

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ПЛАТФОРМИ ДЛЯ ЗАМОВЛЕННЯ КОМП'ЮТЕРНИХ ІГОР
RESEARCH AND SOFTWARE IMPLEMENTATION OF
A PLATFORM FOR ORDERING COMPUTER GAMES

Виконав(ла): студент(ка) 2 курсу, групи КНТ-124м
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

КОЗИР М.І.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент ІЛЬЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КОЗИРА Миколи Ігоровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація платформи для замовлення комп'ютерних ігор. Research and Software Implementation of a Platform for Ordering Computer Games

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “30” вересня 2025 року № 447

2. Строк подання студентом проєкту (роботи) 02 грудня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 30 ” вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Микола КОЗИР
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Михайло КОЦУР
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
90 с., 3 табл., 36 рис., 3 дод., 14 джерел.

C#, VISUAL STUDIO, КОМП'ЮТЕРНА ГРА, МОВА
ПРОГРАМУВАННЯ, МОДЕЛЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для
замовлення комп'ютерних ігор.

Предмет дослідження – програмні засоби для замовлення
комп'ютерних ігор.

Мета роботи – дослідження та розробка програмного забезпечення для
замовлення комп'ютерних ігор.

Матеріали, методи та технічні засоби: мова програмування C#,
середовище розробки Visual Studio.

Результати. Розроблено програмне забезпечення для замовлення
комп'ютерних ігор за допомогою мови програмування C# та середовища
розробки Visual Studio.

Практична цінність роботи полягає у розробці програмного
забезпечення для замовлення комп'ютерних ігор.

Висновки. Виконано проєктування програмного забезпечення для
замовлення комп'ютерних ігор. Розроблено програмне забезпечення для
замовлення комп'ютерних ігор. Здійснено тестування розробленого
програмного забезпечення для замовлення комп'ютерних ігор.

Галузь використання – продаж комп'ютерних ігор.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 90 pages, 3 tables, 36 figures, 3 appendixes, 14 sources.

C#, VISUAL STUDIO, COMPUTER GAME, PROGRAMMING LANGUAGE, MODEL, SOFTWARE.

The object of research is the process of developing software for ordering computer games.

The subject of the research is software for ordering computer games.

The purpose of this work is to develop the software for ordering computer games.

Materials, methods and technical tools: C# programming language, Visual Studio development environment.

Results. The software tools for ordering computer games using the C# programming language and the Visual Studio development environment have been developed.

The practical value of the work lies in the development of software for ordering computer games.

Conclusions. The software for ordering computer games has been designed. The software for ordering computer games has been developed. The developed software for ordering computer games has been tested.

Scope of use – sale of computer games.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	12
1.1 Програмне забезпечення для замовлення комп'ютерних ігор	12
1.2 Аналіз програмного забезпечення для замовлення комп'ютерних ігор ..	15
1.3 Висновки за розділом 1	24
2 Матеріали і методи	26
2.1 Вибір мови програмування	26
2.2 Вибір середовища розробки для створення програмного забезпечення для замовлення комп'ютерних ігор	29
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор	31
2.4 Висновки за розділом 2	34
3 Опис програми	35
3.1 Структура програмного забезпечення для замовлення комп'ютерних ігор	35
3.2 Функціонування програмного забезпечення для замовлення комп'ютерних ігор	38
3.3 Розробка бази даних програмного забезпечення для замовлення комп'ютерних ігор	44
3.4 Проєктування інтерфейсу програмного забезпечення для замовлення комп'ютерних ігор	46
3.5 Висновки за розділом 3	50
4 Експлуатація, тестування та експериментальне дослідження програми	52
4.1 Призначення й умови застосування програми	52
4.2 Характеристики програми для замовлення комп'ютерних ігор	53
4.3 Інструкція по експлуатації програми	54
4.3.1 Звернення до програми	54

4.3.2 Вхідні й вихідні дані	54
4.3.3 Повідомлення.....	54
4.4 Виконання програмного забезпечення для замовлення комп'ютерних ігор.....	55
4.5 Тестування програмного забезпечення для замовлення комп'ютерних ігор.....	59
4.6 Висновки за розділом 4	59
Висновки	60
Перелік джерел посилання	63
Додаток А Технічне завдання	65
Додаток Б Фрагмент тексту програми	69
Додаток В Слайди презентації	83

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

DRM – Digital Rights Management;

БД – база даних;

ІТ – інформаційні технології;

КГ – комп’ютерна гра;

ПЗ – програмне забезпечення.

ВСТУП

На сучасному етапі розвитку цифрових технологій індустрія комп'ютерних ігор посідає провідне місце серед форм розваг і є одним із найбільш динамічно зростаючих сегментів ІТ-ринку. Зі зростанням попиту на відеоігри, виникає об'єктивна необхідність у розробці й удосконаленні програмного забезпечення, що забезпечує зручне, безпечне та швидке замовлення, придбання й управління ігровим контентом [1], [2].

Популярність цифрової дистрибуції спричинила майже повне витіснення фізичних носіїв із ринку. Користувачі очікують можливості миттєвого доступу до ігор після оплати, функцій автоматичного оновлення, захисту авторських прав, а також соціальних інтеграцій і рекомендацій на основі інтересів. Такі функції реалізуються через спеціалізоване програмне забезпечення – цифрові платформи, які стали стандартом у галузі. Розвиток хмарних технологій та поширення ігор за моделлю "ігри як сервіс" (Games-as-a-Service) ще більше підвищує актуальність таких рішень, адже користувачі очікують не лише покупки гри, а й постійної підтримки, оновлень, додаткового контенту та мультиплатформеного доступу. Це вимагає високої надійності, масштабованості та безпеки з боку ПЗ для замовлення ігор [2], [3].

Крім того, актуальність зростає з боку розробників і видавців, які зацікавлені в ефективному каналі дистрибуції, зборі аналітики, маркетингових інструментах і захисті ліцензійного контенту. Таке ПЗ дозволяє їм отримувати прямий зворотний зв'язок від гравців, формувати персоналізовані пропозиції та швидко реагувати на зміни в попиті [4]-[6].

Таким чином, програмне забезпечення для замовлення комп'ютерних ігор є не лише інструментом споживання цифрового контенту, але й критично важливим елементом цифрової екосистеми, що поєднує користувачів, розробників, видавців і постачальників технологій. Його подальший розвиток є актуальним і стратегічно важливим для стабільного зростання геймінгової індустрії [5], [6].

Проте численні переваги, програмне забезпечення для замовлення та придбання комп'ютерних ігор має низку суттєвих недоліків, які можуть впливати як на користувачів, так і на розробників. Одним із основних недоліків є наявність DRM-захисту (Digital Rights Management), який іноді занадто обмежує права споживача. Деякі системи захисту вимагають постійного онлайн контролю, можуть негативно впливати на продуктивність і навіть викликати конфлікти з апаратним або іншим програмним забезпеченням. Також варто відзначити монополізацію ринку окремими платформами, що призводить до обмеженого вибору, ексклюзивності контенту й інколи — до примусового використання кількох клієнтів одночасно, що ускладнює досвід користувача. Втрата доступу до контенту — ще один ризик. У разі видалення гри з платформи, зміни політики компанії або блокування облікового запису користувач фактично втрачає право на користування раніше придбаними іграми, оскільки більшість ПЗ працює за моделлю ліцензії, а не повного володіння продуктом. Також можливі проблеми із сумісністю або технічними помилками, особливо при оновленнях клієнтського ПЗ або при встановленні ігор, які більше не підтримуються. Це створює труднощі, з якими користувачі змушені самотійно справлятися.

Тому актуальною є розробка програмного забезпечення для замовлення комп'ютерних ігор. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки та дослідження програмного забезпечення для замовлення комп'ютерних ігор.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для замовлення комп'ютерних ігор;
- розробити модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор;
- здійснити проєктування програмного забезпечення для замовлення комп'ютерних ігор;

- створити програмне забезпечення для замовлення комп'ютерних ігор;
- виконати тестування розробленого програмного забезпечення для замовлення комп'ютерних ігор.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для замовлення комп'ютерних ігор

Розвиток ринку комп'ютерних ігор тісно пов'язаний із впровадженням спеціалізованого програмного забезпечення, що забезпечує цифрову доставку ігор без традиційних носіїв. З появою надійних Інтернет мереж гравці отримали можливість завантажувати розважальний контент безпосередньо на свої пристрої, що кардинально змінило колишню модель продажу та обміну дисками й картриджами [1].

Різні програми та сервіси дають змогу створювати цілі екосистеми, де користувачі реєструються, обирають і купують ігри, а розробники поширюють свої проєкти без посередників. Такі програмні інструменти об'єднують в одному середовищі каталог продуктів, систему захисту, оплату та управління бібліотекою придбань. Оновлення та додаткові пакети доступні безпосередньо через інтерфейс сервісу, що усуває необхідність вручну шукати й встановлювати файли [1], [2].

Сфера цифрової дистрибуції стала потужним інструментом для розширення ринку: покупець знаходить потрібну гру в кілька кліків, а розробник контролює умови розміщення та просування свого контенту. Середовище такого типу зазвичай має механізми для синхронізації з мобільними або настільними пристроями, що робить користування особливо зручним [1], [2].

Сучасний підхід до поширення ігор показує, як нові технології поступово заміщують фізичні носії, створюючи компактну, хмарну бібліотеку, доступну у будь-який момент. Усе це стало можливим завдяки комплексному програмному забезпеченню, яке інтегрує облік користувачів, обробку замовлень та багаторівневий захист контенту [1], [2].

У сфері поширення комп'ютерних ігор помітно утвердилася модель, коли доступ до розважального контенту забезпечується через спеціально розроблені програми та онлайн сервіси. Замість дисків чи картриджів

користувач отримує можливість миттєво завантажити гру на свій пристрій, користуючись інтерфейсом цифрової платформи. Такі сервіси охоплюють різні сегменти: для ігрових консолей створюються окремі онлайн магазини, що дають змогу придбати або оновити продукт, не виходячи з дому, а для мобільних пристроїв передбачені власні магазини додатків, де можна знайти ігри під будь-які уподобання [2], [3].

Комп'ютерний сегмент теж зберігає лідируючі позиції, адже спеціалізовані майданчики для ПК об'єднують величезні каталоги з інструментами для встановлення, зберігання та активації куплених проєктів. Програмне середовище такого типу зазвичай пропонує засоби управління бібліотекою, функції обліку, системи оновлення та механізми захисту від несанкціонованого копіювання [2], [3].

Поява розповсюдження ігор через Інтернет змінила ринок, зробивши покупку простою та швидкою справою, що не потребує додаткових фізичних витрат і не залежить від наявності на полиці магазину. Самі програми для розповсюдження виступають посередником між розробником і гравцем, забезпечуючи прозору комунікацію, безпечні платежі та підтримку користувачів. Такий підхід дає можливість розширювати вибір, легко знаходити новинки та підтримувати ігрову спільноту інтерактивними сервісами прямо всередині цифрового магазину [2], [3].

Програмне забезпечення, створене для замовлення та отримання комп'ютерних ігор, визначає сучасну форму поширення розважального контенту серед мільйонів гравців. Замість дисків і картриджів користувач отримує миттєвий доступ до ігрових бібліотек прямо зі свого пристрою завдяки централізованим сервісам, що поєднують у собі каталог, платіжні системи та механізми захисту даних [2], [3].

Різноманіття платформ відповідає на попит власників консолей, смартфонів чи настільних комп'ютерів. Окремі рішення забезпечують можливість вибору й придбання ігор для популярних ігрових систем, інші спеціалізуються на мобільних додатках, а окремі сервіси пропонують моделі

для тих, хто прагне повної свободи у використанні контенту. Такі інструменти стали звичним посередником між розробниками й кінцевим споживачем, надаючи всім учасникам процесу простий спосіб публікувати, шукати та завантажувати ігри в будь-який момент [2], [3].

Серед популярних рішень виділяються сервіси, які поєднують спільноту гравців, автоматичні оновлення та гнучкі умови купівлі. Зручність і швидкість підключення до каталогу ігор спрощують доступ, а регулярні акції та розпродажі роблять покупку ще привабливішою. Особливо це помітно на персональних комп'ютерах, де великі майданчики об'єднують мільйони профілів і тисячі проєктів різного жанру [3], [4].

У результаті цифровий спосіб поширення комп'ютерних ігор закріпився як основний, витіснивши потребу у фізичних носіях. Завдяки цьому гравець не обмежений місцем чи часом, а його ігровий простір існує всередині зручного інтерфейсу, що завжди під рукою. Платформи такого типу продовжують розвиватися, додаючи все нові функції та покращуючи користувацький досвід, перетворюючи звичайну покупку гри на повноцінну інтерактивну взаємодію [2], [3].

Ринок комп'ютерних ігор давно переорієнтувався на моделі, де покупець взаємодіє з грою через спеціальні програми чи онлайн платформи. Замість того, щоб шукати диски чи картриджі, користувачі тепер мають у розпорядженні віртуальні вітрини, що відкривають доступ до великих колекцій ігор для будь-яких інтересів. Через зручний інтерфейс можна обрати новинку чи рідкісний проєкт, завантажити його на свій пристрій і відразу розпочати гру без зайвих затримок [3], [4].

Програмні інструменти для дистрибуції ігор об'єднують безліч корисних можливостей: від перегляду описів і оглядів до управління особистою бібліотекою і отримання оновлень. Завдяки такому підходу купівля ігор більше не залежить від фізичних носіїв і полиць магазинів. Розробники за допомогою цих рішень можуть донести свій продукт до гравця швидко і без

посередників, а користувач завжди має у своєму розпорядженні великий вибір контенту [3], [4].

Завдяки високій конкуренції подібні сервіси розширюють умови для покупки ігор: додаються знижки, акції, бонуси чи безкоштовні роздачі, що приваблюють аудиторію та тримають інтерес до нових проєктів. Усе це працює завдяки програмному забезпеченню, що синхронізує каталоги, систему оплати, захист даних та підтримку користувача. Тепер ігрові колекції більше не займають місце вдома, адже вся бібліотека існує у цифровому вигляді й доступна в кілька натискань миші [3], [4].

Сьогодні цифровий спосіб придбання комп'ютерних ігор сприймається як звична частина індустрії, а інтерфейс спеціалізованих платформ робить процес легким і швидким. Користувачеві залишається лише обрати бажаний продукт, а все інше — від безпеки до доставки контенту — забезпечує надійне програмне середовище [3], [4].

Визначено, що програмне забезпечення для замовлення комп'ютерних ігор надає можливість завантаження розважального ігрового контенту безпосередньо на свої пристрої користувача. Різні програми та сервіси для замовлення комп'ютерних ігор дають змогу створювати цілі екосистеми, де користувачі реєструються, обирають і купують ігри, а розробники поширюють свої проєкти без посередників. Такі програмні інструменти об'єднують в одному середовищі каталог продуктів, систему захисту, оплату та управління бібліотекою придбань.

1.2 Аналіз програмного забезпечення для замовлення комп'ютерних ігор

Проаналізуємо програмне забезпечення для замовлення комп'ютерних ігор [5]-[10].

Програмне забезпечення Humble Store (рис. 1.1) працює як онлайн простір, де можна придбати цифровий контент, перш за все ігри для

персональних комп'ютерів. Головна ідея цього ресурсу полягає у поєднанні вигідних пропозицій для гравців із підтримкою благодійності, що вирізняє його серед інших подібних сервісів. Окрім окремих ігор, на платформі представлено спеціальні збірки, які формуються за спільною тематикою і продаються за вигідними умовами [7], [8].

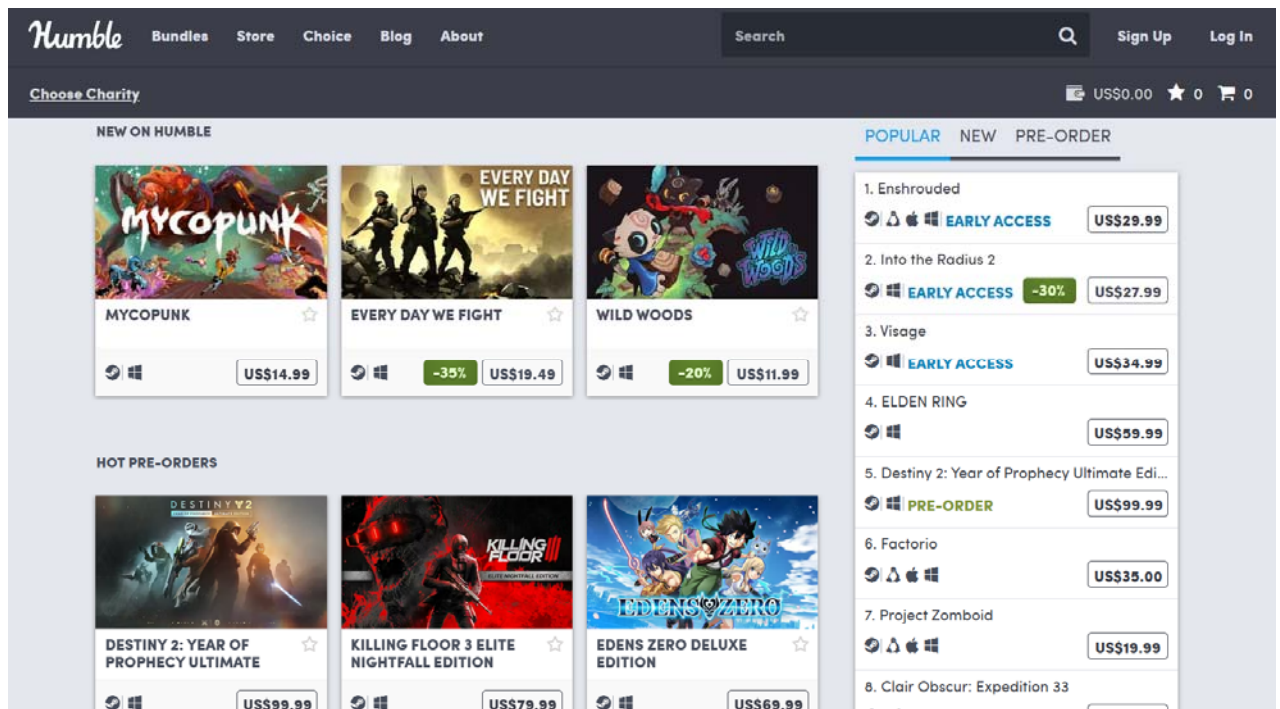


Рисунок 1.1 – Програмне забезпечення Humble Store [8]

Особливість програмного забезпечення Humble Store полягає у тому, що багато придбаних продуктів можна активувати за допомогою ключів у популярних клієнтах дистрибуції ігор, що полегшує доступ до бібліотеки. Деякі матеріали доступні для прямого завантаження, що дозволяє уникнути проміжних ланок [7], [8].

Програмна частина Humble Store реалізована так, щоб користувачеві було зручно орієнтуватися у розділах, швидко оформлювати замовлення та керувати підписками. Крім ігор, покупцеві пропонуються різноманітні книги та музичні матеріали, що відкриває більше можливостей для тих, хто цікавиться не лише ігровим контентом [7], [8].

В окремій частині платформи зберігається колекція проєктів, які залишаються у доступі навіть після припинення підписки. Це робить використання сервісу привабливим для тих, хто цінує можливість поповнити власну колекцію ігор чи книг за гнучкими умовами. Усе це підтримується стабільною архітектурою сайту, що забезпечує безперебійну роботу з каталогом і надає клієнтам легкий і безпечний спосіб отримувати цифровий контент [7], [8].

Основними характеристиками програмного забезпечення Humble Store є такі [7], [8]:

- гнучка система пакетів: користувачі програмного забезпечення Humble Store можуть обирати тематичні добірки ігор або програм, формуючи власну колекцію за вигідною ціною [7], [8];

- інтеграція з іншими сервісами: більшість придбаних ігор у програмному забезпеченні Humble Store активуються через ключі у популярних клієнтах цифрової дистрибуції [7], [8];

- доступ до контенту: частина бібліотеки програмного забезпечення Humble Store складається з ігор та книг, які можна зберігати без обмежень навіть після завершення підписки [7], [8];

- підтримка благодійності: частина виторгу програмного забезпечення Humble Store спрямовується на доброчинність, що залучає користувачів, які хочуть поєднати купівлю з добром [7], [8];

- простий інтерфейс: навігація сайтом програмного забезпечення Humble Store і оформлення покупок реалізовані максимально зрозуміло для будь-якого відвідувача [7], [8];

- різноманітність цифрових товарів: окрім ігор за допомогою програмного забезпечення Humble Store можна придбати електронні книги, музику та інші медіафайли [7], [8];

- постійні розпродажі: регулярні акції дозволяють заощаджувати при формуванні власної бібліотеки [7], [8];

– зручне керування підписками: користувачі програмного забезпечення Humble Store мають можливість швидко оновлювати або скасовувати підписку без втрати доступу до придбаного контенту [7], [8].

Серед недоліків програмного забезпечення Humble Store можна відзначити залежність від сторонніх платформ для активації більшості ігор, що іноді ускладнює управління великою бібліотекою. Також трапляються випадки, коли деякі пропозиції можуть бути недоступними для певних регіонів або обмеженими за часом, що вимагає від користувача постійного відстеження акцій. Крім того, іноді виникають технічні затримки з надсиланням ключів чи оновленням інформації про замовлення, що може вплинути на загальний досвід використання сервісу [7], [8].

Програмне забезпечення Itch (рис. 1.2) пропонує онлайн простір, де кожен охочий може знайти ігри незалежних творців, отримати до них доступ або навіть поширити власні проєкти. Головна мета цього програмного сервісу полягає у тому, щоб забезпечити максимально вільне середовище для розробників, які прагнуть експериментувати з ідеями без обмежень з боку великих видавців [7], [9].

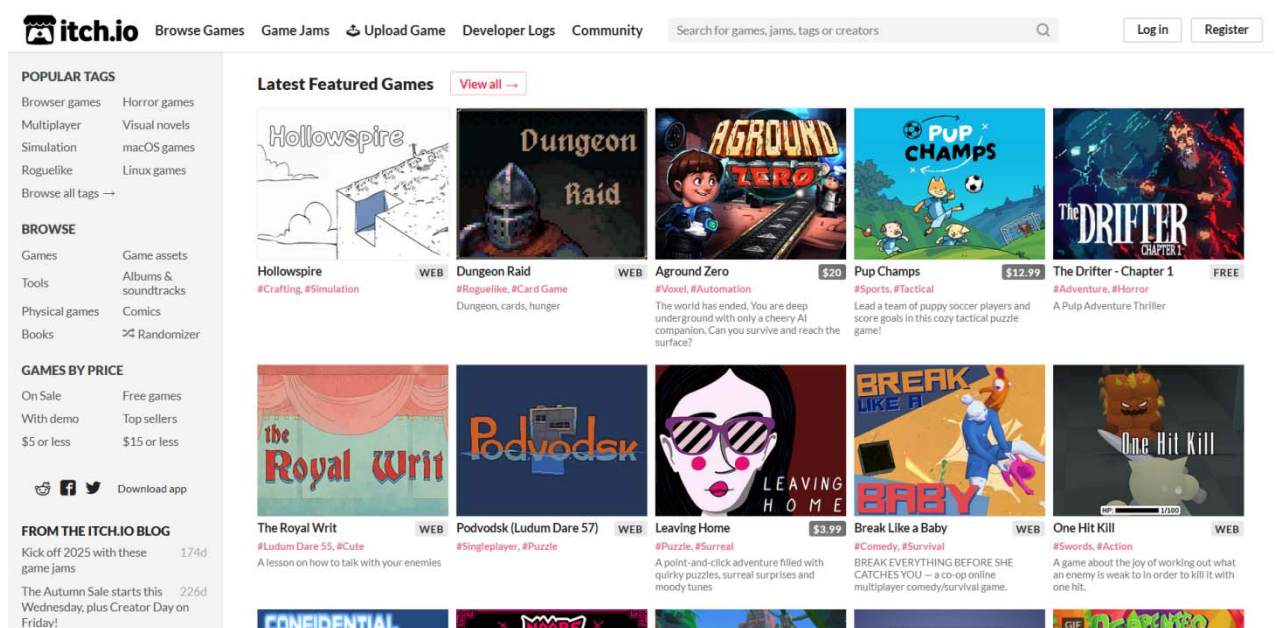


Рисунок 1.2 – Програмне забезпечення Itch [9]

Тут представлено величезну кількість проєктів, серед яких трапляються унікальні приклади ігор, що згодом набувають популярності на ширших ринках. Сам сайт програмного забезпечення Itch побудовано так, щоб користувач міг легко орієнтуватися у великій кількості ігор. Для цього працюють фільтри, що дозволяють вибирати проєкти за популярністю чи жанровими ознаками. Серед пропозицій багато безкоштовних матеріалів, проте частина контенту доступна за добровільну чи символічну оплат [7], [9].

Технічна структура ресурсу програмного забезпечення Itch розрахована на те, щоб як гравці, так і розробники мали зручні інструменти для завантаження, демонстрації та продажу своїх робіт. Цифрова платформа підтримує гнучкі механізми оплати, завантаження та обміну файлами, що спрощує взаємодію між творцями і споживачами контенту. Itch залишається прикладом того, як програмне середовище здатне дати простір ідеям, що не завжди вписуються у стандартні рамки індустрії, і допомогти знайти прихильників навіть найсміливішим експериментам [7], [9].

Програмне забезпечення Itch має такі особливості [7], [9]:

- фокус на інді-розробках: платформа Itch орієнтована на підтримку незалежних творців, які можуть публікувати свої ігри без обмежень [7], [9];
- велика бібліотека контенту: користувачі програмного забезпечення Itch отримують доступ до тисяч унікальних ігор, серед яких є багато експериментальних проєктів [7], [9];
- гнучка модель оплати: частина ігор програмного забезпечення Itch розповсюджується безкоштовно, а інші доступні за символічну суму або за моделлю «плати скільки хочеш» [7], [9];
- зручна система пошуку: фільтрація ігор у програмному забезпеченні Itch здійснена за популярністю, жанром чи тегами дозволяє швидко знайти цікаві варіанти [7], [9];
- проста взаємодія для розробників: у програмному забезпеченні Itch надається інструментарій для завантаження, оновлення та продажу проєктів без складних процедур [7], [9];

- підтримка експериментального контенту: програмного забезпечення Itch стимулює створення нестандартних ігрових рішень, які рідко з'являються у великих магазинах [7], [9];
- прямі завантаження: користувачі програмного забезпечення Itch можуть отримати комп'ютерну гру безпосередньо, обходячи сторонні сервіси активації [7], [9];
- активна спільнота: програмне забезпечення Itch підтримує атмосферу взаємодії між творцями та гравцями комп'ютерної гри через коментарі, блоги та спільні заходи [7], [9].

Серед недоліків програмного забезпечення Itch можна відзначити складність вивчення, оскільки для нових користувачів може знадобитися певний час на вивчення інтерфейсу та функціоналу Blackboard Learn, особливо якщо вони раніше не користувалися подібними системами. Для деяких великих освітніх закладів або організацій, що обробляють великі обсяги даних, можуть виникати обмеження щодо зберігання та обробки інформації. Іншим недоліком є необхідність оплати за деякі функції [9], [13], що може викликати обмеження використання цієї програми невеликими освітніми установами або організаціями з обмеженим бюджетом [7], [9].

Програмне забезпечення Fanatical (рис. 1.3) працює як сервіс для придбання цифрових ігор та супутніх матеріалів, що поєднує зручність онлайн магазину з можливістю економити завдяки вигідним пропозиціям. Основна ідея полягає у продажу великих добірок ігор за привабливими цінами, що дозволяє користувачам суттєво поповнювати власну бібліотеку без значних витрат. Крім повноцінних колекцій, платформа пропонує окремі проекти з відчутною економією у порівнянні зі звичайною вартістю. Особливість реалізації програмного забезпечення Fanatical полягає у тому, що після покупки користувач отримує ключ активації, який можна застосувати у відповідних сервісах дистрибуції. Окремі позиції доступні і без захисту, що спрощує використання придбаного контенту [7], [10].

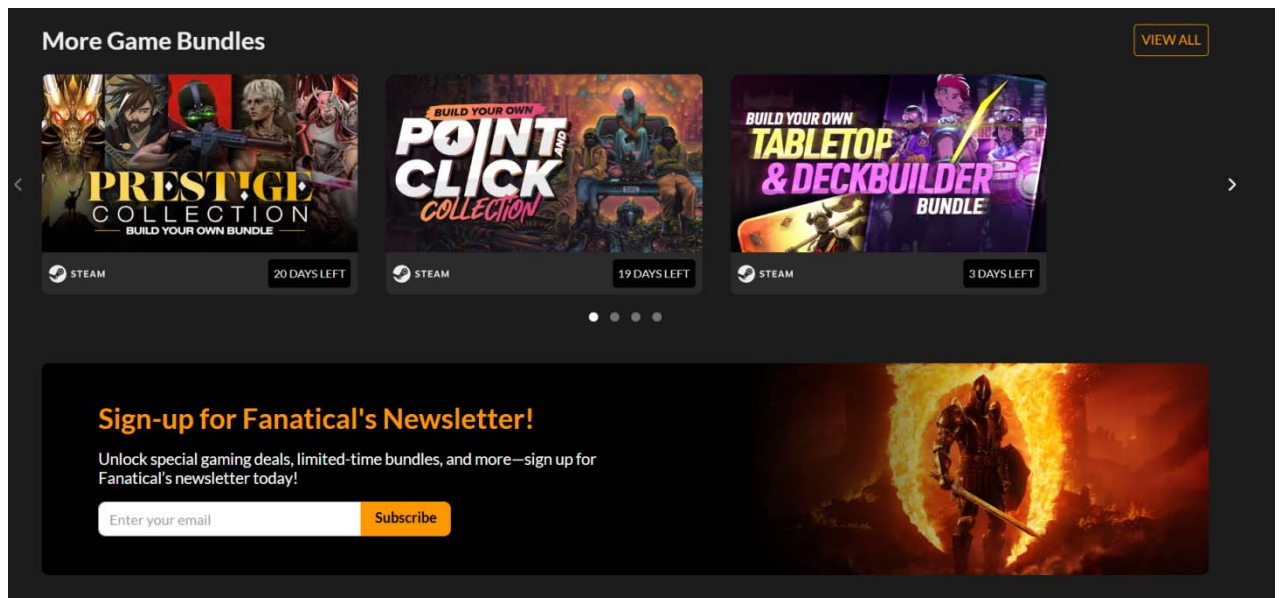


Рисунок 1.3 – Програмне забезпечення Fanatical [10]

Програмне забезпечення Fanatical також доповнює асортимент навчальними матеріалами та книгами, що розширює функціонал платформи і робить її корисною не лише для гравців, а й для тих, хто шукає цифровий контент у суміжних сферах. Спеціальні акційні пропозиції формують додаткову мотивацію користуватися цим сервісом, адже можна знайти цікаві ігри за символічною ціною [7], [10].

Програмне забезпечення Fanatical має такі особливості [7], [10]:

- видача ключів: після покупки користувачі програмного забезпечення Fanatical отримують офіційні ключі активації для застосування у популярних сервісах [7], [10];
- вигідні ігрові збірки: пропонується можливість придбати великі пакети ігор за низькою ціною, що дозволяє значно заощадити [7], [10];
- акційні пропозиції: на платформі програмного забезпечення Fanatical регулярно з'являються спеціальні пропозиції з суттєвими знижками на окремі позиції [7], [10];
- ігри без Digital Rights Management (DRM): частина контенту може бути завантажена без обмежень DRM, що спрощує використання програмного забезпечення Fanatical [7], [10];

- широкий асортимент: крім ігор, можна знайти електронні книги, посібники та навчальні матеріали різних тематик [7], [10];
- підтримка бонусних пропозицій: деякі покупки супроводжуються додатковими вигодами у вигляді спеціальних пропозицій [7], [10];
- простота користування: інтерфейс магазину програмного забезпечення Fanatical побудований зрозуміло, що полегшує процес пошуку та купівлі цифрового контенту [7], [10].

Серед недоліків програмного забезпечення Fanatical можна відзначити обмежений вибір новітніх релізів, адже більшість знижок стосуються менш популярних або вже добре відомих ігор. Деяких користувачів може насторожувати необхідність активувати більшість куплених продуктів у сторонніх клієнтах, що не завжди зручно для тих, хто прагне зберігати всю бібліотеку в одному місці. Крім того, наявність DRM у багатьох позиціях іноді обмежує свободу використання придбаного контенту за межами офіційних сервісів [7], [10].

Порівняльну характеристику програмного забезпечення для замовлення комп'ютерних ігор наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для замовлення комп'ютерних ігор

Критерій порівняння	Humble Store [8]	Itch [9]	Fanatical [10]
Великий вибір комп'ютерних ігор	+	+-	+
Наявність ігор без DRM	+-	+	+-
Зручність інтерфейсу	+	+-	+-
Рівень популярності	+	+-	+-
Офіційні ключі до ігор	+-	+-	+
Різні варіанти підписки	+	+-	-

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для замовлення комп'ютерних ігор. Проте деякі програмні засоби для замовлення комп'ютерних ігор можуть мати модулі захисту, які вимагають постійного онлайн контролю, можуть негативно впливати на продуктивність і навіть викликати конфлікти з апаратним або іншим програмним забезпеченням. Також варто відзначити монополізацію ринку окремими платформами, що призводить до обмеженого вибору, ексклюзивності контенту й інколи — до примусового використання кількох клієнтів одночасно, що ускладнює досвід користувача. Втрата доступу до контенту — ще один ризик. У разі видалення гри з платформи, зміни політики компанії або блокування облікового запису користувач фактично втрачає право на користування раніше придбаними іграми, оскільки більшість ПЗ працює за моделлю ліцензії, а не повного володіння продуктом. Також можливі проблеми із сумісністю або технічними помилками, особливо при оновленнях клієнтського ПЗ або при встановленні ігор, які більше не підтримуються. Це створює труднощі, з якими користувачі змушені самостійно справлятися. Тому актуальною є розробка програмного забезпечення для замовлення комп'ютерних ігор.

При розробці програмного забезпечення для замовлення комп'ютерних ігор необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення переліку комп'ютерних ігор, доступних для завантаження;
- підтримка можливості групування комп'ютерних ігор за обраною користувачем категорією (наприклад, ігри з пошуком предметів, ігри-ребуси, аркадні ігри і т.ін.);
- можливість відображення категорій комп'ютерних ігор;
- можливість авторизації;
- підтримка можливості обчислення та виведення користувачу інформації про кількість комп'ютерних ігор у відповідній категорії;

- підтримка можливості подання інформації про обрану комп'ютерну гру;
- можливість завантаження обраної комп'ютерної гри.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для замовлення комп'ютерних ігор надає можливість завантаження розважального ігрового контенту безпосередньо на свої пристрої користувача. Різні програми та сервіси для замовлення комп'ютерних ігор дають змогу створювати цілі екосистеми, де користувачі реєструються, обирають і купують ігри, а розробники поширюють свої проєкти без посередників. Такі програмні інструменти об'єднують в одному середовищі каталог продуктів, систему захисту, оплату та управління бібліотекою придбань.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для замовлення комп'ютерних ігор. Проте деякі програмні засоби для замовлення комп'ютерних ігор можуть мати модулі захисту, які вимагають постійного онлайн контролю, можуть негативно впливати на продуктивність і навіть викликати конфлікти з апаратним або іншим програмним забезпеченням. Також варто відзначити монополізацію ринку окремими платформами, що призводить до обмеженого вибору, ексклюзивності контенту й інколи — до примусового використання кількох клієнтів одночасно, що ускладнює досвід користувача. Втрата доступу до контенту — ще один ризик. У разі видалення гри з платформи, зміни політики компанії або блокування облікового запису користувач фактично втрачає право на користування раніше придбаними іграми, оскільки більшість ПЗ працює за моделлю ліцензії, а не повного володіння продуктом. Також можливі проблеми із сумісністю або технічними помилками, особливо при оновленнях клієнтського ПЗ або при встановленні ігор, які більше не підтримуються. Це створює труднощі, з якими користувачі змушені

самостійно справлятися. Тому актуальною є розробка програмного забезпечення для замовлення комп'ютерних ігор.

Сформульовано функціональні вимоги до програмного забезпечення для замовлення комп'ютерних ігор.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для замовлення комп'ютерних ігор було використано мову програмування C# [11], [12].

C# – вирізняється здатністю легко пристосовуватися до широкого спектра програмних завдань, що забезпечує їй стійке місце серед основних інструментів розробника. Завдяки поєднанню строгої системи типів із сучасною об'єктною моделлю, мова програмування C# дозволяє формувати чітку внутрішню структуру складних програм, підтримувати взаємозв'язок між компонентами та забезпечувати масштабованість рішень без шкоди для їхньої цілісності [11], [12].

Функціонування мови програмування C# ґрунтується на платформі .NET, яка створює сприятливі умови для реалізації проектів, незалежно від цільового середовища чи операційної системи. Завдяки підтримці розподілених технологій, асинхронного виконання завдань і модульного підходу до організації коду, відкривається можливість розробляти як локальні програми, так і вебсервіси, серверну логіку, хмарні застосунки або інструменти електронної торгівлі [11], [12].

Сучасні засоби, зокрема .NET MAUI, сприяють виходу C# за межі типової асоціації з платформою Windows. Цей інструментарій дає змогу працювати на різних пристроях, включно з мобільними телефонами, планшетами, настільними системами й навіть вбудованими мікроконтролерами. Завдяки впорядкованому синтаксису, чіткій логіці та багатству доступних бібліотек, мова стає однаково зручною як для початківців, так і для досвідчених спеціалістів [11], [12].

Окреме значення надається підтримці якісного кодування — засобам перевірки, абстракції через інтерфейси та шаблонів побудови. Такі особливості дозволяють створювати рішення, орієнтовані на тривале використання, гнучке вдосконалення та безперервну підтримку. Саме це

поєднання технологічної зрілості та прикладної практичності робить мову програмування C# надзвичайно ефективним інструментом у сучасному програмному середовищі [11], [12].

Мова C# вирізняється як ефективне технологічне рішення у сфері створення програмного забезпечення, зокрема тоді, коли йдеться про реалізацію функціоналу цифрового замовлення та придбання комп'ютерних ігор. В основі її привабливості — гармонійне поєднання технічної глибини й зручності розробки в рамках середовища .NET, яке підтримує як різні типи проєктів. Цей технологічний стек дозволяє реалізовувати інтерфейс користувача у вигляді сайту або окремого застосунку, залежно від цілей і завдань, що стоять перед розробником [11], [12].

У контексті роботи з даними C# забезпечує ефективну взаємодію з базами даних різного типу, дозволяючи зберігати, обробляти та аналізувати інформацію про акаунти, замовлення та ігрові бібліотеки. Особливу увагу приділено безпеці — у проєктах, де важлива надійна обробка персональної та фінансової інформації, можливості .NET щодо шифрування, автентифікації й контролю доступу відіграють ключову роль [11], [12].

Із розширенням функціоналу .NET платформа C# перестала бути суто Windows-орієнтованою. Завдяки підтримці кросплатформенності з'явилась можливість охоплювати мобільні пристрої, macOS, Linux та інші системи. Це відкриває шлях до створення мультиплатформеного програмного забезпечення, що є актуальним у разі потреби доступу до цифрової бібліотеки з різних пристроїв [11], [12].

C# також тісно пов'язана з ігровою індустрією — її використання в Unity дозволяє реалізовувати моделі, в яких платформа продажів і сама гра технічно взаємодіють. Завантаження, оновлення, статистика — усе це можливо завдяки тому, що середовище та мова розробки об'єднуються в єдину логіку обробки користувацького досвіду [11], [12].

Стабільність розвитку мови, офіційна підтримка від Microsoft і активна спільнота роблять її інструментом, що здатен забезпечити швидке

розгортання, тривалу підтримку та впевнене масштабування. Завдяки компіляції в машинний код вдається досягти високої швидкодії програм, а сучасна архітектура дозволяє підтримувати структурованість і керованість проєкту [11], [12].

З урахуванням усіх технічних і прикладних характеристик, мова програмування C# логічно постає як універсальний вибір для реалізації повноцінної, захищеної та зручно масштабованої платформи для електронного продажу ігрового контенту [11], [12].

Обґрунтування вибору мови програмування для реалізації програмного забезпечення для замовлення комп'ютерних ігор наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для реалізації програмного забезпечення для замовлення комп'ютерних ігор

Критерій порівняння мов програмування	Мова програмування		
	C#	Python	JavaScript
Підтримка процесу розробки застосунків різних типів	+	+	—
Висока швидкодії	+	+-	—
Інтеграція з базами даних	+	+	+-
Зручність для створення ПЗ для замовлення комп'ютерних ігор	+	—	—
Можливість інтеграції з ігровими рушіями	+	+-	—
Наявність офіційних бібліотек та розширень	+	+-	+-

Отже, для реалізації програмного забезпечення для замовлення комп'ютерних ігор обрано мову програмування C#, яка забезпечує можливість створення кросплатформених рішень, інтеграцію з базами даних, а також

надає механізми безпеки. Крім того, мова програмування C# підтримує зручність розробки та високу продуктивність, а також має активну спільноту та підтримку.

2.2 Вибір середовища розробки для створення програмного забезпечення для замовлення комп'ютерних ігор

В якості середовища розробки для створення програмного забезпечення для замовлення комп'ютерних ігор було обрано середовище Visual Studio [13], [14].

Середовище Visual Studio створене як комплексне рішення для розробки програмного забезпечення, де реалізовано не лише базові інструменти програмування, а й засоби для ефективної організації всього життєвого циклу проєкту. Завдяки щільному поєднанню з технологічною екосистемою Microsoft відкривається можливість проєктування складних систем без залучення додаткових зовнішніх засобів. Усе, що необхідно для розробника, зосереджено в одному інтерфейсі, де реалізовано механізми автодоповнення, перевірки коду в реальному часі, налаштування шаблонів, а також створення і перевірки середовищ виконання без потреби в ручному розгортанні [13], [14].

Система підтримує паралельне використання кількох мов програмування в межах одного рішення, що особливо актуально при створенні багаторівневих або модульних застосунків із клієнтською, серверною і мобільною складовими. Наявність інтерактивних редакторів інтерфейсу, інструментів для проєктування баз даних і API дозволяє мінімізувати повторювані операції та зосередити увагу на функціональній частині застосунку [13], [14].

Завдяки динамічному розвитку, тісній інтеграції з сервісами Microsoft і постійній підтримці користувачької спільноти, середовище розробки Visual Studio зберігає свою актуальність як інструмент, що не лише відповідає

вимогам сучасної розробки, а й задає напрямок технологічного прогресу. У роботі з цим середовищем процес створення ПЗ перетворюється на послідовне, кероване середовище, де кожен етап — від написання коду до тестування й публікації — логічно вписаний у загальну архітектуру циклу розробки [13], [14].

Використання Visual Studio у процесі розробки програмного забезпечення для систем електронного замовлення ігор визначається високим рівнем інтеграції з платформою .NET і ефективною взаємодією з мовою C#. Це середовище надає доступ до широкого набору інструментів, які дозволяють уникнути зайвих маніпуляцій із налаштуванням і зосередитися безпосередньо на розробці функціоналу. Завдяки продуманому інтерфейсу та широким можливостям автоматизації, забезпечується плавний перехід від початкової ідеї до робочого рішення, без необхідності витрачати час на усунення другорядних технічних перешкод [13], [14].

Середовище розробки Visual Studio оптимізовано для створення програм, що мають працювати на різних платформах і пристроях, включно з мобільними клієнтами, десктопними застосунками та вебінтерфейсами. Його синхронізація з хмарною інфраструктурою, інструментами автоматизованої доставки, тестування та відстеження змін забезпечує можливість повного контролю над усім процесом — від написання коду до запуску й подальшого обслуговування [13], [14].

Платформа демонструє здатність поєднувати стабільність роботи з технічною потужністю, що створює сприятливе середовище для побудови надійного і гнучкого програмного продукту, придатного до розширення та підтримки. Усе це формує підґрунтя для виваженого вибору Visual Studio як основного середовища розробки у сфері цифрової торгівлі іграми.

Обґрунтування вибору середовища розробки для створення програмного забезпечення для замовлення комп'ютерних ігор наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для замовлення комп'ютерних ігор

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	MonoDevelop	JetBrains Rider
Повноцінна підтримка C# та .NET	+	+-	+
Зручність створення інтерфейсу користувача	+	-	+-
Підтримка інструментів для тестування	+	+-	+
Відлагодження та профілювання	+	+-	+
Швидкість запуску та стабільність	+	+-	+-

Таким чином, для створення програмного забезпечення для замовлення комп'ютерних ігор обрано середовище розробки Visual Studio, що поєднує в собі інтелектуальні підказки, засоби рефакторингу, вбудовану підтримку систем контролю версій, а також зручний інтерфейс для роботи з базами даних, програмними сервісами та користувацьким інтерфейсом. Це середовище створене так, щоб розробник міг швидко переходити від ідеї до реалізації без зайвих технічних бар'єрів.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор

З метою можливості розробки програмного забезпечення для замовлення комп'ютерних ігор створимо відповідну модель у вигляді діаграми

прецедентів. При розробленні моделі взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор враховуємо те, що користувачі програми мають можливості пошуку потрібних комп'ютерних ігор, зокрема, виконувати їх сортування (фільтрацію за заданими критеріями), авторизуватися та оформлювати замовлення товару (комп'ютерної гри), адміністратори мають можливості роботи з адміністративною панеллю, зокрема, можливість оновлення товарів.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для замовлення комп'ютерних ігор можна описати таким чином:

- користувач програмного забезпечення для замовлення комп'ютерних ігор авторизується;
- після авторизації користувача сервер програмного забезпечення для замовлення комп'ютерних ігор у відповідь повертає основну сторінку вебсайту для замовлення комп'ютерних ігор;
- користувач бачить відповідну сторінку програмного забезпечення для замовлення комп'ютерних ігор та шукає потрібну комп'ютерну гру, наприклад, за назвою, або виконує сортування ігор для подальшого пошуку;
- користувач за необхідності читає опис відповідної комп'ютерної гри;
- користувач натискає відповідну кнопку для оформлення замовлення на комп'ютерну гру;
- користувач завершує роботу з програмним забезпеченням для замовлення комп'ютерних ігор.

Враховуючи сценарії роботи користувачів програмного забезпечення для замовлення комп'ютерних ігор (користувач, відвідувач та адміністратор) при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

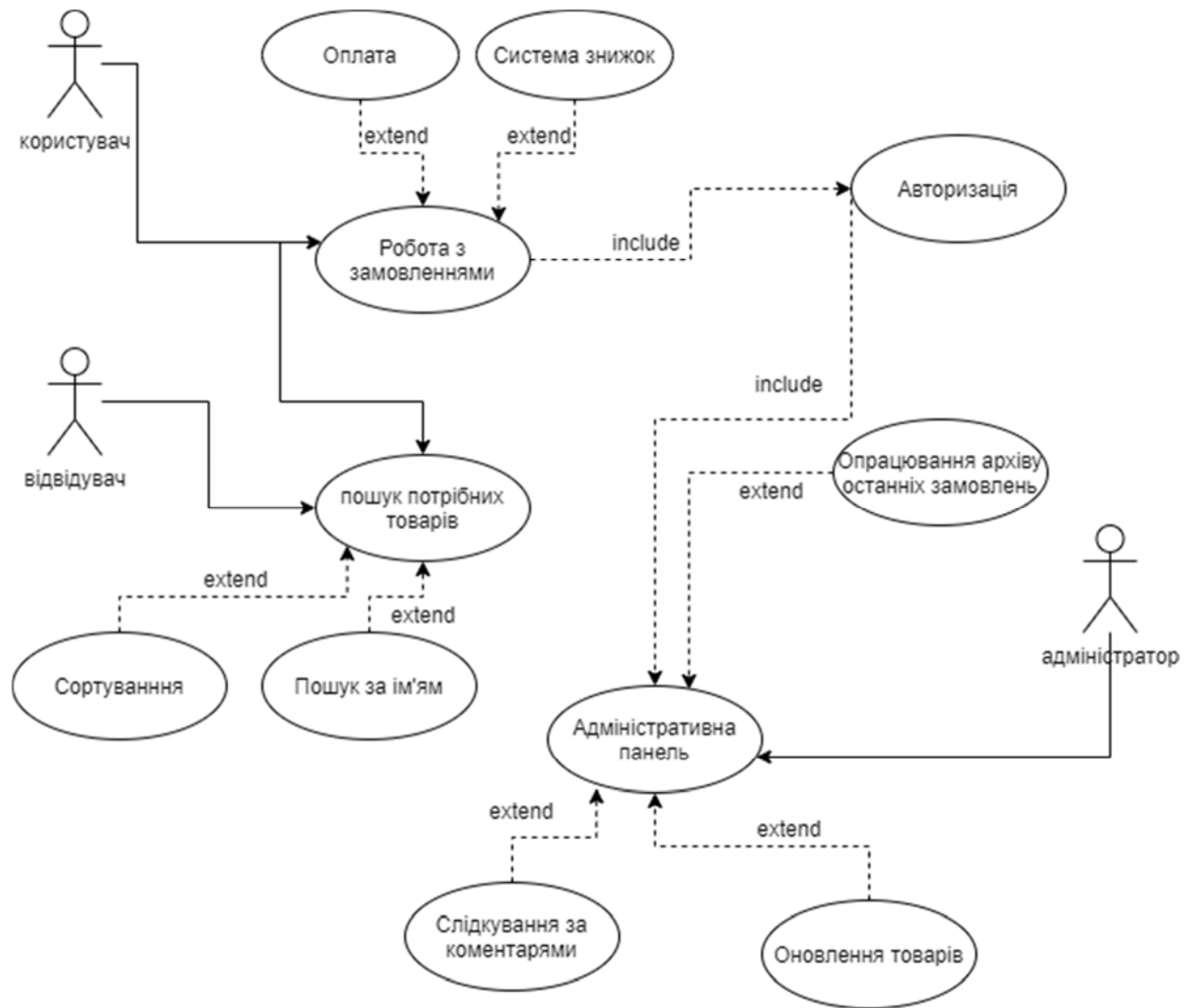


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор

Опишемо основні дії адміністратора програмного забезпечення для замовлення комп'ютерних ігор:

- адміністратор програмного забезпечення для замовлення комп'ютерних ігор авторизується;
- після авторизації адміністратора сервер програмного забезпечення для замовлення комп'ютерних ігор у відповідь на запит повертає основну сторінку для замовлення комп'ютерних ігор;
- адміністратор програмного забезпечення для замовлення комп'ютерних ігор бачить відповідну сторінку та переходить в адміністративну панель;

- адміністратор програмного забезпечення для замовлення комп'ютерних ігор переглядає обрану інформацію та за необхідності оновлює інформацію про комп'ютерні ігри;

- адміністратор програмного забезпечення для замовлення комп'ютерних ігор завершує роботу з програмою.

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для замовлення комп'ютерних ігор.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для замовлення комп'ютерних ігор обрано мову програмування C#, яка забезпечує можливість створення кросплатформених рішень, інтеграцію з базами даних, а також надає механізми безпеки. Крім того, мова C# підтримує зручність розробки та високу продуктивність, а також має активну спільноту та підтримку.

Для створення програмного забезпечення для замовлення комп'ютерних ігор обрано середовище розробки Visual Studio, що поєднує в собі інтелектуальні підказки, засоби рефакторингу, вбудовану підтримку систем контролю версій, а також зручний інтерфейс для роботи з базами даних, програмними сервісами та користувацьким інтерфейсом. Це середовище створене так, щоб розробник міг швидко переходити від ідеї до реалізації без зайвих технічних бар'єрів.

Створено модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для замовлення комп'ютерних ігор.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для замовлення комп'ютерних ігор

Структуру програмної платформи для замовлення комп'ютерних ігор у вигляді діаграми класів наведено на рис. 3.1.

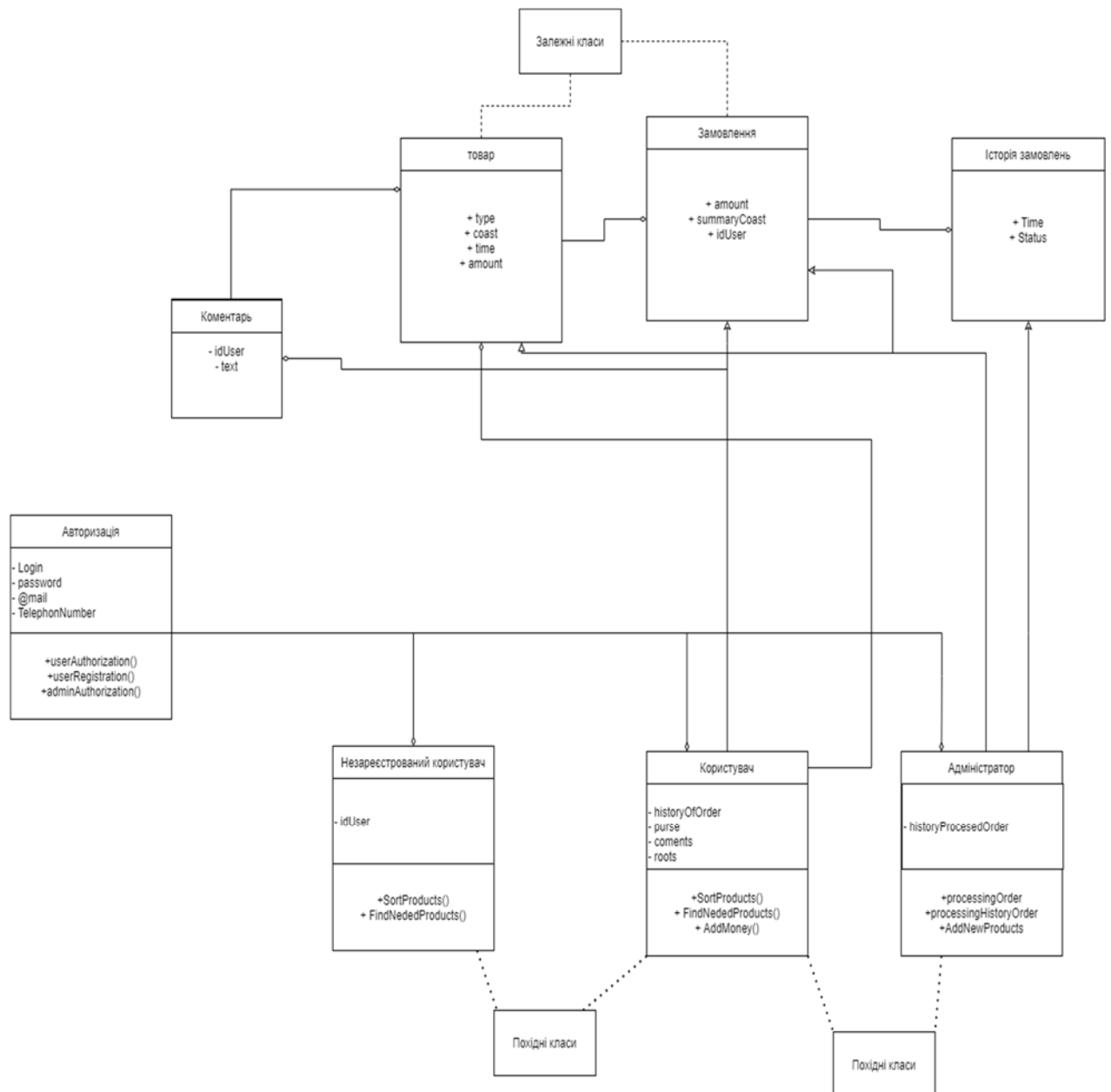


Рисунок 3.1 – Структура програмного забезпечення для замовлення комп'ютерних ігор

Як видно з рис. 3.1, основними класами програмної платформи для замовлення комп'ютерних ігор Користувач, Незареєстрований користувач, Адміністратор, Товар, Замовлення, Коментар, Історія замовлень та інші.

Діаграма класів програмної платформи для замовлення комп'ютерних ігор відображає структуру основних об'єктів системи, їхні характеристики та взаємозв'язки між собою. В її основі лежить клас Користувач, який виступає узагальненням для інших типів користувачів. Від нього наслідуються класи Незареєстрований користувач і Адміністратор, кожен з яких розширює базову модель специфічними можливостями — наприклад, адміністратор має доступ до редагування товарів і модерації коментарів, тоді як незареєстрований користувач обмежується переглядом каталогу ігор.

Клас Товар моделює окрему гру або пакет ігор, що доступні для придбання. У ньому зберігається інформація про назву, жанр, ціну, рейтинг, системні вимоги та інші метадані. Об'єкти цього класу пов'язані з класом Коментар, який фіксує відгуки користувачів. Кожен коментар містить текст, оцінку, дату публікації та посилання на користувача, який його залишив.

Замовлення моделюється через окремий клас Замовлення, який містить перелік товарів, дату покупки, стан оплати та спосіб доставки (наприклад, ключ активації або пряме завантаження). У зв'язку з цим класом стоїть Історія замовлень, яка слугує для збереження повної хронології покупок конкретного користувача. Вона може реалізовуватись як окремий клас або як складова частина об'єкта користувача, що дозволяє переглядати попередні придбання та повторно завантажувати ігри.

Загалом діаграма класів окреслює логіку роботи програмної платформи для замовлення комп'ютерних ігор як сукупність об'єктів, що взаємодіють між собою в контексті цифрової торгівлі, де користувачі можуть переглядати, купувати, коментувати та зберігати історію ігрових покупок у безпечному середовищі.

Діаграму компонентів програмної платформи для замовлення комп'ютерних ігор наведено на рис. 3.2.

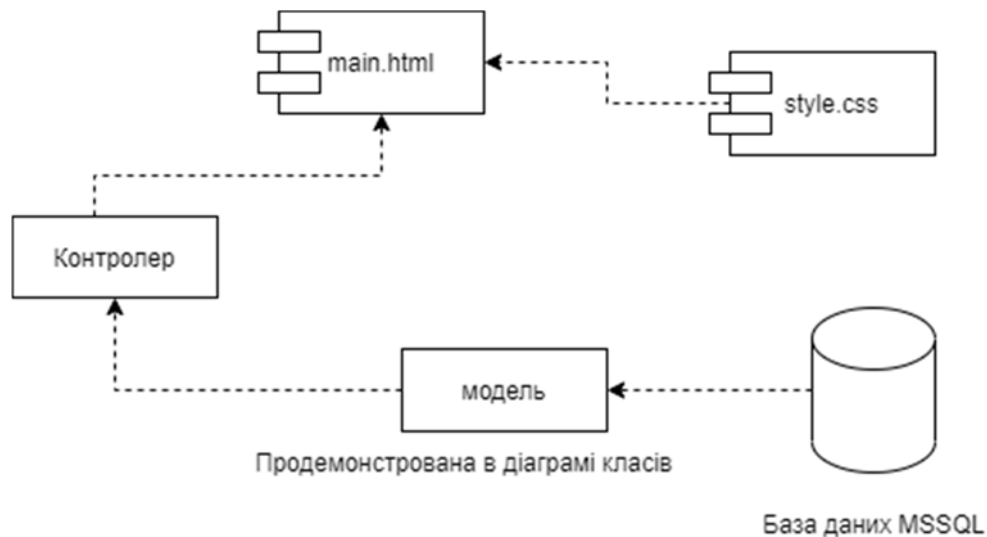


Рисунок 3.2 – Діаграма компонентів програмної платформи
для замовлення комп'ютерних ігор

Діаграма компонентів програмної платформи для замовлення комп'ютерних ігор відображає логічний поділ системи на окремі складові, кожна з яких виконує свою визначену роль у загальній архітектурі застосунку. Основу інтерфейсної частини становить компонент `main.html`, який відповідає за структуру та зміст вебсторінки, що відображається користувачу. У цьому компоненті зосереджено розміщення блоків товарів, форм авторизації, інформації про замовлення та інших елементів взаємодії.

Візуальне оформлення програмної платформи для замовлення комп'ютерних ігор реалізується через окремий компонент `style.css`, який регламентує стильову презентацію всієї системи. Завдяки цьому забезпечується єдиний зовнішній вигляд сторінок, зокрема кольорова схема, розмітка, анімації, відступи та адаптивність інтерфейсу до різних пристроїв. Цей файл не містить логіки, але тісно пов'язаний з `main.html`, формуючи його візуальну частину.

Обробкою запитів користувача програмної платформи для замовлення комп'ютерних ігор займається Контролер — логічний модуль, що діє як посередник між інтерфейсом і внутрішніми механізмами системи. Саме він

приймає запити з вебсторінки, обробляє введені дані, ініціює звернення до моделі та забезпечує повернення відповідей назад до клієнтської частини.

Функціональність програмної платформи для замовлення комп'ютерних ігор, пов'язана з опрацюванням бізнес-логіки, реалізована в компоненті Модель, де зосереджено роботу з даними: облік користувачів, обробка замовлень, перевірка наявності товарів, розрахунок вартості, генерація історії покупок тощо. Вона визначає структуру об'єктів, які використовуються системою, та формує логічну відповідність між тим, що відображається, і тим, як це зберігається.

Усі постійні дані, зокрема інформація про товари, замовлення, користувачів, відгуки та транзакції, зберігаються в базі даних, яка є центральним компонентом, пов'язаним безпосередньо з моделлю. Комунікація між моделлю та базою здійснюється через запити, які забезпечують отримання, додавання, зміну та видалення записів.

Узгоджена взаємодія цих компонентів формує цілісну архітектуру системи, де фронтенд, логіка та збереження даних організовані так, щоб забезпечити зручне й безпечне замовлення комп'ютерних ігор через вебінтерфейс.

3.2 Функціонування програмного забезпечення для замовлення комп'ютерних ігор

Функціонування програмного забезпечення для замовлення комп'ютерних ігор подамо за допомогою діаграми послідовностей (рис. 3.3) дій зареєстрованого користувача.

Діаграма послідовностей, що моделює логіку роботи програмної платформи для замовлення комп'ютерних ігор, демонструє поетапну взаємодію між основними учасниками процесу: Користувачем, Системою, компонентом Замовлення та Базою даних. У цій моделі показано, як

ініціюється цикл покупки — від пошуку гри до підтвердження оформленого замовлення.

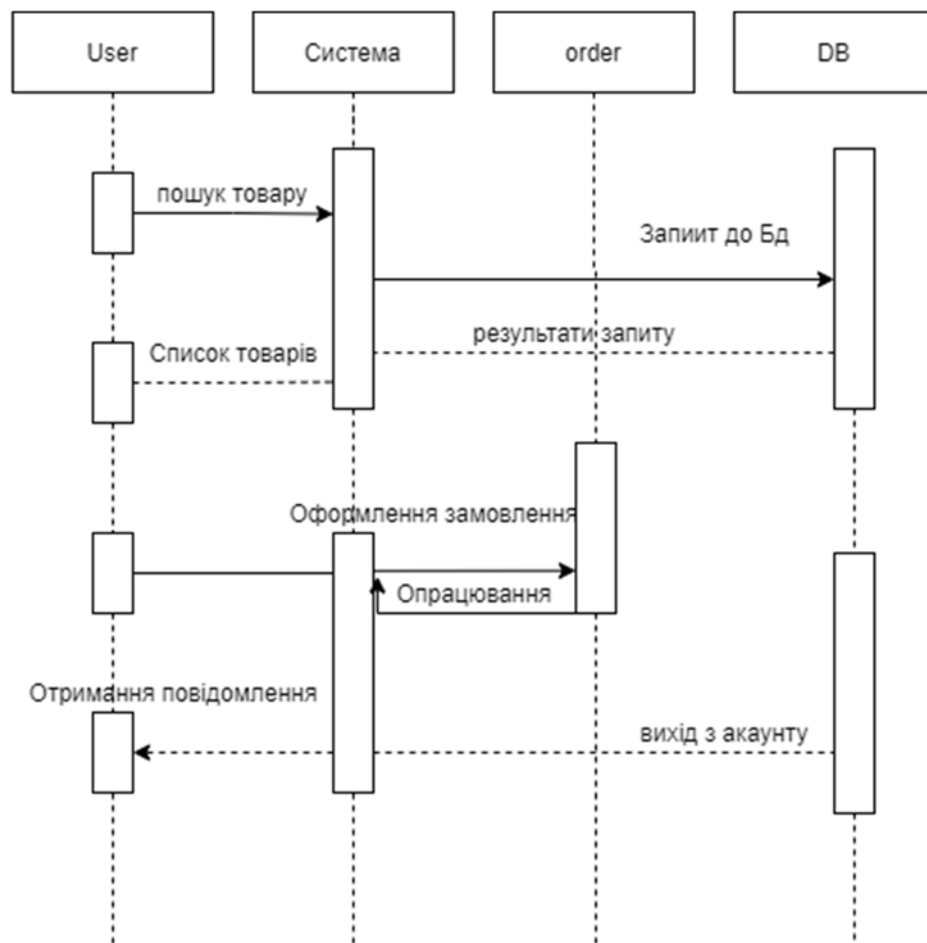


Рисунок 3.3 – Схема функціонування програмної платформи для замовлення комп'ютерних ігор у вигляді діаграми послідовностей

Все починається з дії користувача, який вводить запит для пошуку ігор на платформі. Система отримує введену інформацію, аналізує її та формує відповідний запит до бази даних. База даних повертає список доступних товарів, що відповідають критеріям пошуку, після чого система передає ці дані користувачу у вигляді результатів на екрані. Користувач, переглянувши перелік, вибирає потрібні товари й передає сформований список обраних позицій до системи для подальшої обробки.

Після отримання обраних товарів, система ініціює створення замовлення, передаючи дані в компонент Order, який перевіряє наявність усіх

об'єктів, розраховує підсумкову суму та фіксує інформацію про покупку. Потім відбувається запис усіх деталей у базу даних — до відповідних таблиць додається новий запис, що включає відомості про користувача, перелік придбаних ігор, дату й час замовлення, а також статус оплати.

Після завершення цього процесу система генерує повідомлення про успішне оформлення замовлення і надсилає його користувачеві. У результаті діаграма ілюструє цілком завершений сценарій дій, де всі об'єкти взаємодіють у чітко визначеній черговості, забезпечуючи стабільну та зрозумілу логіку електронної покупки. Така структура дозволяє легко масштабувати функціональність, додаючи нові кроки — наприклад, авторизацію, перевірку знижок або інтеграцію з платіжною системою.

Діаграму кооперації програмної платформи для замовлення комп'ютерних ігор наведено на рис. 3.4.

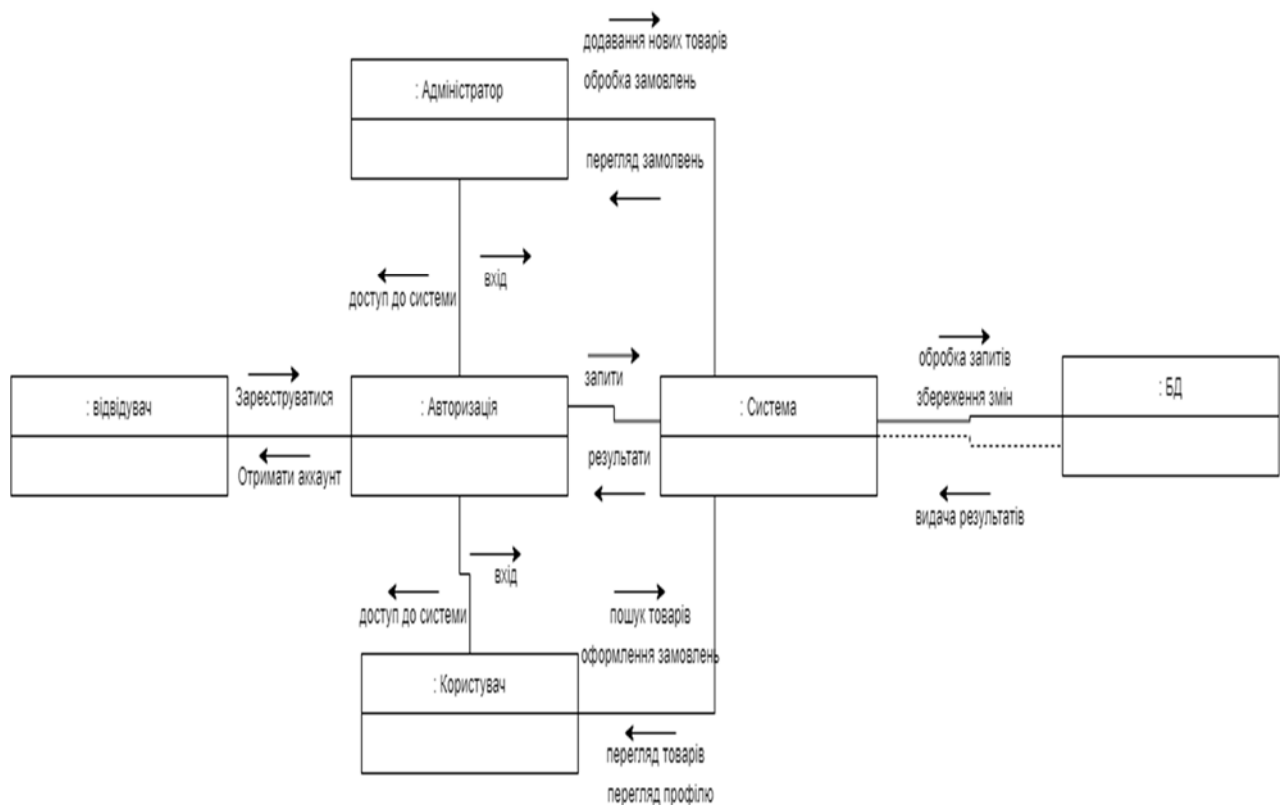


Рисунок 3.4 – Діаграма кооперації програмної платформи
для замовлення комп'ютерних ігор

Діаграма кооперації, що описує роботу програмної платформи для замовлення комп'ютерних ігор, відображає структуру взаємодії між ключовими об'єктами під час процесу ідентифікації користувача та доступу до функціоналу системи. У центрі сценарію — Відвідувач, який звертається до системи з наміром отримати доступ до персоналізованих можливостей.

Першим етапом є ініціація авторизації: коли відвідувач надає облікові дані (логін та пароль), цей запит потрапляє до модуля Авторизація, що виконує перевірку коректності введеної інформації. Авторизаційний компонент надсилає запит до Баз даних, де зберігаються облікові записи. Після пошуку в базі система отримує підтвердження — або про успішне входження, або про помилку аутентифікації.

У разі успішного входу створюється новий об'єкт — Користувач, що виступає вже автентифікованою сутністю у подальшій роботі. Цей користувач отримує доступ до функцій, які раніше були недоступні відвідувачу: перегляд історії замовлень, оформлення покупок, додавання коментарів, управління профілем тощо. Також може бути створений об'єкт Адміністратор, що редагувати інформацію у системі та переглядати замовлення, створені іншими користувачами.

Усі ці взаємодії відбуваються за участі Системи, яка координує обмін повідомленнями між компонентами. Саме вона отримує команди від користувача, переспрямовує запити до авторизаційного блоку, звертається до бази даних, і в разі успішної перевірки — передає керування створеному об'єкту користувача.

Таким чином, діаграма кооперації демонструє, як об'єкти тісно взаємодіють між собою в конкретному контексті — починаючи з анонімного запиту та завершуючи повноцінною участю користувача в роботі платформи. Зображена структура дозволяє побачити не лише логіку обміну повідомленнями, а й розподіл відповідальності між об'єктами, що забезпечує чітку організацію процесу доступу та персоналізації функціоналу.

Діаграму станів програмної платформи для замовлення комп'ютерних ігор наведено на рис. 3.5.

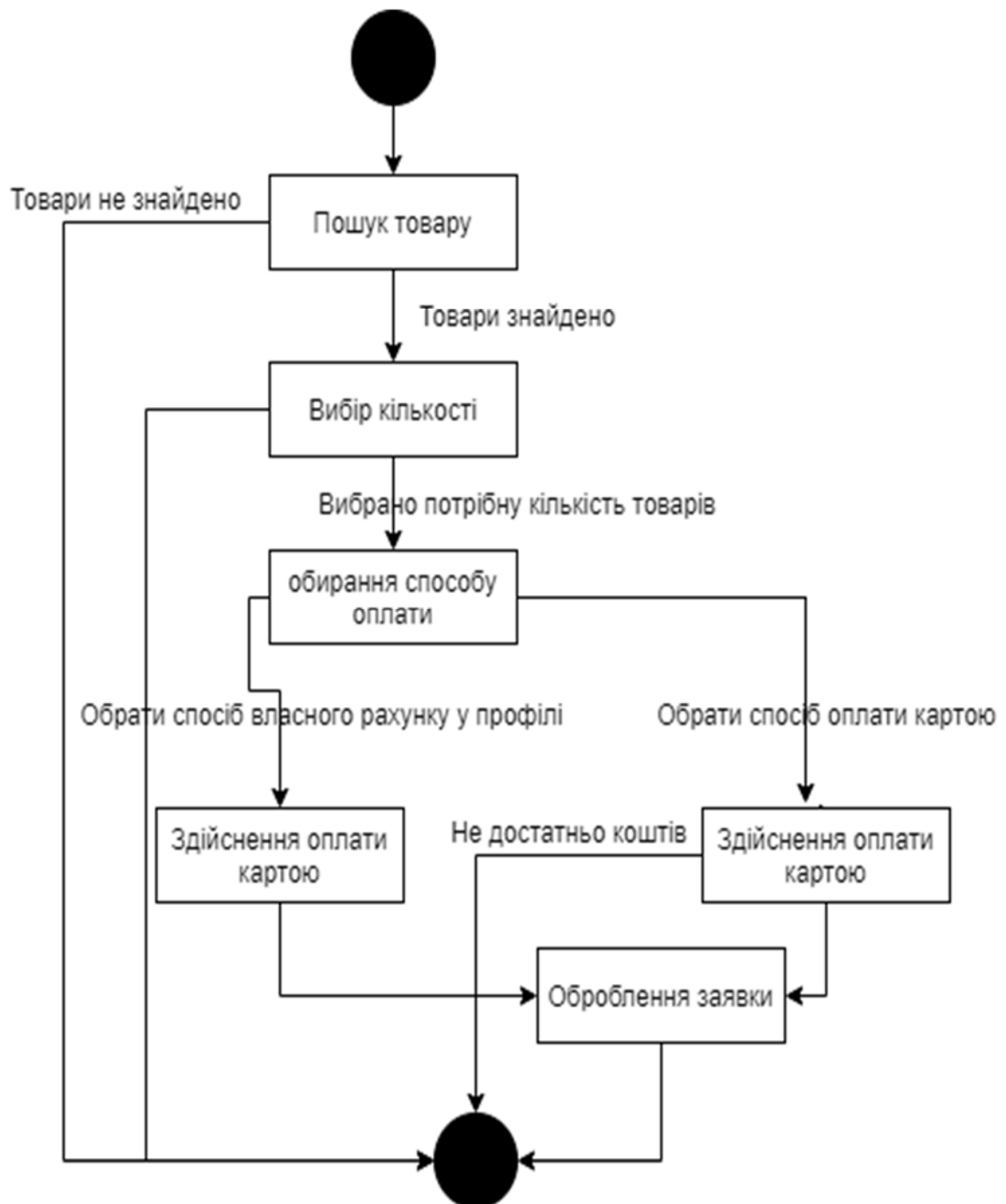


Рисунок 3.5 – Діаграма станів програмної платформи
для замовлення комп'ютерних ігор

Діаграма станів, яка описує поведінку програмної платформи для замовлення комп'ютерних ігор, демонструє послідовну зміну внутрішніх

станів системи під час виконання користувачем ключових дій, пов'язаних із процесом купівлі. У цій моделі кожен стан відображає конкретний етап взаємодії з платформою, а переходи між ними ініціюються діями користувача або результатами обробки інформації.

Початковим станом виступає Пошук товару, в якому система перебуває одразу після запуску сеансу користувача. У цьому режимі виконується аналіз запитів, формуються списки відповідних ігор, відображаються фільтри, рейтинги та інші допоміжні інструменти для прийняття рішення. Як тільки користувач обирає конкретну позицію, система переходить у стан Вибір кількості, де визначається, скільки копій гри або цифрових ключів буде додано до кошика.

Після цього активується наступний етап, Обирання способу оплати. Тут відбувається вибір між доступними варіантами транзакцій: електронні гаманці, банківські картки, платіжні системи чи внутрішній баланс користувача. На цьому етапі також можуть застосовуватися бонуси, знижки або промокоди, що впливає на фінальний розрахунок.

Далі програмне платформи для замовлення комп'ютерних ігор автоматично переходить у стан Оброблення заявки. У цьому режимі відбувається перевірка валідності платіжних даних, підтвердження наявності товару, фіксація замовлення в базі даних, а також підготовка до надсилання користувачеві підтвердження або цифрового коду для завантаження.

Завершення процесу може перевести програмну платформу для замовлення комп'ютерних ігор у фінальний стан, який відображає успішне оформлення замовлення або повідомлення про помилку в разі виникнення проблем на етапі обробки. Така діаграма дозволяє чітко уявити, як змінюється логічний стан системи впродовж усього життєвого циклу однієї транзакції, і як забезпечується послідовність і надійність у взаємодії з користувачем.

Діаграму діяльності програмної платформи для замовлення комп'ютерних ігор наведено на рис. 3.6.

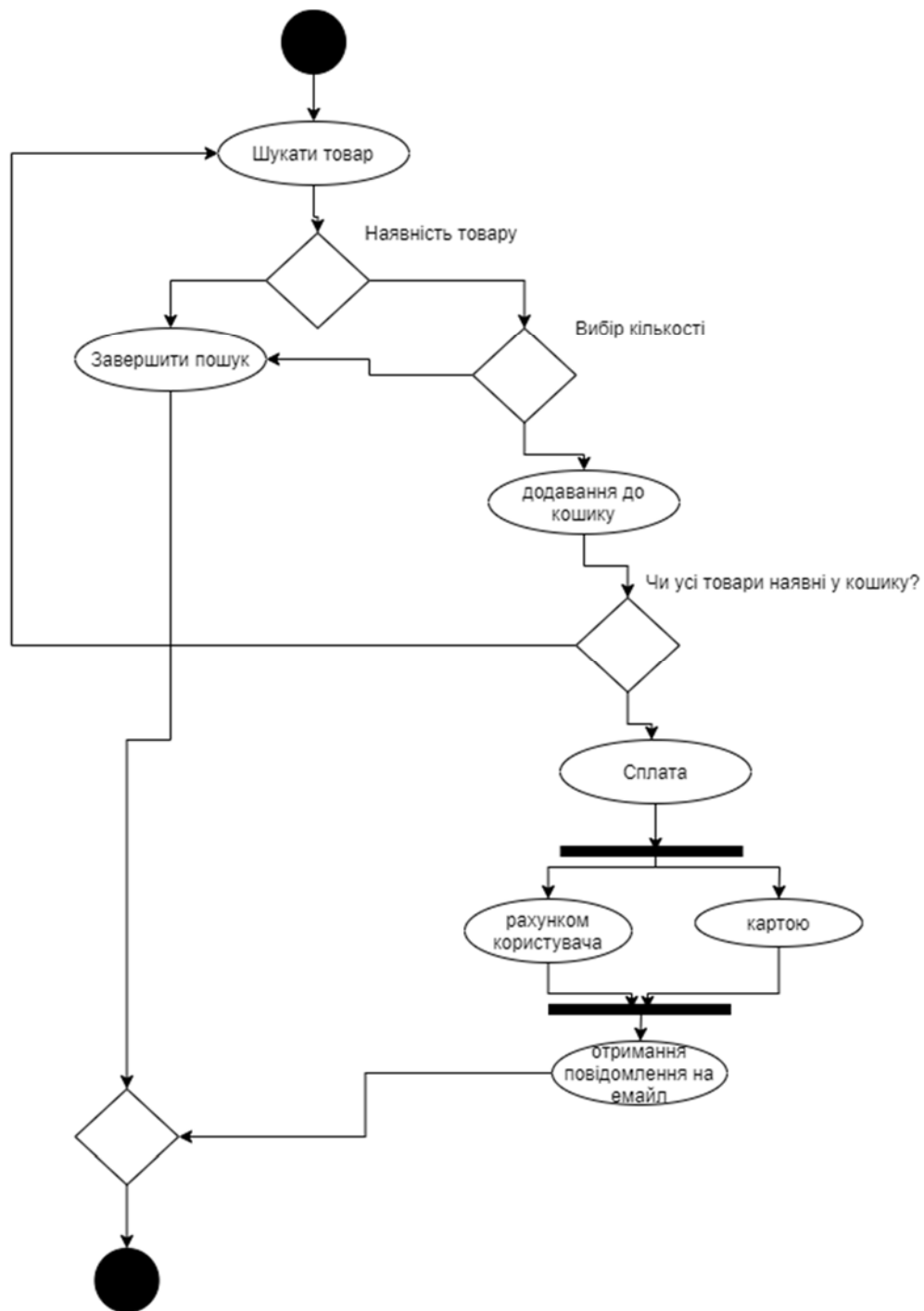


Рисунок 3.6 – Діаграма станів програмної платформи
для замовлення комп'ютерних ігор

3.3 Розробка бази даних програмного забезпечення для замовлення комп'ютерних ігор

База даних програмного забезпечення для замовлення комп'ютерних ігор слугує центральним сховищем усіх структурованих даних, необхідних для

функціонування платформи. Її структура організована у вигляді взаємопов'язаних таблиць, які зберігають відомості про користувачів, товари, замовлення, коментарі, оплату та інші ключові аспекти цифрової торгівлі.

Схему бази даних програмного забезпечення для замовлення комп'ютерних ігор наведено на рис. 3.7.

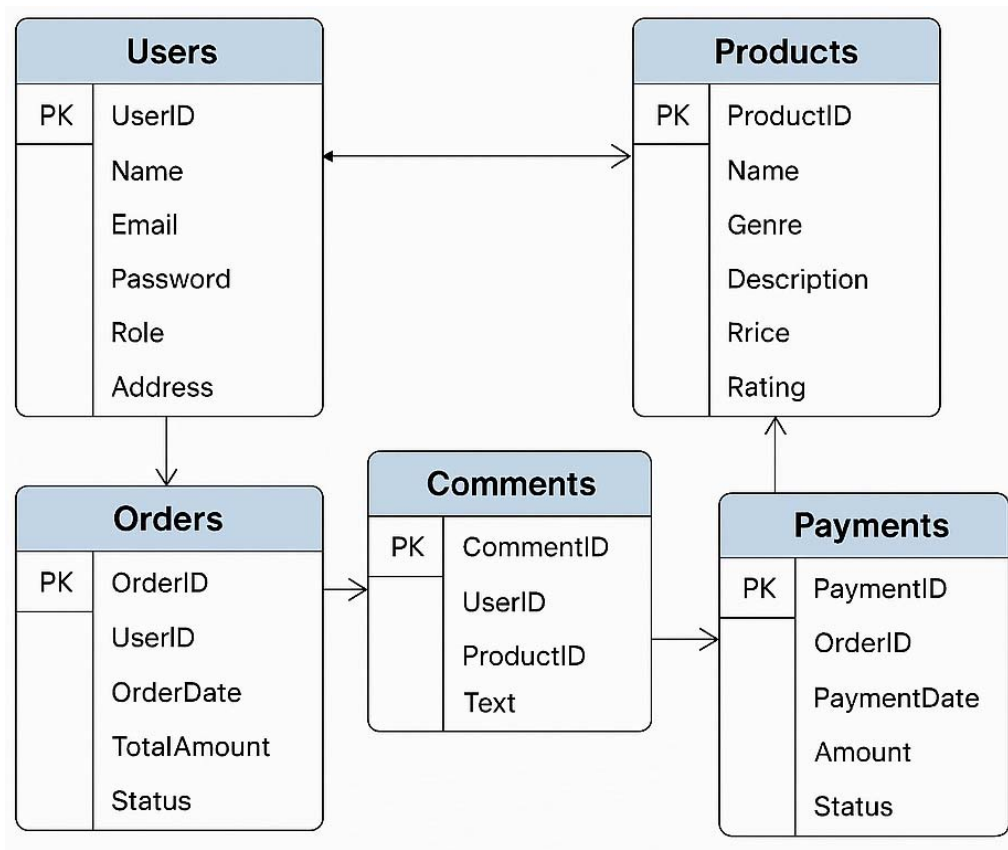


Рисунок 3.7 – Схема бази даних програмного забезпечення для замовлення комп'ютерних ігор

Основою логіки бази даних є таблиця **Users**, що містить інформацію про всіх зареєстрованих користувачів системи: імена, електронні адреси, паролі (зашифровані), ролі (звичайний користувач, адміністратор), а також контактні дані. З цією таблицею пов'язані інші сутності — наприклад, таблиця **Orders**, яка зберігає історію покупок кожного користувача. Кожен запис у ній включає дату замовлення, загальну суму, статус (оплачено, в обробці,

скасовано), а також зовнішній ключ, що вказує на користувача, який здійснив покупку.

Для опису ігор, доступних у системі, використовується таблиця Products, яка містить назви, жанри, опис, ціну, рейтинг, платформу, системні вимоги та посилання на медіафайли. З цією таблицею пов'язується таблиця Orders — вона дозволяє деталізувати, які саме ігри входять до кожного окремого замовлення. Кожен запис у цій таблиці містить ідентифікатор товару, замовлення та кількість одиниць.

Окремо функціонує таблиця Comments, яка фіксує відгуки користувачів щодо ігор. Вона пов'язана як із таблицею користувачів, так і з таблицею товарів, що дозволяє виводити список коментарів до певної гри або переглядати історію дописів конкретного користувача.

Ще один важливий елемент — таблиця Payments, яка включає відомості про проведені транзакції, способи оплати, статус платежу, дату та суму.

Завдяки реляційній структурі, база даних забезпечує цілісність, надійність та ефективність у зберіганні й обробці даних, дозволяючи платформі швидко виконувати запити, обробляти великі обсяги інформації та масштабуватись без втрати продуктивності. Така архітектура створює надійний фундамент для стабільної роботи цифрового магазину комп'ютерних ігор.

3.4 Проєктування інтерфейсу програмного забезпечення для замовлення комп'ютерних ігор

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для замовлення комп'ютерних ігор враховані функціональні вимоги до програми:

- підтримка можливості відображення переліку комп'ютерних ігор, доступних для завантаження;
- підтримка можливості групування комп'ютерних ігор за обраною користувачем категорією (наприклад, ігри з пошуком предметів, ігри-ребуси, аркадні ігри і т.ін.);
- можливість відображення категорій комп'ютерних ігор;
- можливість авторизації;
- підтримка можливості обчислення та виведення користувачу інформації про кількість комп'ютерних ігор у відповідній категорії;
- підтримка можливості подання інформації про обрану комп'ютерну гру;
- можливість завантаження обраної комп'ютерної гри.

Інтерфейс головної форми програмного забезпечення для замовлення комп'ютерних ігор наведено на рис. 3.8.

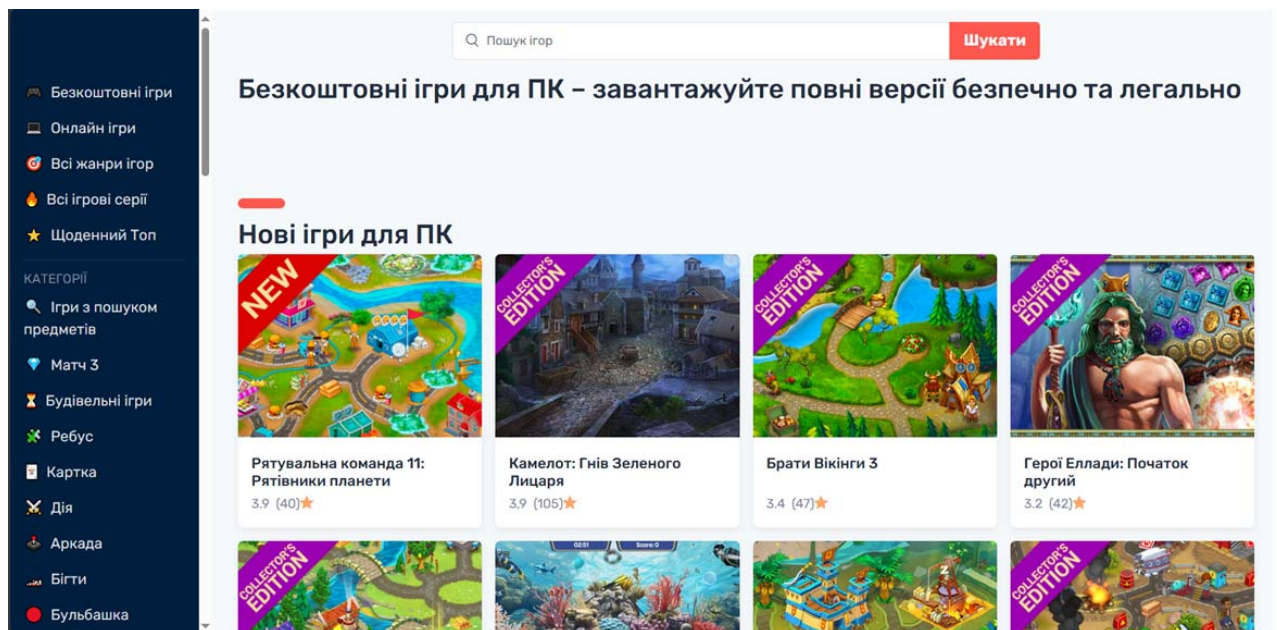
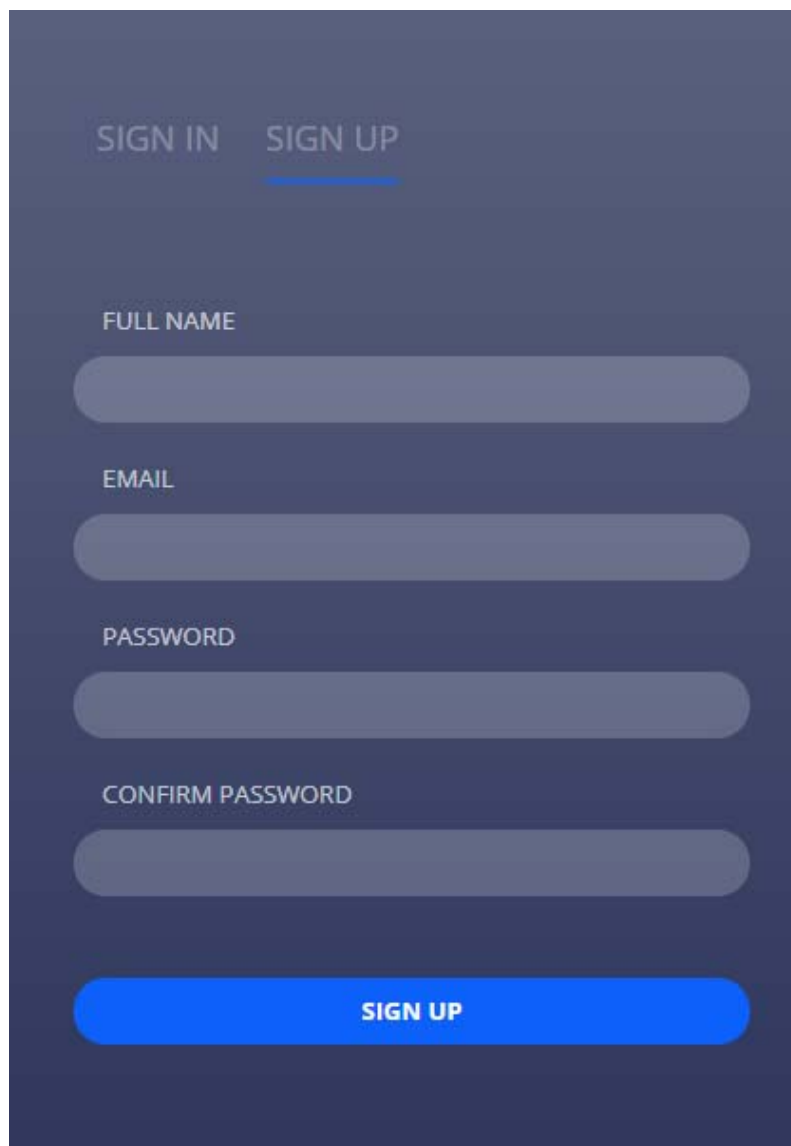


Рисунок 3.8 – Інтерфейс головної форми програмного забезпечення для замовлення комп'ютерних ігор

Форму авторизації користувачів програмної платформи для замовлення комп'ютерних ігор наведено на рис. 3.9.



The image shows a user registration form on a dark blue background. At the top, there are two links: "SIGN IN" and "SIGN UP", with "SIGN UP" being underlined. Below these are four input fields, each with a label above it: "FULL NAME", "EMAIL", "PASSWORD", and "CONFIRM PASSWORD". Each field is represented by a rounded rectangle. At the bottom of the form is a prominent blue button with the text "SIGN UP" in white capital letters.

Рисунок 3.9 – Форма авторизації у програмній платформі для замовлення комп'ютерних ігор

Сторінку жанрів ігор програми для замовлення комп'ютерних ігор наведено на рис. 3.10.

Сторінку з прикладом групування комп'ютерних ігор за категорією Аркадні ігри наведено на рис. 3.11.

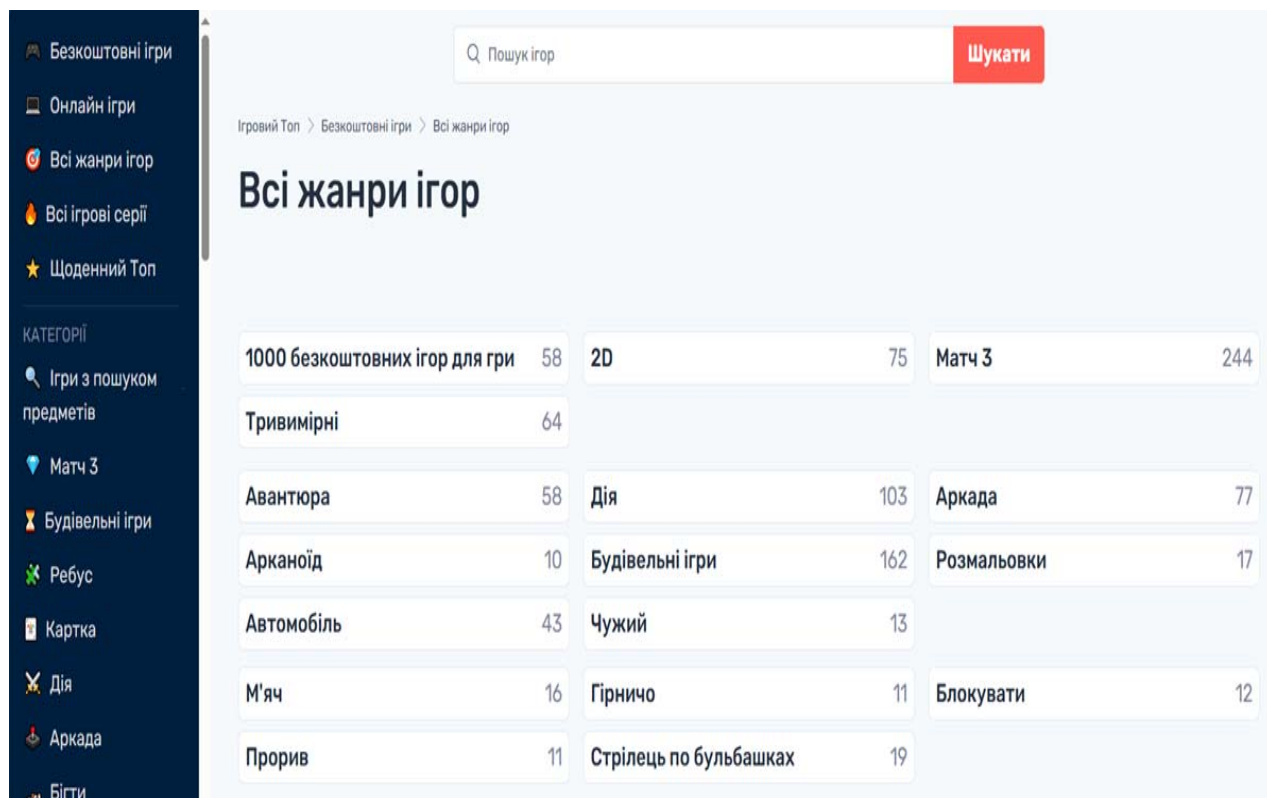


Рисунок 3.10 – Сторінка жанрів ігор програми
для замовлення комп'ютерних ігор

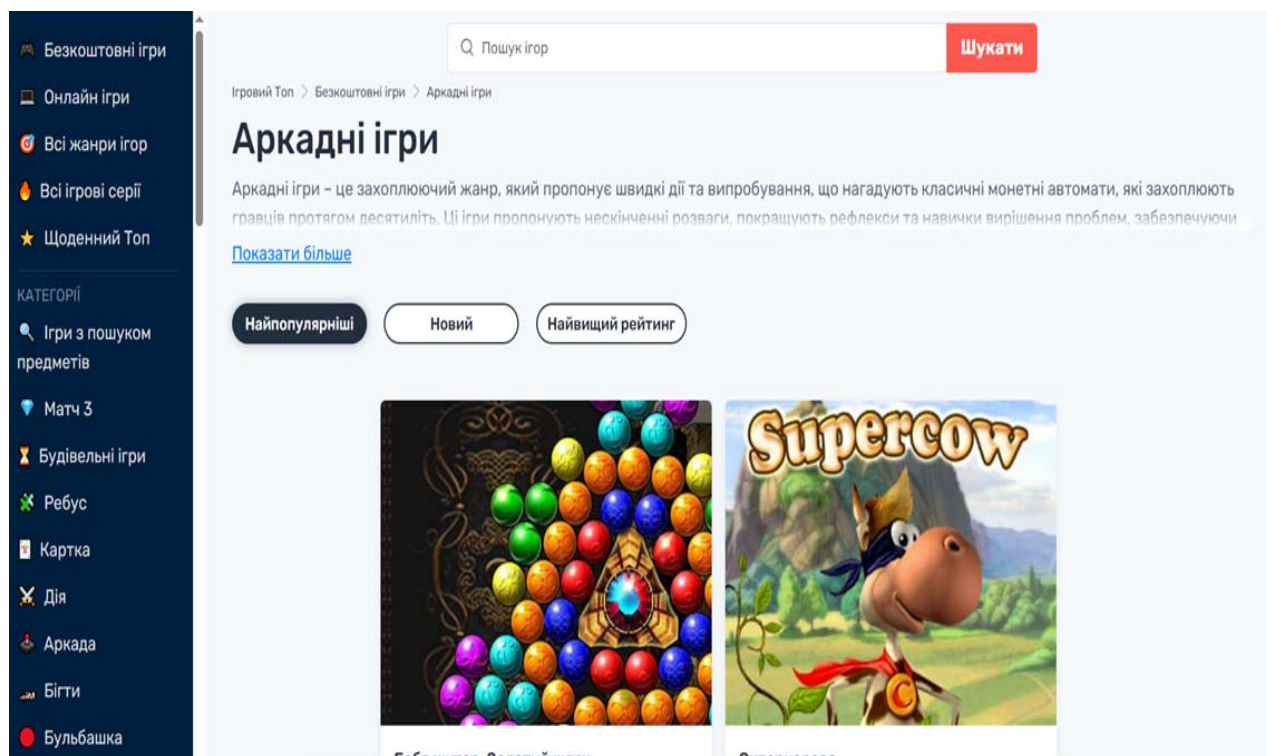


Рисунок 3.11 – Групування ігор за категорією Аркадні ігри

Сторінка з інформацією про гру наведена на рис. 3.12.

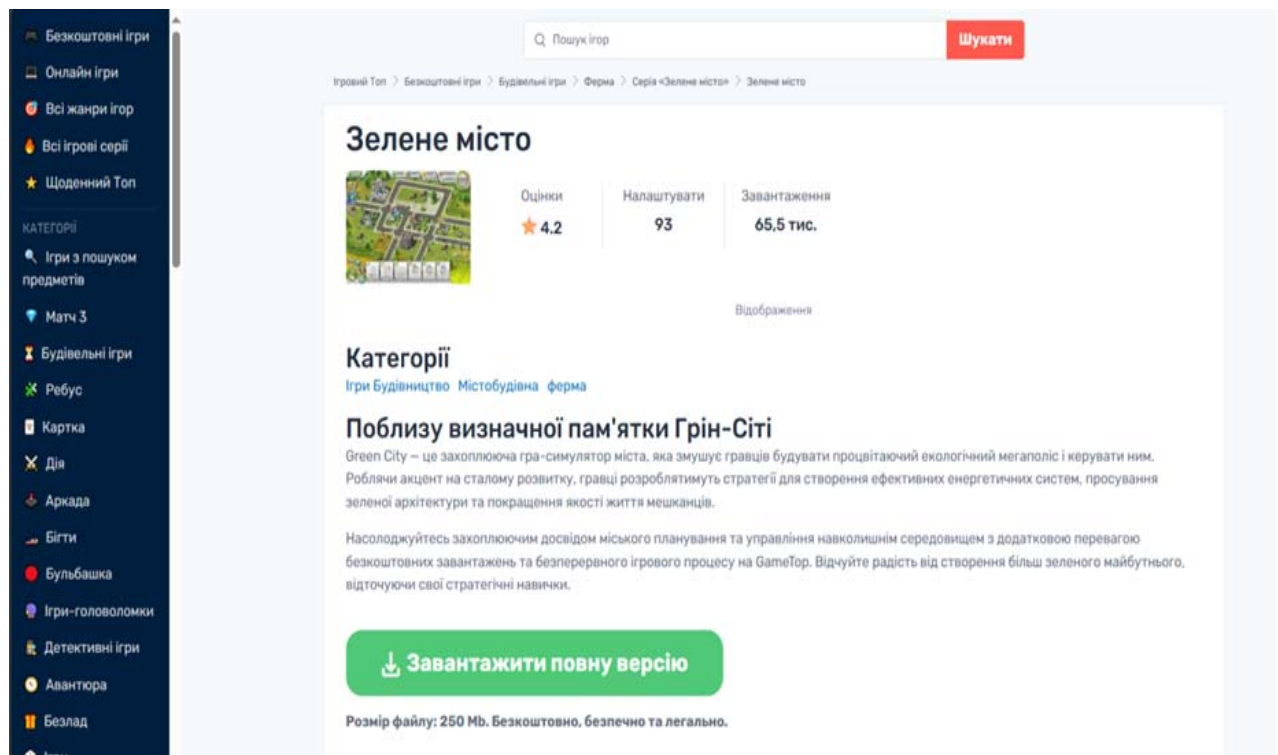


Рисунок 3.12 – Сторінка з інформацією про гру

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для замовлення комп'ютерних ігор.

Запропоновано структуру програмного забезпечення для замовлення комп'ютерних ігор. Визначено, що основними класами програмної платформи для замовлення комп'ютерних ігор Користувач, Незареєстрований користувач, Адміністратор, Товар, Замовлення, Коментар, Історія замовлень та інші.

Функціонування програмного забезпечення для замовлення комп'ютерних ігор подано за допомогою діаграми послідовностей дій зареєстрованого користувача. Діаграма послідовностей, що моделює логіку роботи програмної платформи для замовлення комп'ютерних ігор, демонструє поетапну взаємодію між основними учасниками процесу: Користувачем, Системою, компонентом Замовлення та Базою даних. У цій моделі показано,

як ініціюється цикл покупки — від пошуку гри до підтвердження оформленого замовлення.

Розроблено базу даних програмного забезпечення для замовлення комп'ютерних ігор, яка слугує центральним сховищем усіх структурованих даних, необхідних для функціонування платформи. Її структура організована у вигляді взаємопов'язаних таблиць, які зберігають відомості про користувачів, товари, замовлення, коментарі, оплату та інші ключові аспекти цифрової торгівлі.

Описано особливості реалізації програмного забезпечення для замовлення комп'ютерних ігор. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для замовлення комп'ютерних ігор. Програма написана на мові C#, яка забезпечує можливість створення кросплатформених рішень, інтеграцію з базами даних, а також надає механізми безпеки. Крім того, мова C# підтримує зручність розробки та високу продуктивність, а також має активну спільноту та підтримку. В якості середовища розробки для створення програмного забезпечення для замовлення комп'ютерних ігор обрано Visual Studio, що поєднує в собі інтелектуальні підказки, засоби рефакторингу, вбудовану підтримку систем контролю версій, а також зручний інтерфейс для роботи з базами даних, програмними сервісами та користувацьким інтерфейсом.

Програмне забезпечення для замовлення комп'ютерних ігор забезпечує виконання таких функцій:

- підтримка можливості відображення переліку комп'ютерних ігор, доступних для завантаження;
- підтримка можливості групування комп'ютерних ігор за обраною користувачем категорією (наприклад, ігри з пошуком предметів, ігри-ребуси, аркадні ігри і т.ін.);
- можливість відображення категорій комп'ютерних ігор;
- можливість авторизації;
- підтримка можливості обчислення та виведення користувачу інформації про кількість комп'ютерних ігор у відповідній категорії;
- підтримка можливості подання інформації про обрану комп'ютерну гру;
- можливість завантаження обраної комп'ютерної гри.

Для роботи програмного забезпечення для замовлення комп'ютерних ігор потрібні такі технічні та програмні ресурси: комп'ютерна система

повинна бути оснащена процесором з тактовою частотою не нижче 2,5 ГГц (рекомендовано багатоядерний), щоб забезпечити швидке опрацювання запитів і стабільну взаємодію з базою даних. Оперативна пам'ять має становити не менше 8 ГБ, що дозволяє підтримувати одночасну роботу веб-сервера, бази даних і фонових служб без зниження продуктивності. Для зберігання інформації про товари, замовлення та історію покупок потрібен жорсткий диск або SSD із обсягом вільного простору від 50 ГБ і вище, залежно від масштабів системи та кількості користувачів. З боку програмного середовища необхідно використовувати сучасну операційну систему — наприклад, Windows 10/11 або одна з останніх стабільних версій Linux. Для коректного запуску серверної частини рекомендується встановити .NET Runtime (відповідної версії під розробку), веб-сервер типу IIS або Apache, а також СУБД, сумісну з реалізованою моделлю даних — наприклад, Microsoft SQL Server або PostgreSQL. Для безпечної роботи варто передбачити активну систему захисту й оновлення компонентів. Браузер на стороні клієнта має підтримувати сучасні вебтехнології HTML5, CSS3 та JavaScript для коректного відображення інтерфейсу користувача.

Функціональні характеристики розробленого програмного забезпечення для замовлення комп'ютерних ігор наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для замовлення комп'ютерних ігор наведено у додатку Б.

4.2 Характеристики програми для замовлення комп'ютерних ігор

Програмне забезпечення для замовлення комп'ютерних ігор вирізняється зручністю використання, що проявляється у зрозумілому інтерфейсі та простій навігації між розділами каталогу. У такій системі реалізовано стабільну роботу із запитамі користувачів, швидке опрацювання пошукових запитів і безпечну обробку замовлень. Передбачено гнучку архітектуру, що дозволяє доповнювати функції та масштабувати можливості

залежно від обсягу товарів і кількості одночасних підключень. В основі закладено механізми збереження історії покупок, формування замовлень та їхнього відстеження. Програмне середовище підтримує взаємодію з платіжними сервісами, що гарантує захист особистих даних і конфіденційність фінансових операцій. Інтеграція з базою даних забезпечує цілісність інформації про товари, користувачів та транзакції. У роботі використовуються сучасні підходи до безпеки та шифрування, що мінімізує ризики втрати даних. Вся структура побудована так, щоб підтримувати безперервну доступність сервісу та своєчасне оновлення контенту.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для замовлення комп'ютерних ігор на клієнтській стороні необхідно перейти на відповідну вебсторінку.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для замовлення комп'ютерних ігор є інформація про обрані користувачем комп'ютерні ігри, категорії ігор та ін.

Вихідними даними програмного забезпечення для замовлення комп'ютерних ігор є сформовані сторінки порталу відповідно до дій користувача.

4.3.3 Повідомлення

Користувач програмного забезпечення для замовлення комп'ютерних ігор може отримати такі повідомлення: повідомлення про успішне

завантаження комп'ютерної гри, повідомлення про проблеми завантаження гри та інші.

4.4 Виконання програмного забезпечення для замовлення комп'ютерних ігор

Після запуску програмного забезпечення для замовлення комп'ютерних ігор користувачу відображається головна сторінка (рис. 4.1). Зліва сторінки розташовано панель, за допомогою якої можна відобразити ігри певної категорії. В основній області сторінці відображено інформацію про комп'ютерні ігри.

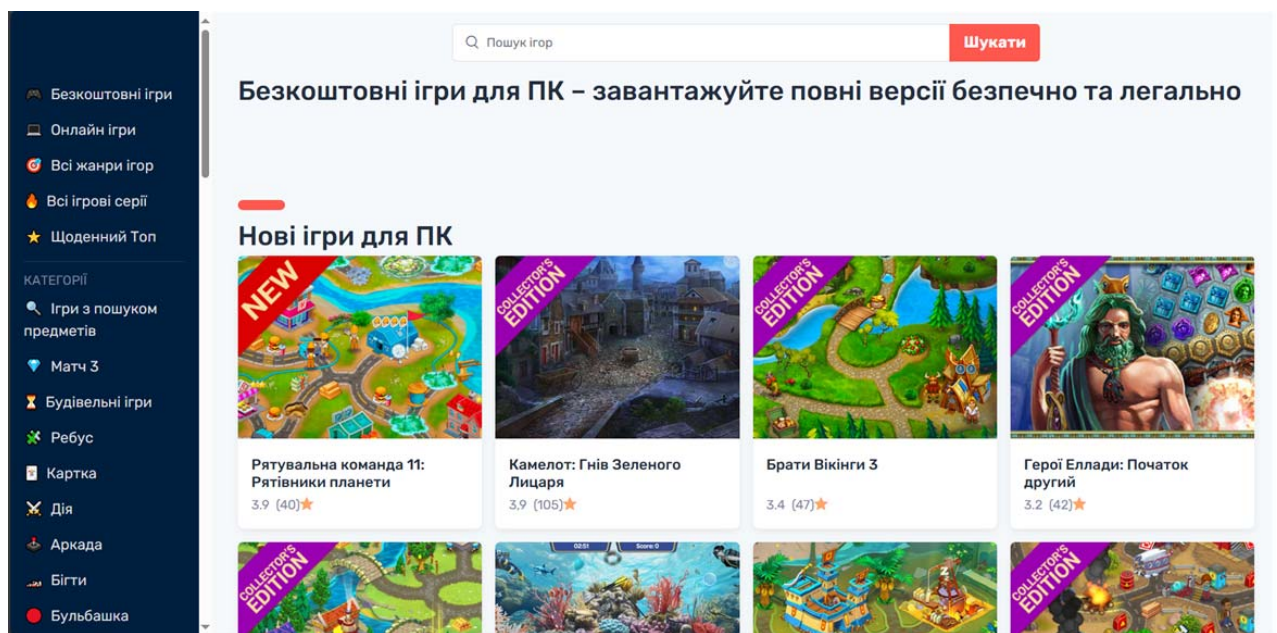


Рисунок 4.1 – Головна сторінка програмної платформи для замовлення комп'ютерних ігор

Форму авторизації користувачів програмної платформи для замовлення комп'ютерних ігор наведено на рис. 4.2.

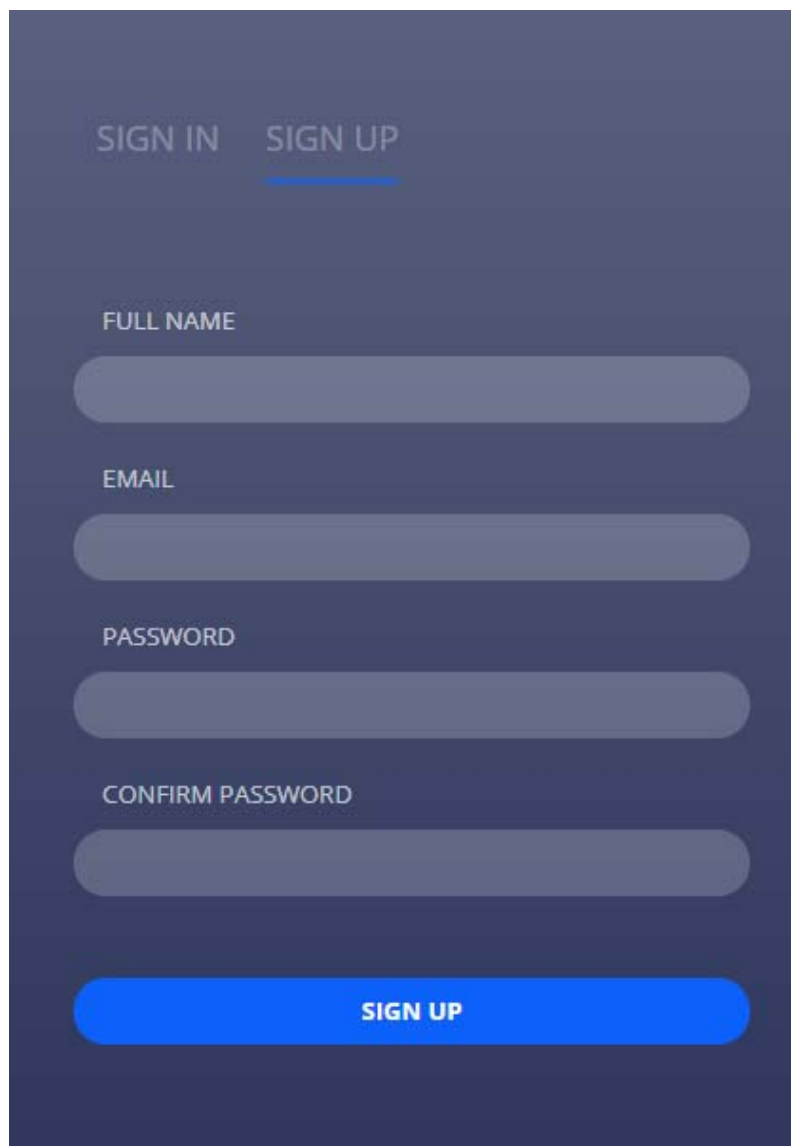
The image shows a registration form on a dark blue background. At the top, there are two links: 'SIGN IN' and 'SIGN UP', with 'SIGN UP' being underlined in blue. Below these are four input fields with labels: 'FULL NAME', 'EMAIL', 'PASSWORD', and 'CONFIRM PASSWORD'. Each label is in white uppercase letters above a rounded rectangular input field. At the bottom of the form is a large, rounded blue button with the text 'SIGN UP' in white uppercase letters.

Рисунок 4.2 – Форма авторизації у програмній платформі для замовлення комп'ютерних ігор

Для ознайомлення з категоріями комп'ютерних ігор необхідно обрати пункт меню Всі жанри ігор зліва головної сторінки. На сторінці Всі жанри ігор (рис. 4.3) відображається перелік ігрових жанрів, а також кількість комп'ютерних ігор у відповідній категорії.

При виборі ігрового жанру відображається сторінка (рис. 4.4), яка містить опис цієї категорії комп'ютерних ігор, а також перелік ігор, що відповідають цьому жанру.

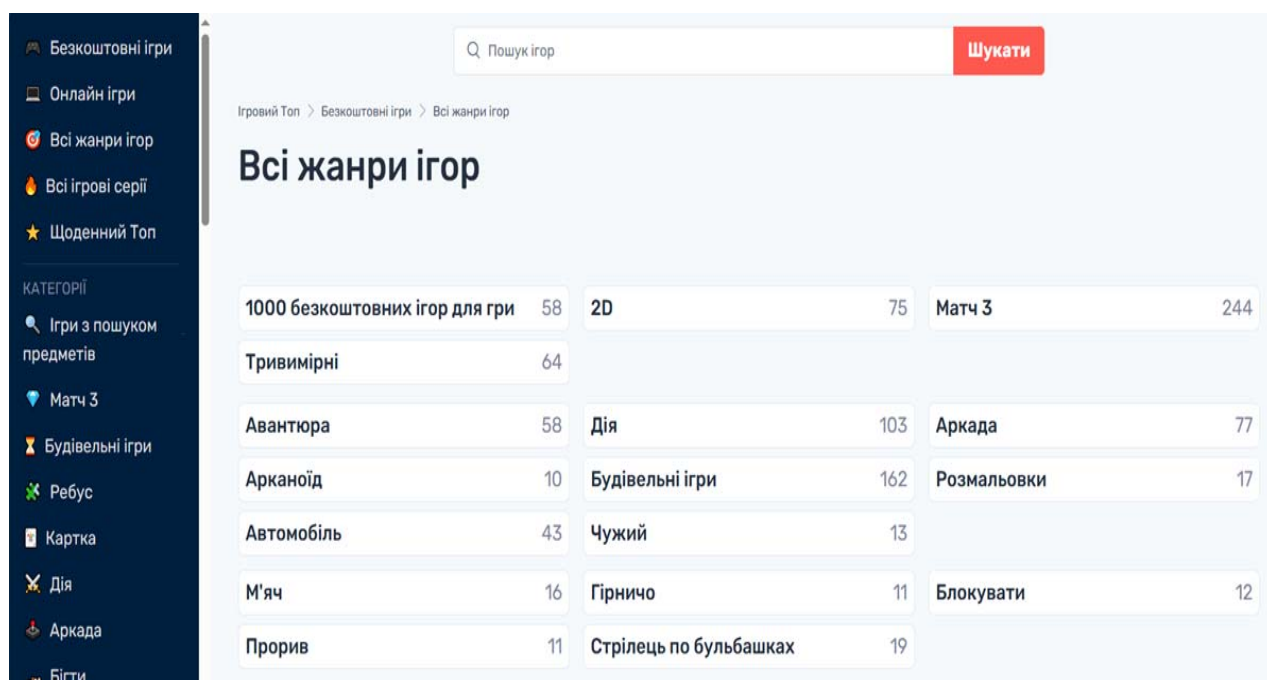


Рисунок 4.3 – Сторінка жанрів ігор програми
для замовлення комп'ютерних ігор

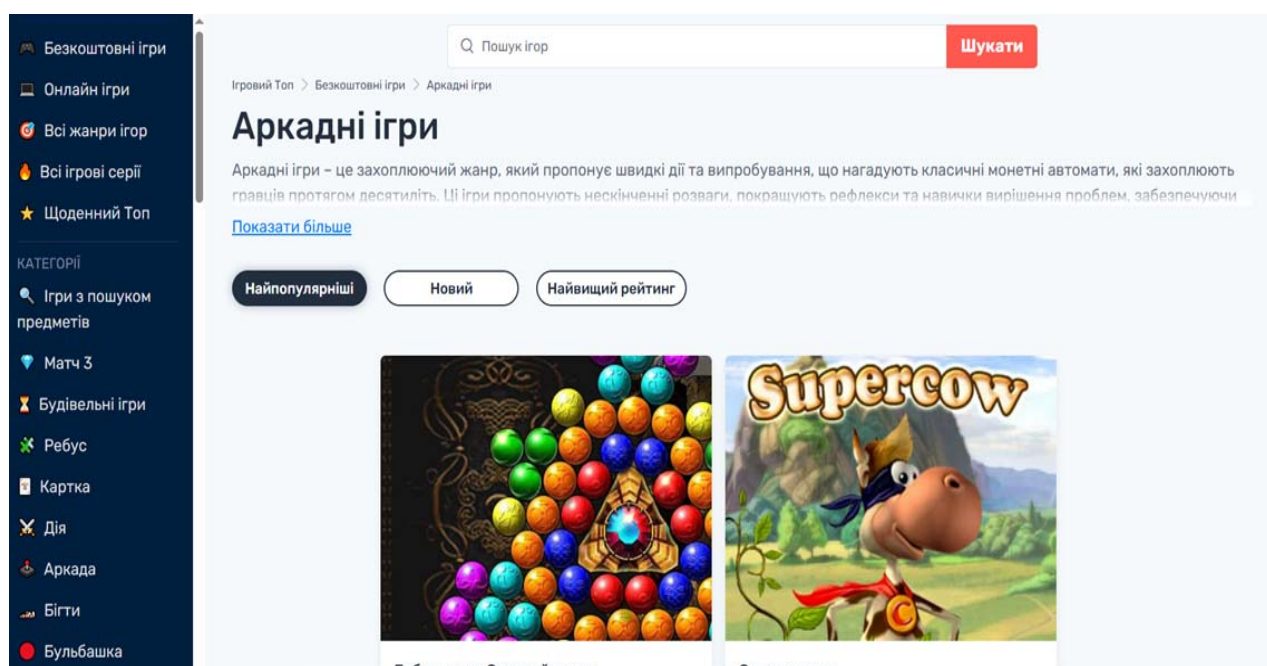


Рисунок 4.4 – Групування ігор за категорією Аркадні ігри

При виборі конкретної гри відображається сторінка цієї гри (рис. 4.5) з детально інформацією про неї. Для завантаження обраної гри необхідно натиснути відповідну кнопку.

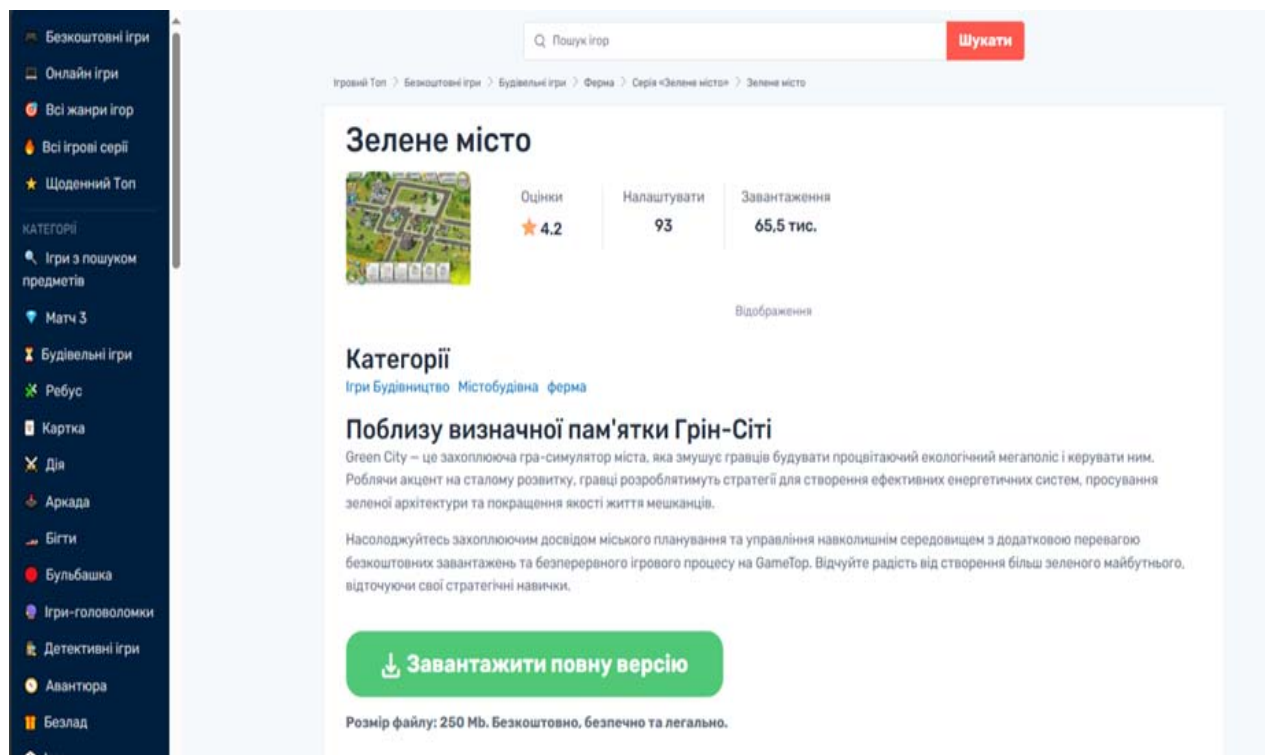


Рисунок 4.5 – Сторінка з інформацією про гру

Для оцінювання комп'ютерної гри (рис. 4.6) зареєстрованому користувачу необхідно обрати кількість зірок (від однієї до п'яти), що відповідає рівню враження від гри (від найменшого до найбільшого).

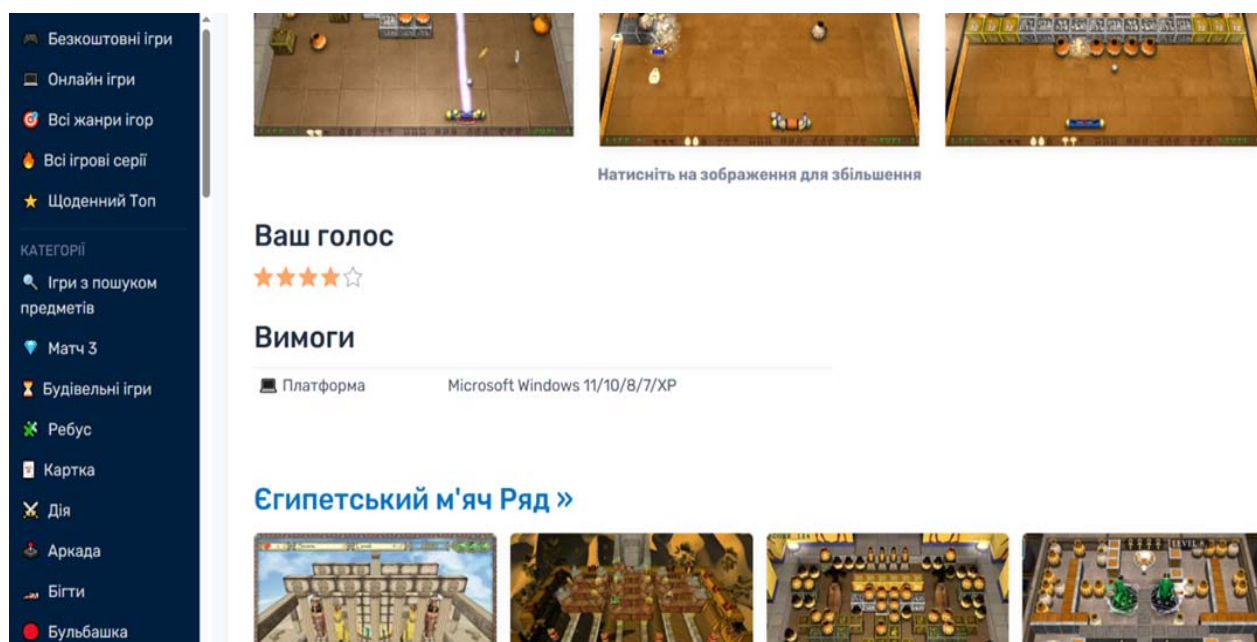


Рисунок 4.6 – Оцінювання комп'ютерної гри

4.5 Тестування програмного забезпечення для замовлення комп'ютерних ігор

Виконано тестування програмного забезпечення для замовлення комп'ютерних ігор.

Виявлено, що програма для замовлення комп'ютерних ігор функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу організації подання інформації та продажу комп'ютерних ігор.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою продажу комп'ютерних ігор.

4.6 Висновки за розділом 4

Описано програмне забезпечення для замовлення комп'ютерних ігор. Виконано тестування розробленого програмного забезпечення для замовлення комп'ютерних ігор. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою продажу комп'ютерних ігор.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для замовлення комп'ютерних ігор.

Визначено, що програмне забезпечення для замовлення комп'ютерних ігор надає можливість завантаження розважального ігрового контенту безпосередньо на свої пристрої користувача. Різні програми та сервіси для замовлення комп'ютерних ігор дають змогу створювати цілі екосистеми, де користувачі реєструються, обирають і купують ігри, а розробники поширюють свої проєкти без посередників. Такі програмні інструменти об'єднують в одному середовищі каталог продуктів, систему захисту, оплату та управління бібліотекою придбань.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для замовлення комп'ютерних ігор. Проте деякі програмні засоби для замовлення комп'ютерних ігор можуть мати модулі захисту, які вимагають постійного онлайн контролю, можуть негативно впливати на продуктивність і навіть викликати конфлікти з апаратним або іншим програмним забезпеченням. Також варто відзначити монополізацію ринку окремими платформами, що призводить до обмеженого вибору, ексклюзивності контенту й інколи — до примусового використання кількох клієнтів одночасно, що ускладнює досвід користувача. Втрата доступу до контенту — ще один ризик. У разі видалення гри з платформи, зміни політики компанії або блокування облікового запису користувач фактично втрачає право на користування раніше придбаними іграми, оскільки більшість ПЗ працює за моделлю ліцензії, а не повного володіння продуктом. Також можливі проблеми із сумісністю або технічними помилками, особливо при оновленнях клієнтського ПЗ або при встановленні ігор, які більше не підтримуються. Це створює труднощі, з якими користувачі змушені

самостійно справлятися. Тому актуальною є розробка програмного забезпечення для замовлення комп'ютерних ігор.

Сформульовано функціональні вимоги до програмного забезпечення для замовлення комп'ютерних ігор.

Для реалізації програмного забезпечення для замовлення комп'ютерних ігор обрано мову програмування C#, яка забезпечує можливість створення кросплатформених рішень, інтеграцію з базами даних, а також надає механізми безпеки. Крім того, мова C# підтримує зручність розробки та високу продуктивність, а також має активну спільноту та підтримку.

Для створення програмного забезпечення для замовлення комп'ютерних ігор обрано середовище розробки Visual Studio, що поєднує в собі інтелектуальні підказки, засоби рефакторингу, вбудовану підтримку систем контролю версій, а також зручний інтерфейс для роботи з базами даних, програмними сервісами та користувацьким інтерфейсом. Це середовище створене так, щоб розробник міг швидко переходити від ідеї до реалізації без зайвих технічних бар'єрів.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для замовлення комп'ютерних ігор.

Практична цінність роботи полягає у тому, що створено програмне забезпечення для замовлення комп'ютерних ігор.

Запропоновано структуру програмного забезпечення для замовлення комп'ютерних ігор. Визначено, що основними класами програмної платформи для замовлення комп'ютерних ігор Користувач, Незареєстрований користувач, Адміністратор, Товар, Замовлення, Коментар, Історія замовлень та інші.

Функціонування програмного забезпечення для замовлення комп'ютерних ігор подано за допомогою діаграми послідовностей дій зареєстрованого користувача. Діаграма послідовностей, що моделює логіку роботи програмної платформи для замовлення комп'ютерних ігор, демонструє

поетапну взаємодію між основними учасниками процесу: Користувачем, Системою, компонентом Замовлення та Базою даних. У цій моделі показано, як ініціюється цикл покупки — від пошуку гри до підтвердження оформленого замовлення.

Розроблено базу даних програмного забезпечення для замовлення комп'ютерних ігор, яка слугує центральним сховищем усіх структурованих даних, необхідних для функціонування платформи. Її структура організована у вигляді взаємопов'язаних таблиць, які зберігають відомості про користувачів, товари, замовлення, коментарі, оплату та інші ключові аспекти цифрової торгівлі.

Описано особливості реалізації програмного забезпечення для замовлення комп'ютерних ігор. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор.

Виконано тестування розробленого програмного забезпечення для замовлення комп'ютерних ігор. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою продажу комп'ютерних ігор.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Exploring Different Types of Video Game Digital Distribution Platforms [Electronic resource]. – Access mode: <https://bambamtastic.com/what-the-different-types-of-video-game-digital-distribution-platforms/>.
2. Where Do PC Gamers Buy Games? The Ultimate Guide [Electronic resource]. – Access mode: <https://psutierlist.com/where-do-pc-gamers-buy-games/>.
3. Top 10 Best Online Video Game Stores [Electronic resource]. – Access mode: <https://www.top10.com/online-video-game-stores>.
4. Top 10 Online Video Game Platforms to Buy and Download PC Games [Electronic resource]. – Access mode: <https://www.extrabux.com/en/guide/6925063>.
5. Best Places to Buy PC Games Online [Electronic resource]. – Access mode: <https://www.gamersdecide.com/pc-game-news/10-best-places-buy-pc-games-online>.
6. What is the best digital distribution platform for PC Games [Electronic resource]. – Access mode: <https://www.slant.co/topics/5430/~digital-distribution-platform-for-pc-games>.
7. Alternatives to Steam For Buying PC Games Online [Electronic resource]. – Access mode: <https://www.online-tech-tips.com/8-alternatives-to-steam-for-buying-pc-games-online/>.
8. Humble Store [Electronic resource]. – Access mode: <https://www.humblebundle.com/store>.
9. Itch [Electronic resource]. – Access mode: <https://itch.io/>.
10. Fanatical [Electronic resource]. – Access mode: <https://www.fanatical.com/en/>.
11. C# Tutorial [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/c-sharp/csharp-programming-language/>.
12. Working with C# [Electronic resource]. – Access mode: <https://code.visualstudio.com/docs/languages/csharp>.

13. Visual Studio [Electronic resource]. – Access mode: <https://visualstudio.microsoft.com/>.

14. VS Code Goes Transparent as Open-Source AI Editor [Electronic resource]. – Access mode: <https://visualstudiomagazine.com/articles/2025/06/30/vs-code-goes-transparent-as-open-source-ai-editor.aspx>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для замовлення комп'ютерних ігор.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація платформи для замовлення комп'ютерних ігор», затверджене наказом Національного університету «Запорізька політехніка» № 447 від 30 вересня 2025 р.

А.2 Призначення розробки

Програмний продукт призначений для замовлення комп'ютерних ігор.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для замовлення комп'ютерних ігор забезпечує виконання таких функцій:

- підтримка можливості відображення переліку комп'ютерних ігор, доступних для завантаження;
- підтримка можливості групування комп'ютерних ігор за обраною користувачем категорією (наприклад, ігри з пошуком предметів, ігри-ребуси, аркадні ігри і т.ін.);
- можливість відображення категорій комп'ютерних ігор;
- можливість авторизації;
- підтримка можливості обчислення та виведення користувачу інформації про кількість комп'ютерних ігор у відповідній категорії;

- підтримка можливості подання інформації про обрану комп'ютерну гру;
- можливість завантаження обраної комп'ютерної гри.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для замовлення комп'ютерних ігор повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для замовлення комп'ютерних ігор повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для замовлення комп'ютерних ігор необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для замовлення комп'ютерних ігор потрібні такі технічні та програмні ресурси: комп'ютерна система повинна бути оснащена процесором з тактовою частотою не нижче 2,5 ГГц (рекомендовано багатоядерний), щоб забезпечити швидке опрацювання запитів і стабільну взаємодію з базою даних. Оперативна пам'ять має становити не менше 8 ГБ, що дозволяє підтримувати одночасну роботу веб-

сервера, бази даних і фонових служб без зниження продуктивності. Для зберігання інформації про товари, замовлення та історію покупок потрібен жорсткий диск або SSD із обсягом вільного простору від 50 ГБ і вище, залежно від масштабів системи та кількості користувачів. З боку програмного середовища необхідно використовувати сучасну операційну систему — наприклад, Windows 10/11 або одна з останніх стабільних версій Linux. Для коректного запуску серверної частини рекомендується встановити .NET Runtime (відповідної версії під розробку), веб-сервер типу IIS або Apache, а також СУБД, сумісну з реалізованою моделлю даних — наприклад, Microsoft SQL Server або PostgreSQL. Для безпечної роботи варто передбачити активну систему захисту й оновлення компонентів. Браузер на стороні клієнта має підтримувати сучасні вебтехнології HTML5, CSS3 та JavaScript для коректного відображення інтерфейсу користувача.

A.3.6 Вимоги до маркування і пакування

Програма для замовлення комп'ютерних ігор може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для замовлення комп'ютерних ігор».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для замовлення комп'ютерних ігор є інформація про обрані користувачем комп'ютерні ігри, категорії ігор та ін.

Вихідними даними програмного забезпечення для замовлення комп'ютерних ігор є сформовані сторінки порталу відповідно до дій користувача.

ДОДАТОК Б
Фрагмент тексту програми

```

namespace PlatfKompShp.Web.Controllers
{
    public class OblczapsController : Controller
    {
        private readonly KrstMan<ApplicationKorstvch> _krstMan;
        private readonly SgnInMangr<ApplicationKorstvch>
        _SgnInMangr;
        private readonly RolMangr<IdentityRol> _RolMangr;
        private readonly Mapr _mapr;
        private readonly PlatfKompShppingKoshk
        _platfKompShppingKoshk;
        private readonly IZamovl _zamovlService;

        public OblczapsController(RolMangr<IdentityRol> RolMangr,
        KrstMan<ApplicationKorstvch> krstMan,
        SgnInMangr<ApplicationKorstvch> SgnInMangr, PlatfKompShppingKoshk
        platfKompShppingKoshk, IZamovl zamovlService)
        {
            _krstMan = krstMan;
            _SgnInMangr = SgnInMangr;
            _RolMangr = RolMangr;
            _mapr = new Mapr();
            _platfKompShppingKoshk = platfKompShppingKoshk;
            _zamovlService = zamovlService;
        }

        [Authorize(Roles = "Admn")]
        public async Zadch<IActionRslft> Index()
        {
            IEnumerable<ApplicationKorstvch> korstvchs =
            _krstMan.Korstvchs;
            var models = await
            _mapr.ApplicationKorstvchsToOblczapsProflModelsAsync(korstvchs,
            _zamovlService, _krstMan);

            var model = new OblczapsIndexModel
            {
                Oblczapss = models
            };

            return View(model);
        }

        public async Zadch<IActionRslft> Search(string srchQry)
        {
            if(string.IsNullOrEmpty(srchQry))
            {

```

```

        return RedirectToAction("Index");
    }

    var kvries = string.IsNullOrEmpty(srchQry) ? null :
    Regex.Replace(srchQry, @"\s+", " ").Trim().ToLower().Split(" ");
    var models = await
    _mapr.ApplicationKorstvchsToOblczapsProflModelsAsync(_krstMan.Kor
    stvchs.Where(korstvch => kvries.Any(Qry => (korstvch.FirstImya +
    " "+ korstvch.LastImya + " "+
    korstvch.Post).ToLower().Contains(Qry))), _zamovlService, _krstMan)
    ;

    var model = new OblczapsIndexModel
    {
        Oblczapss = models,
        SrchQry = srchQry
    };

    return View("Index", model);
}

public async Zadch<IActionRslft> Profl(string korstvchId)
{
    ApplicationKorstvch korstvch;
    if (!string.IsNullOrEmpty(korstvchId) &&
    Korstvch.IsInRol("Admn"))
    {
        korstvch = await
    _krstMan.FindByIdAsync(korstvchId);
    }
    else
    {
        korstvch = await
    _krstMan.GetKorstvchAsync(Korstvch);
    }

    var Rols = await _krstMan.GetRolsAsync(korstvch);

    if (korstvch != null)
    {
        var model =
    _mapr.ApplicationKorstvchToOblczapsProflModel(korstvch,
    _zamovlService, Rols.FirstOrDefault());
        return View(model);
    }

    return RedirectToAction("Index", "Home");
}

```

```

public IActionResult AdmnProfl(string korstvchId)
{
    return RedirectToAction("Profl", new { korstvchId });
}

public IActionResult Lgn(string rtnUrl = "/")
{
    if (_SgnInMangr.IsSignedIn(Korstvch))
    {
        return RedirectToAction("Index", "Home");
    }
    rtnUrl = rtnUrl.Replace("%2F", "/");

    var model = new OblczapsLgnModel
    {
        RtrnUrl = rtnUrl
    };

    return View(model);
}

public async Zadch<IActionResult> Lgn(OblczapsLgnModel
lgn)
{
    if (!MdlState.IsValid)
    {
        return View(lgn);
    }

    var korstvch = await
_krstMan.FindByPostAsync(lgn.Post);

    if (korstvch != null)
    {
        var rslft = await
_SgnInMangr.ParlSgnInAsync(korstvch, lgn.Parl, false, false);
        if (rslft.Usph)
        {
            if (string.IsNullOrEmpty(lgn.RtrnUrl))
            {
                return RedirectToAction("Index", "Home");
            }
            return Redirect(lgn.RtrnUrl);
        }
    }
    MdlState.AddModelError("IncorrectInput",
"Korstvchimya or Parl is incorrect");
}

```

```

        return View(lgn);
    }

    public IActionResult Regstr(string rtnUrl = "/")
    {
        rtnUrl = rtnUrl.Replace("%2F", "/");

        if (_SgnInMangr.IsSignedIn(Korstvch) &&
!Korstvch.IsInRol("Admn"))
        {
            return RedirectToAction("Index", "Home");
        }

        var model = new ObzczapsRegstrModel
        {
            RtnUrl = rtnUrl
        };

        return View(model);
    }

    public async Zadch<IActionResult>
Regstr(ObzczapsRegstrModel regstr)
    {
        var u = await _krstMan.FindByPostAsync(regstr.Post ??
        "");
        if (u != null)
        {
            MdlState.AddModelError("Post", "Post is already
        taken");
        }
        else if (MdlState.IsValid)
        {
            var korstvch =
        _mapr.ObzczapsRegstrModelToApplicationKorstvch(regstr);
            var rslft = await _krstMan.CreateAsync(korstvch,
        regstr.Parl);

            if (rslft.Usph)
            {
                var lgnModel = new ObzczapsLgnModel
                {
                    Post = regstr.Post,
                    Parl = regstr.Parl,
                };
                await _krstMan.AddToRolAsync(korstvch,
        "Customer");
                if(!_SgnInMangr.IsSignedIn(Korstvch) )

```

```

        {
            await
            _SgnInMangr.ParlSgnInAsync(korstvch, registr.Parl, false, false);
        }
        if(!string.IsNullOrEmpty(regstr.RtrnUrl))
        {
            return Redirect(regstr.RtrnUrl);
        }
        return RedirectToAction("Index", "Home");
    }
}

return View(regstr);
}

public async Zadch<IActionRslft> Logout()
{
    if(_SgnInMangr.IsSignedIn(Korstvch))
    {
        await _SgnInMangr.SignOutAsync();
        _platfKompShppingKoshk.ClearKoshk();
    }
    return RedirectToAction("Index", "Home");
}

public async Zadch<IActionRslft> Settings(string
korstvchId)
{
    ApplicationKorstvch korstvch;
    if (!string.IsNullOrEmpty(korstvchId) &&
Korstvch.IsInRol("Admn"))
    {
        korstvch = await
        _krstMan.FindByIdAsync(korstvchId);
        GetRols();
    }
    else
    {
        korstvch = await
        _krstMan.GetKorstvchAsync(Korstvch);
    }
    if (korstvch != null)
    {
        string RolImya = (await
        _krstMan.GetRolsAsync(korstvch)).FirstOrDefault();
        string RolId = _RolMangr.Rols.FirstOrDefault(Rol
=> Rol.Imya == RolImya).Id;

```

```

        var model =
_mapr.ApplicationKorstvchToOblczapsSettingsModel(korstvch,
RolId);
        return View(model);
    }
    return RedirectToAction("Index", "Home");
}

private void GetRols()
{
    var Rols = _RolMangr.Rols.Select(Rol => new
RolDropdownModel
    {
        Id = Rol.Id,
        Imya = Rol.Imya
    });
    ViewBag.Rols = new SelectList(Rols, "Id", "Imya");
}

public async Zadch<IActionRslft>
SaveProfl(OblczapsSettingsModel model)
{
    GetRols();
    if (MdlState.IsValid)
    {
        var korstvch = await
_krstMan.FindByPostAsync(model.Post);

_mapr.OblczapsSettingsModelToApplicationKorstvch(model,
korstvch);

        if (!string.IsNullOrEmpty(model.OldParl)
            || !string.IsNullOrEmpty(model.NewParl)
            ||
!string.IsNullOrEmpty(model.NewParlConfirmation))
        {
            if (string.IsNullOrEmpty(model.OldParl))
            {
                MdlState.AddModelError("OldParl", "Enter
your current parl to change it");
                return View("Settings", model);
            }
            if (string.IsNullOrEmpty(model.NewParl))
            {
                MdlState.AddModelError("NewParl", "Enter
your new parl to change it");
                return View("Settings", model);
            }
        }
    }
}

```

```

        var rslft = await
_krstMan.ChangeParlAsync(korstvch, model.OldParl ?? "",
model.NewParl ?? "");
        if (!rslft.Success)
        {
            ModelState.AddModelError("OldParl",
"Incorrect parl, please enter your current parl to change it");
            return View("Settings", model);
        }
    }

    var Rol = _RolMangr.Rols.First(r => r.Id ==
model.RolId).Imya;
    var korstvchRol = (await
_krstMan.GetRolsAsync(korstvch)).FirstOrDefault();
    if (Rol != korstvchRol)
    {
        await _krstMan.RemoveFromRolAsync(korstvch,
korstvchRol);
        await _krstMan.AddToRolAsync(korstvch, Rol);
    }

    await _krstMan.UpdateAsync(korstvch);

    await HttpContext.RefreshLgnAsync();

    if (Korstvch.IsInRol("Admn"))
    {
        return RedirectToAction("Profl", new {
korstvchId = korstvch.Id });
    }
    return RedirectToAction("Profl");
}
return View("Settings", model);
}

public async Zadch<IActionResult> Deactivate(string
korstvchId)
{
    ApplicationKorstvch korstvch;
    if (!string.IsNullOrEmpty(korstvchId) &&
Korstvch.IsInRol("Admn"))
    {
        korstvch = await
_krstMan.FindByIdAsync(korstvchId);
    }
}

```

```

        else
        {
            korstvch = await
_krstMan.GetKorstvchAsync(Korstvch);
        }

        if(korstvch!=null)
        {
            if(string.IsNullOrEmpty(korstvchId) ||
!Korstvch.IsInRol("Admn"))
            {
                await _SgnInMangr.SignOutAsync();
                _platfKompShppingKoshk.ClearKoshk();
            }
            await _krstMan.DeleteAsync(korstvch);
            return RedirectToAction("Index", "Home");
        }
        return RedirectToAction("Profl");
    }

    public IActionResult AccessDenied()
    {
        return View();
    }

    public IActionResult Cancel(string rtnUrl = "/")
    {
        rtnUrl = rtnUrl.Replace("%2F", "/");

        return Redirect(rtnUrl);
    }
}

namespace PlatfKompShp.Web.Controllers
{
    public class HomeController : Controller
    {
        private readonly IGame _GameSrvc;
        private readonly Mapr _mapr;

        public HomeController(IFood GameSrvc)
        {
            _GameSrvc = GameSrvc;
            _mapr = new Mapr();
        }
    }
}

```

```

        public IActionResult Index()
        {
            var preferredGame = _GameSrvc.GetPreferred(10);
            var model =
_mappr.FoodsToHomeIndexModel(preferredGame);
            return View(model);
        }

        public IActionResult Search(string srchQry)
        {
            if (string.IsNullOrEmpty(srchQry) ||
string.IsNulOrEmpty(srchQry))
            {
                return RedirectToAction("Index");
            }

            var searchedGames =
_GameSrvc.GetFilteredFoods(srchQry);
            var model =
_mappr.FoodsToHomeIndexModel(searchedGames);

            return View(model);
        }
    }
}

```

```

namespace GmShp
{
    public partial class FrmMn : Form
    {
        Db db;
        Usr crntUsr;

        public FrmMn()
        {
            InitializeComponent();
            db = new Db();
            db.LdGms();
            FllLst();
        }

        void FllLst()
        {
            lstGms.Items.Clear();
            foreach (var gm in db.Gms)
            {

```

```

        lstGms.Items.Add($"{gm.Id}: {gm.Ttl} -
        ${gm.Prc}");
    }
}

private void btnRgstr_Click(object sndr, EventArgs rg)
{
    string nm = txtNm.Text.Trim();
    if (nm == "")
    {
        MessageBox.Show("NM NVD");
        return;
    }

    int d = db.Usrs.Count + 1;
    crntUsr = new Usr(d, nm);
    db.Usrs.Add(crntUsr);
    lblStts.Text = $"RGSTRD: {nm}";
}

private void btnPchs_Click(object sndr, EventArgs rg)
{
    if (crntUsr == null)
    {
        MessageBox.Show("FRST RGSTR");
        return;
    }

    if (lstGms.SelectedIndex == -1)
    {
        MessageBox.Show("SLCT GM");
        return;
    }

    int gmId = lstGms.SelectedIndex + 1;
    Gm gm = db.Gms.Find(g => g.Id == gmId);
    crntUsr.Gms.Add(gm);
    MessageBox.Show($"GM {gm.Ttl} PCHSD");
}

private void btnCrt_Click(object sndr, EventArgs rg)
{
    if (crntUsr == null)
    {
        MessageBox.Show("NO USR");
        return;
    }
}

```

```

        lstCrt.Items.Clear();
        double ttl = 0;
        foreach (var gm in crntUsr.Gms)
        {
            lstCrt.Items.Add($"{gm.Ttl} - ${gm.Prc}");
            ttl += gm.Prc;
        }
        lblTtl.Text = $"TTL: ${ttl}";
    }
}

public class Db
{
    public List<Usr> Usrs { get; set; }
    public List<Gm> Gms { get; set; }

    public Db()
    {
        Usrs = new List<Usr>();
        Gms = new List<Gm>();
    }

    public void LdGms()
    {
        for (int x = 1; x <= 10; x++)
        {
            Gms.Add(new Gm(x, $"Gm{x}", x * 5));
        }
    }
}

public class Usr
{
    public int Id { get; set; }
    public string Nm { get; set; }
    public List<Gm> Gms { get; set; }

    public Usr(int d, string nm)
    {
        Id = d;
        Nm = nm;
        Gms = new List<Gm>();
    }
}

public class Gm
{
    public int Id { get; set; }

```

```

        public string Ttl { get; set; }
        public double Prc { get; set; }

        public Gm(int d, string ttl, double prc)
        {
            Id = d;
            Ttl = ttl;
            Prc = prc;
        }
    }
}

```

```

namespace GmShp
{
    public partial class FrmMn : Form
    {
        Db db;
        Usr crntUsr;

        public FrmMn()
        {
            InitializeComponent();
            db = new Db();
            db.LdGms();
            FllLst();
        }

        void FllLst()
        {
            lstGms.Items.Clear();
            foreach (var gm in db.Gms)
            {
                lstGms.Items.Add($"{gm.Id}: {gm.Ttl} -
${gm.Prc}");
            }
        }

        private void btnRgstr_Click(object sndr, EventArgs rg)
        {
            string nm = txtNm.Text.Trim();
            if (nm == "")
            {
                MessageBox.Show("NM NVD");
                return;
            }

            int d = db.Usrs.Count + 1;

```

```

        crntUsr = new Usr(d, nm);
        db.Usrs.Add(crntUsr);
        lblStts.Text = $"RGSTRD: {nm}";
    }

    private void btnPchs_Click(object sndr, EventArgs rg)
    {
        if (crntUsr == null)
        {
            MessageBox.Show("FRST RGSTR");
            return;
        }

        if (lstGms.SelectedIndex == -1)
        {
            MessageBox.Show("SLCT GM");
            return;
        }

        int gmId = lstGms.SelectedIndex + 1;
        Gm gm = db.Gms.Find(g => g.Id == gmId);
        crntUsr.Gms.Add(gm);
        MessageBox.Show($"GM {gm.Ttl} PCHSD");
    }

    private void btnCrt_Click(object sndr, EventArgs rg)
    {
        if (crntUsr == null)
        {
            MessageBox.Show("NO USR");
            return;
        }

        lstCrt.Items.Clear();
        double ttl = 0;
        foreach (var gm in crntUsr.Gms)
        {
            lstCrt.Items.Add($"{gm.Ttl} - ${gm.Prc}");
            ttl += gm.Prc;
        }
        lblTtl.Text = $"TTL: ${ttl}";
    }
}
}
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ПЛАТФОРМИ ДЛЯ ЗАМОВЛЕННЯ
КОМП'ЮТЕРНИХ ІГОР**

Виконав
ст. гр. КНТ-124м

Микола КОЗИР

Керівник
к.т.н., доцент

Михайло КОЦУР

Рисунок В.1 – Слайд 1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

Об'єктом дослідження є процес розробки програмного забезпечення для замовлення комп'ютерних ігор.

Предмет дослідження – програмні засоби для замовлення комп'ютерних ігор.

Мета роботи – дослідження та розробка програмного забезпечення для замовлення комп'ютерних ігор.

Рисунок В.2 – Слайд 2

ЗАВДАННЯ РОБОТИ

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для замовлення комп'ютерних ігор;
- розробити модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор;
- здійснити проектування програмного забезпечення для замовлення комп'ютерних ігор;
- створити програмне забезпечення для замовлення комп'ютерних ігор;
- виконати тестування розробленого програмного забезпечення для замовлення комп'ютерних ігор.

3

Рисунок В.3 – Слайд 3

ПОРІВНЯННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАМОВЛЕННЯ КОМП'ЮТЕРНИХ ІГОР

Критерій порівняння	Humble Store	Itch	Fanatical
Великий вибір комп'ютерних ігор	+	+–	+
Наявність ігор без DRM	+–	+	+–
Зручність інтерфейсу	+	+–	+–
Рівень популярності	+	+–	+–
Офіційні ключі до ігор	+–	+–	+
Різні варіанти підписки	+	+–	–

4

Рисунок В.4 – Слайд 4

ВИБІР МОВИ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	C#	Python	JavaScript
Підтримка процесу розробки застосунків різних типів	+	+	—
Висока швидкодії	+	+—	—
Інтеграція з базами даних	+	+	+—
Зручність для створення ПЗ для замовлення комп'ютерних ігор	+	—	—
Можливість інтеграції з ігровими рушіями	+	+—	—
Наявність офіційних бібліотек та розширень	+	+—	+—

5

Рисунок В.5 – Слайд 5

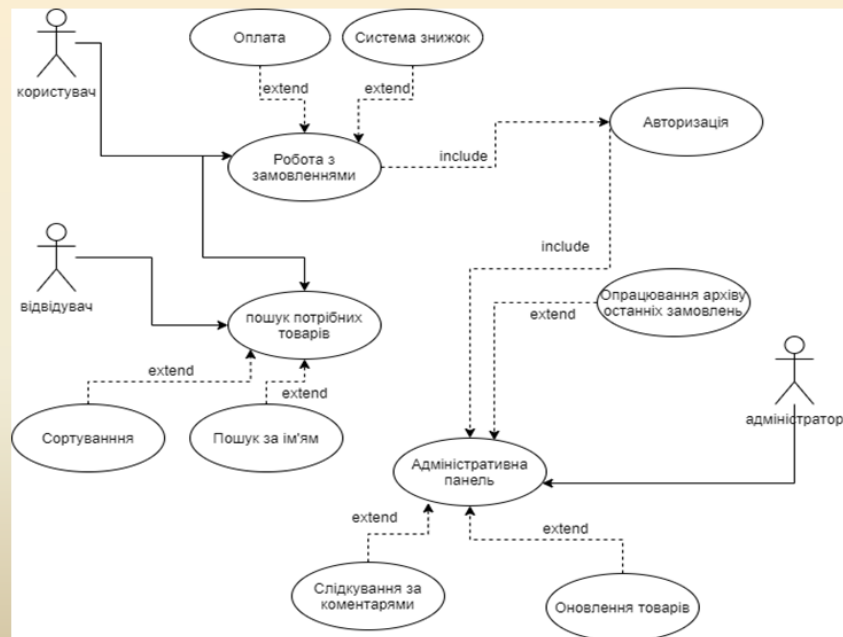
ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	MonoDevelop	JetBrains Rider
Повноцінна підтримка C# та .NET	+	+—	+
Зручність створення інтерфейсу користувача	+	—	+—
Підтримка інструментів для тестування	+	+—	+
Відлагодження та профілювання	+	+—	+
Швидкість запуску та стабільність	+	+—	+—

6

Рисунок В.6 – Слайд 6

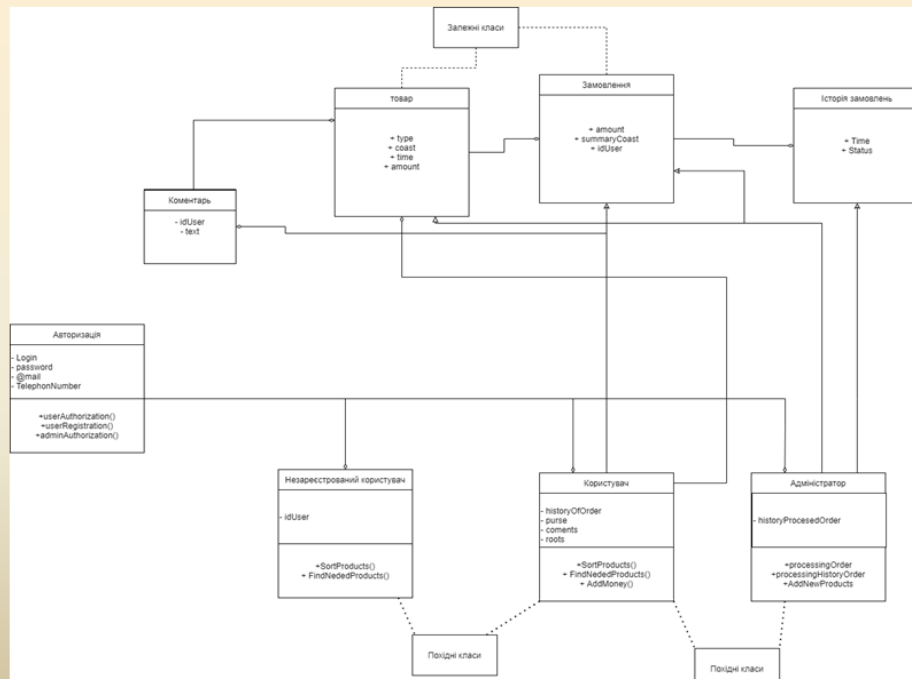
МОДЕЛЬ ВЗАЄМОДІЇ КОРИСТУВАЧА З ПЗ ДЛЯ ЗАМОВЛЕННЯ КОМП'ЮТЕРНИХ ІГОР



7

Рисунок В.7 – Слайд 7

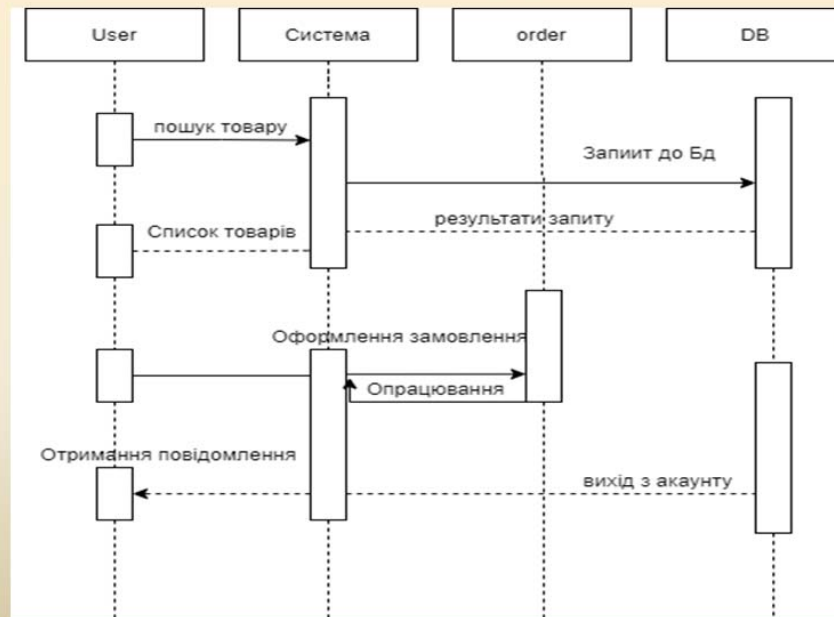
СТРУКТУРА ПЗ ДЛЯ ЗАМОВЛЕННЯ КОМП'ЮТЕРНИХ ІГОР



8

Рисунок В.8 – Слайд 8

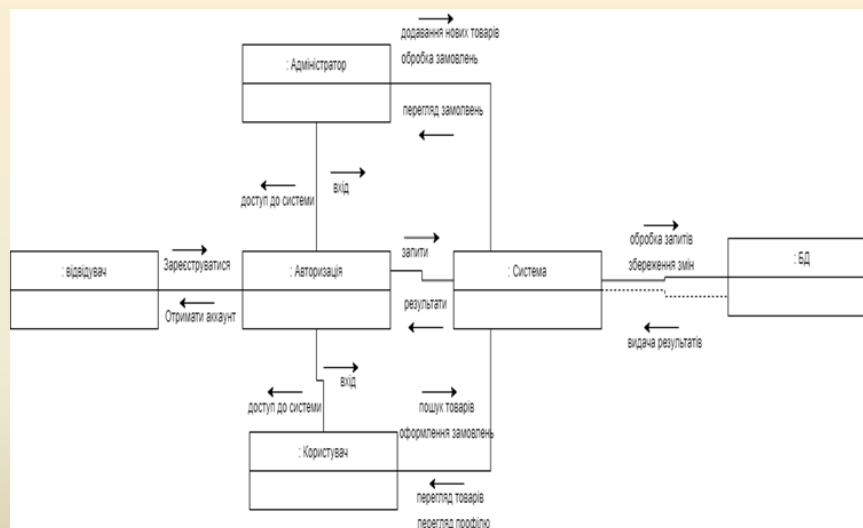
СХЕМА ФУНКЦІОНУВАННЯ ПЗ У ВИГЛЯДІ ДІАГРАМИ ПОСЛІДОВНОСТЕЙ



9

Рисунок В.9 – Слайд 9

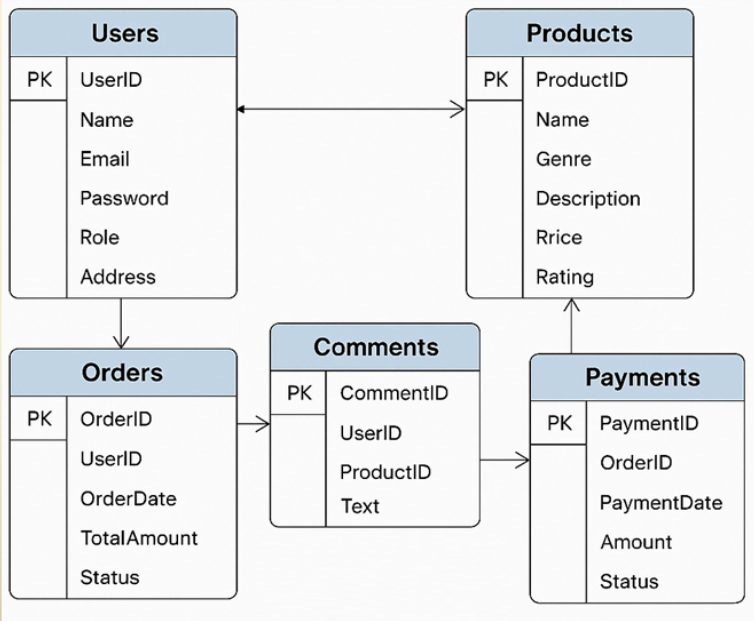
ДІАГРАМИ КООПЕРАЦІЇ ПЗ ДЛЯ ЗАМОВЛЕННЯ КОМП'ЮТЕРНИХ ІГОР



10

Рисунок В.10 – Слайд 10

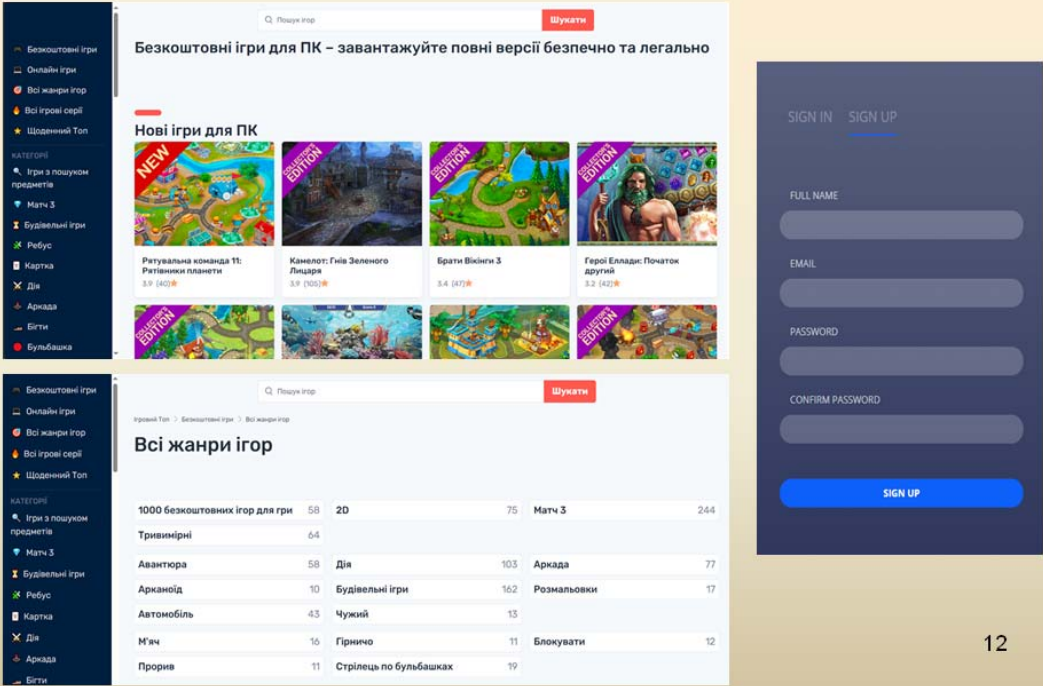
СХЕМА БАЗИ ДАНИХ ПЗ ДЛЯ ЗАМОВЛЕННЯ КОМП'ЮТЕРНИХ ІГОР



11

Рисунок В.11 – Слайд 11

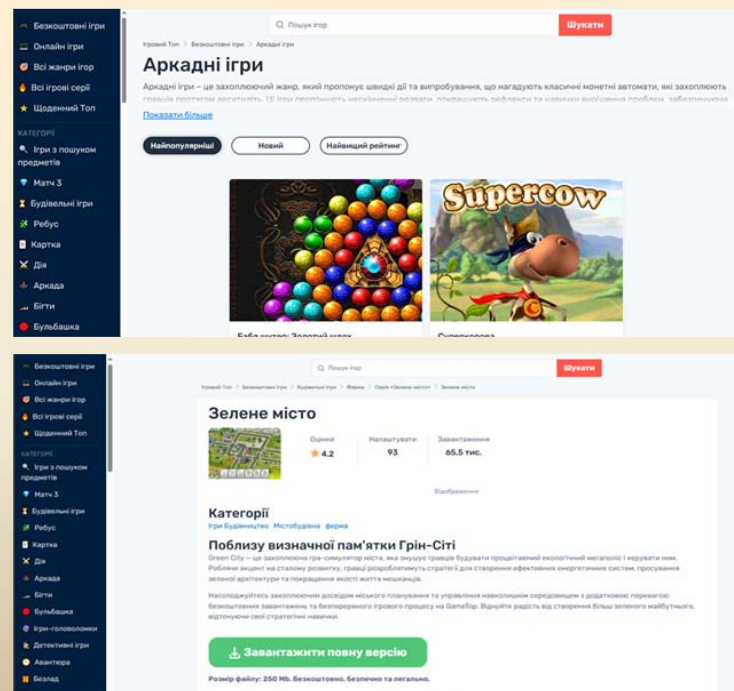
РОБОТА З ПРОГРАМОЮ. ГОЛОВНА СТОРІНКА. ФОРМА АВТОРИЗАЦІЇ. СТОРІНКА ЖАНРІВ ІГОР



12

Рисунок В.12 – Слайд 12

РОБОТА З ПРОГРАМОЮ. ГРУПУВАННЯ ІГОР ЗА КАТЕГОРІЄЮ. СТОРІНКА З ІНФОРМАЦІЄЮ ПРО ГРУ



13

Рисунок В.13 – Слайд 13

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для замовлення комп'ютерних ігор.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для замовлення комп'ютерних ігор, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для замовлення комп'ютерних ігор.

Практичне значення роботи полягає у тому, що створено програмне забезпечення для замовлення комп'ютерних ігор.

Виконано тестування розробленого програмного забезпечення для замовлення комп'ютерних ігор. Після проведення тестування встановлено, що програмного забезпечення для замовлення комп'ютерних ігор відповідає технічним вимогам і може успішно використовуватися для своєї передбаченої мети.

14

Рисунок В.14 – Слайд 14