password": "\$2b\$10\$fGqHSHs4jI/QDdhaLQ8vvedfdyI/9bE9CGw/NoORGkerckNNIB6cW", "profilePic": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAQABIAAD/2wBDAAIEBQYFBAYGBQY_", "role": "KODPCTVBAY", "createdAt": 2025-03-20T10:05:01.532+00:00, "updatedAt": 2025-03-20T17:06:06.389+00:00, "__v": 0 }</pre>
<pre>{ "_id": ObjectId("67e3d063414f07ef3919b990"), "name": "test2", "email": "test2@gmail.com", "password": "\$2b\$10\$f34.gWVy.HDNNQ66y2SIIecdhf5FIkG1rNjvi2Adezfq4o8EOccr0", "profilePic": "", "role": "KODPCTVBAY", "createdAt": 2025-03-20T10:05:01.532+00:00, "updatedAt": 2025-03-20T17:06:06.389+00:00, "__v": 0 }</pre>

Рисунок В.1 - Дані про користувачів

QUERY RESULTS: 1-11 OF 11

<pre>{ "_id": ObjectId("67dd38e7aae6e29cb6b6d44e"), "productName": "Red Dead Redemption 2", "developer": "Rockstar Games", "genre": "екшн", "productImage": Array(2), "systemRequirements": "Мінімальні: 64-розрядні процесор та операційна система ОС:Windows 10 -", "price": 2599, "sellingPrice": 2199, "createdAt": 2025-03-21T10:01:11.738+00:00, "updatedAt": 2025-03-21T10:01:11.738+00:00, "__v": 0 }</pre>
<pre>{ "_id": ObjectId("67dd3b4caae6e29cb6b6d452"), "productName": "Half Life: Alyx", "developer": "Valve", "genre": "екшн", "productImage": Array(2), "systemRequirements": "Мінімальні: 64-розрядні процесор та операційна система ОС:Windows 10 П.", "price": 1195, "sellingPrice": 1195, "createdAt": 2025-03-21T10:01:11.738+00:00, "updatedAt": 2025-03-21T10:01:11.738+00:00, "__v": 0 }</pre>

Рисунок В.2 - Дані про товари

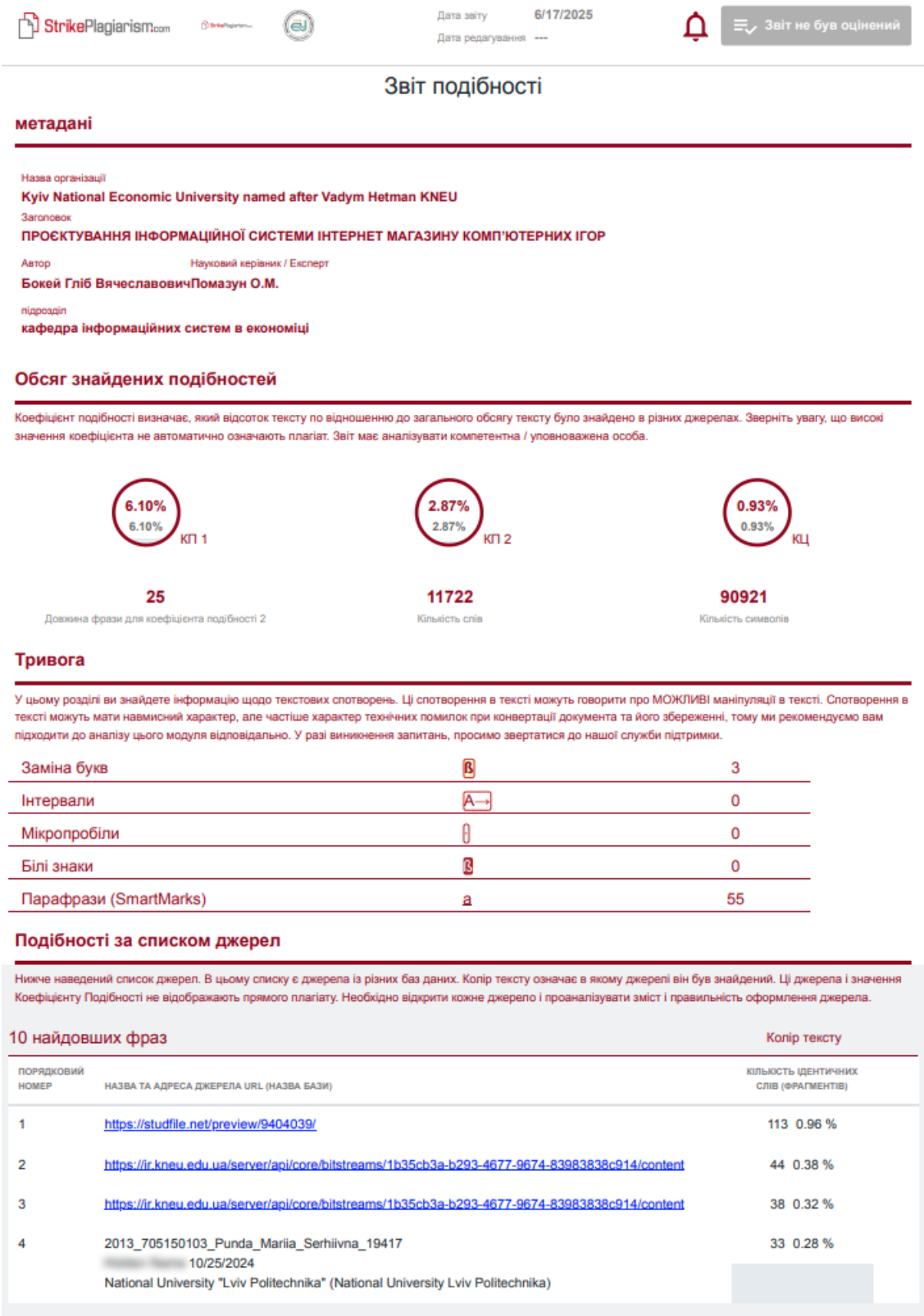
QUERY RESULTS: 1-8 OF 8

<pre>{ "_id": ObjectId("67dfc6063ec0a6bb1a5630ae"), "productId": "67dd3de8aae6e29cb6b6d466", "quantity": 6, "userId": "67dbcf39d9b0fd583b5fbd08", "createdAt": 2025-03-23T08:27:50.649+00:00, "updatedAt": 2025-03-23T11:32:30.664+00:00, "__v": 0 }</pre>
<pre>{ "_id": ObjectId("67dfc60b3ec0a6bb1a5630cb"), "productId": "67dd3ebbaae6e29cb6b6d46a", "quantity": 6, "userId": "67dbcf39d9b0fd583b5fbd08", "createdAt": 2025-03-23T08:27:55.782+00:00, "updatedAt": 2025-03-23T08:28:03.420+00:00, "__v": 0 }</pre>
<pre>{ "_id": ObjectId("67dfef09886aca0315d06b9f"), "productId": "67dfd0a612f161daad48379e", "quantity": 5, "userId": "67dbcf39d9b0fd583b5fbd08", "createdAt": 2025-03-23T11:22:49.811+00:00, "updatedAt": 2025-03-23T11:22:49.811+00:00, "__v": 0 }</pre>

Рисунок В.3 - Дані про кошик

ДОДАТОК Г

Результат перевірки на плагіат



5	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3566a2/download	32 0.27 %
6	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekom_en_6_050101.doc	26 0.22 %
7	https://ua.kursoviki.com.ua/metodychni-vkazivky/article_post/1249-metodychni-vkazivki-dlya-vikonannya-kursoвого-проекту-z-distiplini-sistemi-obroblennya-yekonomichnoi-informatsii-kneu	25 0.21 %
8	https://ir.kneu.edu.ua/bitstreams/b4dc7446-58fa-4498-aeaa-4d6afe087057/download	25 0.21 %
9	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekom_en_6_050101.doc	24 0.20 %
10	2013_705150103_Punda_Maria_Serhiivna_19417 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	22 0.19 %
з бази даних RefBooks (0.00 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
з домашньої бази даних (0.04 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Інформаційна система автоматизації процесів малого підприємства на прикладі кав'ярні 6/11/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	5 (1) 0.04 %
з програми обміну базами даних (0.86 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	2013_705150103_Punda_Maria_Serhiivna_19417 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	55 (2) 0.47 %
2	Веб-орієнтована система «medicine manager» засобами Angular JS 5/30/2022 National Forestry University of Ukraine (ЦДН НЛТУ України)	27 (2) 0.23 %
3	ФККПІ_2021_1236ак_Джурасв_ОВ 7/11/2024 Ukrainian national aviation university (Ukrainian national aviation university)	19 (2) 0.16 %
з Інтернету (5.20 %)		
ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://studfile.net/preview/9404039/	113 (1) 0.96 %
2	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekomen_6_050101.doc	105 (6) 0.90 %
3	https://ir.kneu.edu.ua/server/api/core/bitstreams/1b35cb3a-b293-4677-9674-83983838c914/content	82 (2) 0.70 %
4	https://ir.kneu.edu.ua/bitstreams/22acc4f6-0461-4d18-a9a4-ff2fbd8c10ff/download	59 (5) 0.50 %
5	https://ir.kneu.edu.ua/bitstreams/40f69492-bf2d-4dcb-b3e1-7da1833a5b34/download	50 (4) 0.43 %
6	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3566a2/download	32 (1) 0.27 %

Рисунок Г.2 - Результат перевірки на плагіат

7	http://reposit.nupp.edu.ua/bitstream/PoltNTU/4431/1/%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA_%D0%A2%D0%A0%D0%9F%D0%97_%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%B5%D0%BD%D0%BE-converted.pdf	31 (2) 0.26 %
8	https://ir.kneu.edu.ua/bitstreams/b4dc7446-58fa-4498-aeaa-4d6afe087057/download	25 (1) 0.21 %
9	https://ua.kursoviki.com.ua/metodychni_vkazivky/article_post/1249-metodichni-vkazivki-dlya-vikonannya-kurovogo-proyektu-z-distiplini-sistemi-obroblennya-yekonomichnoi-informatsii-kneu	25 (1) 0.21 %
10	https://ir.kneu.edu.ua/server/api/core/bitstreams/8886d65e-0996-456f-8b04-70b8d9683e0e/content	22 (1) 0.19 %
11	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	20 (1) 0.17 %
12	https://etiq.vntu.edu.ua/repository/getfile.php/2391.pdf	19 (2) 0.16 %
13	https://elartu.tntu.edu.ua/bitstream/lib/48325/1/Master_Thesis_SB-62_Ratyshyn_M_O_2024.pdf	12 (1) 0.10 %
14	https://ir.kneu.edu.ua/bitstreams/57721939-fba5-4699-afde-625c26f42ccb/download	9 (1) 0.08 %
15	https://ir.kneu.edu.ua/bitstreams/c676f728-994c-4ac7-a47f-5c95e51267d8/download	5 (1) 0.04 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------

Рисунок Г.3 - Результат перевірки на плагіат