

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ТОРГОВОГО МАЙДАНЧИКА
ЗАСОБАМИ MAGENTO»

Виконав: студент 4 курсу, групи 6.1210-2пi
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми програмна інженерія
(назва освітньої програми)

А. Бужедді
(ініціали та прізвище)
Керівник завідувач кафедри програмної інженерії,
доцент, к.ф.-м.н. Лісняк А.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної
математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.

(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Бужедді Айману

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка торгового майданчика засобами Magento

керівник роботи Лісняк Андрій Олександрович, к.ф.-м.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі, аналіз предметної області.

2. Проєктування.

3. Реалізація та тестування.

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

презентація за темою доповіді

6. Консультанти розділів роботи

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

7. Дата видачі завдання 25.12.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	26.01.2024	
2.	Збір вихідних даних.	16.02.2024	
3.	Обробка методичних та теоретичних джерел.	15.03.2024	
4.	Розробка першого та другого розділу.	12.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	20.06.2024	

Студент _____
(підпис)

А. Бужедді
(ініціали та прізвище)

Керівник роботи _____
(підпис)

А.О. Лісняк
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка торгового майданчика засобами Magento»: 60 с., 36 рис., 1 табл., 13 джерел, 3 додатки.

CMS, DATABASE, ELASTIC SEARCH, MAGENTO, MVC, PHP, UML.

Об'єкт дослідження – система, торговий майданчик, інструменти для електронної комерції.

Мета роботи – розробити торговий майданчик.

Методи дослідження – моделювання, проектування, програмний, аналітичний.

У роботі розглядається процес реалізації та тестування системи електронної комерції на базі CMS Magento та пошукового сервера Elasticsearch. Описано основні інструменти розробки, структуру системи та ключові класи, пов'язані з оформленням замовлень. Наведено деталі підготовки та налаштування оточення для розгортання системи.

Проаналізовано роль UML у процесі розробки, представлено діаграми варіантів використання, діяльності, послідовності та розгортання. Розглянуто питання тестування системи, зокрема unit-тестування з використанням фреймворку PHPUnit. Також надано керівництво користувача для ефективної взаємодії з системою.

Таким чином, розроблено торговий майданчик, що надає зручний інтерфейс та необхідний функціонал для використання можливостей електронної комерції.

SUMMARY

Bachelor's qualifying paper "Development of a Trading Platform Using Magento": 60 pages, 36 figures, 1 table, 13 references, 3 supplements.

CMS, DATABASE, ELASTIC SEARCH, MAGENTO, MVC, PHP, UML.

The object of the study is the system, trading platform, e-commerce tools.

The aim of the study is to develop a trading platform.

The methods of research are modeling, design, programming, analytical.

The paper describes the process of implementing and testing an e-commerce system based on Magento CMS and Elasticsearch search engine. It outlines the main development tools, system structure, and key classes related to ordering. It also provides details of preparing and configuring the environment for deploying the system.

The role of UML in the development process is analyzed, and diagrams of use cases, activities, sequence, and deployment are presented. The paper also considers the issues of system testing, in particular unit testing using the PHPUnit framework. In addition, a user guide for effective interaction with the system is provided.

The result of this process is a trading platform that provides a user-friendly interface and the necessary functionality to use e-commerce opportunities.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	8
1 Технічне завдання	10
1.1 Терміни та визначення.....	10
1.1.1 Загальні терміни	10
1.1.2 Технічні терміни	10
1.2 Функціональні вимоги.....	11
1.2.1 Призначення і цілі створення системи	11
1.2.2 Загальні функціональні можливості системи	11
1.3 Нефункціональні вимоги.....	12
1.3.1 Інтерфейс користувача	12
1.3.2 Підтримка браузерів	12
1.3.3 Вимоги до продуктивності.....	13
1.3.4 Вимоги до безпеки.....	13
1.4 Опис предметної області	13
1.5 Опис системи	14
1.6 Огляд та порівняння аналогів	15
2 Проєктування.....	18
2.1 Використання UML під час розробки системи.....	18
2.2 Діаграма варіантів використання	19
2.2.1 Опис варіантів використання.....	23
2.3 Діаграма діяльності.....	30
2.4 Діаграма послідовності.....	33
2.5 Діаграма розгортання.....	35
3 Реалізація та тестування	37

3.1	Опис інструментів розробки	37
3.2	Основні класи системи	37
3.3	Підготовка та налаштування оточення	38
3.4	Тестування проєкту	42
3.5	Керівництво користувача	43
3.5.1	Рівень підготовки користувача	43
3.5.2	Реєстрація/вхід	44
3.5.3	Процес оформлення замовлення	46
	Висновки	53
	Перелік посилань	54
	Додаток А Файл оточення	56
	Додаток Б Тести	58
	Додаток В Посилання на Git	60

ВСТУП

Електронна комерція стала невід’ємною частиною сучасного бізнес-середовища. Щодня мільйони клієнтів по всьому світу здійснюють покупки в інтернет-магазинах, віддаючи перевагу зручності та широкому вибору товарів. Однак створення власного інтернет-магазину може бути складним завданням, особливо для малого та середнього бізнесу з обмеженими ресурсами. Існуючі платформи електронної комерції часто є вузькоспеціалізованими або занадто складними для налаштування, що створює бар’єри для входу на ринок.

Підприємці шукають гнучкі та масштабовані рішення, які дозволять їм легко створювати та керувати інтернет-магазинами, адаптуючись до мінливих вимог ринку. Magento, одна з провідних платформ електронної комерції з відкритим вихідним кодом, пропонує потужні інструменти для створення торгових майданчиків з широким спектром функцій та можливостей кастомізації.

Виходячи з цього, було вирішено створити торговий майданчик, який дозволяв би гнучко налаштовувати інтернет-магазин та робив процес покупки простим та зрозумілим.

Актуальність дослідження: актуальність теми зумовлена потребою необхідності розробки ефективних та гнучких рішень для створення інтернет-магазинів, які забезпечують масштабованість, безпеку та зручність як для продавців, так і для покупців.

З огляду на це, можна виділити наступні цілі і задачі нашого дослідження.

Мета: розробити торговий майданчик з використанням можливостей платформи Magento.

Задачі:

- сформулювати вимоги до торгового майданчику;
- спроектувати та побудувати архітектуру майданчику;
- реалізувати торговий майданчик;

– протестувати роботу майданчику.

Об'єкт дослідження: процес розробки торгового майданчику, інструменти для ведення електронної комерції.

Предмет дослідження: CMS Magento 2.

Методи дослідження: моделювання, проєктування, програмний, аналітичний.

Перший розділ присвячено збору та аналізуванню вимог до торгового майданчику, огляду подібних систем і опису системи.

У другому розділі розглянуто етапи проєктування торгового майданчику, наведено опис прецедентів та побудовано UML-діаграми.

Третій розділ присвячено реалізації та тестуванню роботи торгового майданчику, наведено покрокове керівництво користувача, яке описує процес роботи з майданчиком.

1 ТЕХНІЧНЕ ЗАВДАННЯ

1.1 Терміни та визначення

1.1.1 Загальні терміни

Система – торговий майданчик створений на основі CMS Magento.

Magento – система керування вмістом з відкритим кодом, призначена для створення інтернет-магазинів.

ВВ – Варіант Використання чи Use Case.

ДВІ – Діаграма Варіантів Використання чи Use Case Diagram.

ДП – Діаграма Послідовності.

ДД – Діаграма Діяльності.

ДР – Діаграма Розгортання.

Відвідувач – не зареєстрований або неавторизований користувач системи.

Зареєстрований користувач – користувач, який авторизувався в системі.

Адміністратор – користувач, який має права на керування системою.

1.1.2 Технічні терміни

БД – база даних, місце збереження інформації системи.

CMS – програмне забезпечення для організації вебсайтів чи інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах.

Модуль – структурна одиниця Magento.

1.2 Функціональні вимоги

1.2.1 Призначення і цілі створення системи

Функціональне призначення системи – реалізувати функціонал торгового майданчику засобами CMS Magento.

Експлуатаційне призначення системи – система може експлуатуватися користувачами на різних рівнях, залежно від наданих прав.

Мета створення системи – розробка торгового майданчика засобами CMS Magento.

1.2.2 Загальні функціональні можливості системи

Система має надавати відвідувачам такі можливості:

- реєстрація в системі;
- вхід до системи;
- перегляд каталогу товарів;
- перегляд товару;
- додавання/видалення товару до/з кошика;
- вказання кількості одиниць товару;
- оформлення замовлення;
- введення email;
- вказання платіжних даних;
- вказання даних для доставки;
- вибір платіжної системи;
- вибір способу доставки;
- пошук/фільтрація товарів.

Система має надавати зареєстрованим користувачам такі додаткові можливості:

- вихід з системи;
- керування профілем;
- перегляд історії замовлень;
- формування списку бажань.

Система має надавати адміністраторам повні права на взаємодію з системою, серед основних можна виділити наступні:

- керування замовленнями (перегляд/ видалення/редагування/зміна статусу);
- керування категоріями товарів;
- керування товарами.

1.3 Нефункціональні вимоги

1.3.1 Інтерфейс користувача

Інтерфейс системи має бути адаптованим, забезпечуючи зручне і ефективно відображення на різних пристроях, включаючи настільні комп'ютери, ноутбуки, планшети та мобільні телефони. Це забезпечує зручність використання для всіх категорій користувачів, від відвідувачів до адміністраторів.

1.3.2 Підтримка браузерів

Система має підтримувати останні версії основних веббраузерів, зокрема Mozilla Firefox, Google Chrome, Safari, Microsoft Edge та Opera. Це забезпечить, що користувачі можуть ефективно взаємодіяти з системою незалежно від браузера.

1.3.3 Вимоги до продуктивності

Система повинна забезпечувати швидке завантаження сторінок (не більше ніж за 1 секунду) та ефективну обробку великих об'ємів даних, включаючи завантаження та візуалізацію великих об'ємів даних, не впливаючи на загальну продуктивність.

1.3.4 Вимоги до безпеки

Система не повинна надавати доступ до функцій керування та особистої інформації для неавторизованих користувачів, захищаючи дані та функціональність від несанкціонованого використання.

Система не повинна без згоди користувача запам'ятовувати та використовувати платіжні дані.

1.4 Опис предметної області

Предметною областю є розробка торгового майданчика. Система має надавати користувачам можливість перегляду каталогу товарів, вибору потрібних продуктів, додавання їх до кошика та здійснення покупки. Взаємодія з системою залежить від статусу користувача: відвідувачі можуть переглядати та замовляти товари, тоді як зареєстровані користувачі мають додаткові можливості, такі як керування профілем, збереження інформації про доставку та платіжні дані, а також перегляд історії замовлень.

Процес оформлення замовлення відбувається наступним чином.

Користувач має можливість зареєструватися або увійти до свого акаунта для забезпечення синхронізації даних та історії покупок. Це не є обов'язковою умовою але надає можливість моніторингу процесу опрацювання замовлення.

Далі користувач переходить до каталога, вибирає бажані товари та додає їх до кошика. Система дозволяє детально переглянути опис товарів, фотографії та характеристики. Після вибору необхідних товарів, користувач переходить до кошика. У ньому він має можливість змінити кількість товарів або видалити непотрібні, а також переглянути загальну вартість замовлення.

Якщо користувача влаштовує вартість, то він вводить email, дані для доставки (ПШБ, адреса), обирає спосіб доставки та оплати. Система підтримує різні методи оплати та доставки, забезпечуючи гнучкість у виборі. Після заповнення форми, користувач натискає кнопку «Оформити замовлення», система обробляє замовлення та зберігає всю інформацію в базі даних.

Користувач отримує електронний лист з підтвердженням замовлення та інформацією для стеження за доставкою. Також користувач може переглянути деталі замовлення у своєму профілі на сайті.

1.5 Опис системи

Торговий онлайн-майданчик дозволяє компаніям та індивідуальним продавцям вести комерцію через інтернет, надаючи платформу для продажу товарів та послуг широкому колу споживачів. Така система забезпечує зручний інтерфейс як для кінцевих користувачів, так і для адміністраторів, що включає управління товарами, обробку замовлень, інтеграцію з платіжними системами, логістику доставки, а також підтримку клієнтів.

Майданчики онлайн-торгівлі повинні бути масштабованими та адаптивними, здатними підтримувати високу навантаженість і забезпечувати високу доступність сервісів для користувачів з різних регіонів. Забезпечення інтуїтивно зрозумілого користувацького інтерфейсу, надійності операцій та високої швидкості обробки даних є ключовими для успішної експлуатації таких систем.

Саме такі можливості і функціонал забезпечує CMS Magento, на платформі якої створено понад 100 тисяч інтернет-магазинів.

1.6 Огляд та порівняння аналогів

Щоб детально зрозуміти переваги торговельного майданчику на платформі Magento порівняно з іншими платформами для онлайн-торгівлі, розглянемо декілька популярних систем: Shopify, WooCommerce і BigCommerce.

Shopify є однією з найбільш популярних платформ для створення онлайн-магазинів завдяки своїй інтуїтивно зрозумілій панелі управління та швидкості налаштування. Користувачам доступний широкий вибір тем та додатків для адаптації функціоналу, а також вбудовані платіжні рішення і SSL-сертифікати для захисту транзакцій. Проте, ця платформа має свої обмеження, включаючи менший контроль над технічними аспектами та кодом магазину. Також вартість підписки і транзакційні збори можуть бути вищими порівняно з іншими рішеннями [8].

WooCommerce відрізняється своєю гнучкістю і можливістю повного контролю над вебсайтом, що є особливо важливим для користувачів WordPress. Платформа пропонує великий вибір плагінів і тем, які можна використовувати для розширення функціоналу магазину, і є доступна у базовому варіанті безкоштовно, хоча додаткові плагіни можуть вимагати платежі. Однак, WooCommerce вимагає значних технічних знань для оптимізації і налаштування, а також залежить від WordPress, що може впливати на безпеку і продуктивність магазину [12].

BigCommerce надає широкий спектр вбудованих функцій, що дозволяє користувачам управляти великими каталогами продуктів і підтримувати високі обсяги продажів без потреби встановлювати додаткові плагіни. Платформа підтримує масштабування бізнесу без обмежень на обсяг транзакцій. Недоліки включають вищу вартість підписки в порівнянні з іншими платформами і меншу гнучкість в налаштуванні інтерфейсу, що може бути критично для деяких користувачів, які шукають специфічні рішення для свого магазину [4].

Проаналізувавши переваги та недоліки інших платформ, можемо порівняти їх з торговим майданчиком на базі Magento (табл. 1.1).

Таблиця 1.1 – Порівняння аналогів

Критерій	Magento 2	Shopify	WooCommerce	BigCommerce
Головна перевага	Велика гнучкість і масштабованість	Легкість використання і налаштування	Повний контроль і гнучкість на базі WordPress	Сильна підтримка великих каталогів без додаткових плагінів
Налаштування інтерфейсу	Повна кастомізація	Обмежена кастомізація	Повна кастомізація	Обмежена кастомізація
Вартість	Змінна, залежно від використаних рішень	Від середньої до високої, залежно від плану	Низька базова вартість, зростає з додаванням плагінів	Висока, особливо для великих магазинів
Технічна складність	Висока	Низька	Середня	Середня
Спільнота та підтримка	Велика спільнота розробників	Широка користувацька база і підтримка	Велика спільнота WordPress	Активна спільнота і корпоративна підтримка
Масштабованість	Висока для великих і середніх бізнесів	Підходить для малих і середніх бізнесів	Залежить від хостингу WordPress	Оптимізована для великих об'ємів продажів
Безпека	Високий рівень захисту з можливістю розширення	Вбудовані функції безпеки	Залежить від зовнішніх плагінів та хостингу	Високий стандарт безпеки

Ця таблиця надає чіткий огляд основних аспектів кожної платформи, що дозволяє зробити інформований вибір на користь однієї з них, в залежності від потреб бізнесу. Magento виокремлюється серед інших платформ завдяки своїй високій масштабованості та гнучкості. Вона ідеально підходить для середніх та великих підприємств, які планують розширення та потребують повного контролю над своїми онлайн-операціями.

2 ПРОЄКТУВАННЯ

2.1 Використання UML під час розробки системи

Unified Modeling Language (UML) є стандартизованою мовою моделювання, що застосовується в процесах розробки програмного забезпечення для візуалізації, специфікації, конструювання та документації архітектури систем. Впровадження UML сприяє формуванню глибокого розуміння структурних та поведінкових аспектів системи серед учасників проєкту.

На етапі формулювання вимог, UML надає точне визначення функціональних і нефункціональних вимог через діаграми варіантів використання. Такі діаграми забезпечують інтуїтивно зрозуміле зображення взаємодії користувачів з системою, що є критично важливим для валідації вимог замовника [6].

У фазі проєктування архітектури, діаграми класів і композиції UML дозволяють детально специфікувати структуру даних і взаємозв'язки між компонентами системи. Діаграми послідовностей і співпраці використовуються для моделювання взаємодії між об'єктами в часі, що є незамінним для розробки логіки внутрішніх процесів [10].

Діаграми діяльні у UML відіграють значну роль у проєктуванні інтерфейсів користувача, адже вони ілюструють потоки управління даними та навігаційні сценарії, сприяючи тим самим створенню ефективних та інтуїтивно зрозумілих користувацьких інтерфейсів.

На завершальних стадіях розробки, включно з інтеграцією та тестуванням, компонентні діаграми UML допомагають аналізувати взаємодії та залежності між різними модулями системи. Це сприяє виявленню і вирішенню інтеграційних конфліктів та помилок [10].

Крім того, UML є ефективним засобом для документації проєкту, забезпечуючи необхідний рівень деталізації та зрозумілості для подальшого обслуговування, адаптації та розширення системи. Стандартизація документації

за допомогою UML сприяє легшій передачі знань всередині команди та ефективній колаборації.

Отже, використання Unified Modeling Language в процесах розробки програмного забезпечення сприяє формуванню зрозумілої, ефективної та високофункціональної архітектури, що є важливим для забезпечення якості та відповідності системи вимогам користувачів та бізнесу.

2.2 Діаграма варіантів використання

Діаграма варіантів використання в Unified Modeling Language (UML) є інструментом для візуалізації функціональних вимог системи, що демонструє взаємодії між користувачами або зовнішніми системами, відомими як актори, та самою системою. Ці актори можуть бути людьми, іншими системами, або навіть часовими подіями, які ініціюють певні дії в системі. Актори з'єднані з варіантами використання, які представляють функціональні можливості або цілі, що система виконує на користь акторів. Ці варіанти використання зображені у формі овалів і позначені назвами, що описують дії [6].

Зв'язки між акторами та варіантами використання вказують на те, як актори можуть взаємодіяти з системою. Ці зв'язки можуть бути прямими, що свідчить про безпосередню участь акторів у варіантах використання, або косвеними, коли взаємодія відбувається через інші механізми або в середовищі системи.

Для моделювання більш складних взаємодій, UML надає механізми, такі як загальні використання і розширення. Загальні використання дозволяють вказати варіанти використання, які необхідно включити в інший варіант використання, часто застосовуються для відображення повторного використання функціональності. Розширення, з іншого боку, використовуються для додавання додаткової функціональності до варіанту використання за певних умов, що дозволяє моделювати необов'язкову або умовну поведінку [6, 10].

Умовні зв'язки, такі як генералізації, можуть застосовуватися для моделювання спадщини між акторами або варіантами використання, вказуючи на загальний клас, який спеціалізується на більш конкретних ролях або функціях. Це дозволяє більш гнучко адаптувати систему до різноманітних сценаріїв використання і ролей користувачів.

Отже, діаграма варіантів використання є основним інструментом в процесі проектування систем, забезпечуючи ясне і зрозуміле представлення функціональних можливостей системи і взаємодій з її користувачами, що є ключовим для успішного аналізу і реалізації проєкту.

На рисунку 2.1 представлена діаграма варіантів використання, яка демонструє функціональні можливості користувачів головного бізнес-процесу «Оформити замовлення», а також суміжних і допоміжних з ним процесів.

На діаграмі представлено акторів «Відвідувач», «Авторизований користувач» та «Адміністратор», кожен з яких може взаємодіяти з системою на різних рівнях. Також представлений актор «Користувач», який включає в себе спільні функціональні можливості відвідувача та авторизованого користувача.

Виділено 1 основний варіант використання – «Оформлення замовлення». Користувач переходить до каталогу товарів, вибирає бажані товари та додає їх до кошика. Система відображає детальну інформацію про товар: опис, фотографії, характеристики, тощо. Після вибору необхідних товарів, користувач переходить до кошика. У ньому він має можливість змінити кількість товарів або видалити непотрібні, а також побачити загальну вартість замовлення.

Далі користувач переходить до інтерфейсу оформлення замовлення, де вводить email, дані для доставки (ПІБ, адреса), обирає спосіб доставки та оплати. Якщо користувач авторизувався, то більшість даних можна буде підтягнути з профіля. Після заповнення форми, користувач натискає кнопку «Place Order», система обробляє замовлення та зберігає всю інформацію в базі даних.

Користувач отримує електронний лист з підтвердженням замовлення та інформацією для стеження за доставкою. Також користувач може переглянути деталі замовлення в профілі.

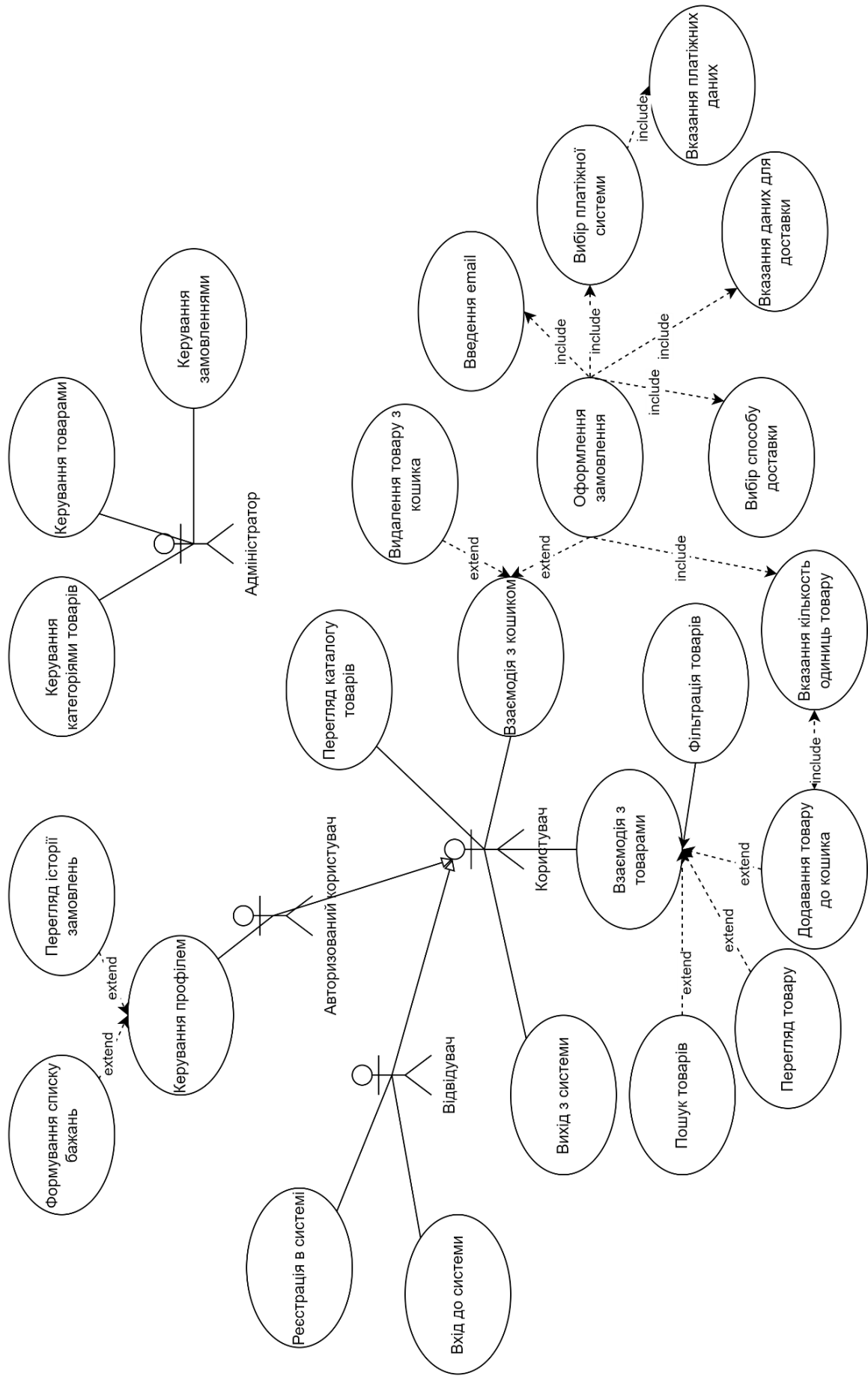


Рисунок 2.1 – Діаграма варіантів використання

Варіанти використання описують, як кінцеві користувачі або зовнішні актори можуть використовувати функціональні можливості системи. Кожен такий варіант представляє детальну послідовність дій, які актор виконує для досягнення певної мети або результату під час взаємодії з системою. Ці варіанти використання є ключовими для визначення та документування вимог до функціональності програмної системи та описують різноманітні шляхи, якими користувачі або зовнішні сутності взаємодіють з системою для досягнення своїх цілей.

Кожен варіант використання відтворює конкретний сценарій, який складається з кроків або дій, необхідних актору для реалізації цільової діяльності в системі. Ці сценарії відображають різні методи застосування системних можливостей акторами для вирішення їхніх завдань чи задоволення потреб. Таким чином, варіанти використання забезпечують ефективний механізм для моделювання та документації поведінкових вимог системи з точки зору її користувачів та зовнішніх сутностей, підкреслюючи різні сценарії взаємодії акторів з відповідним функціоналом системи (див. рис. 2.2 – 2.3).

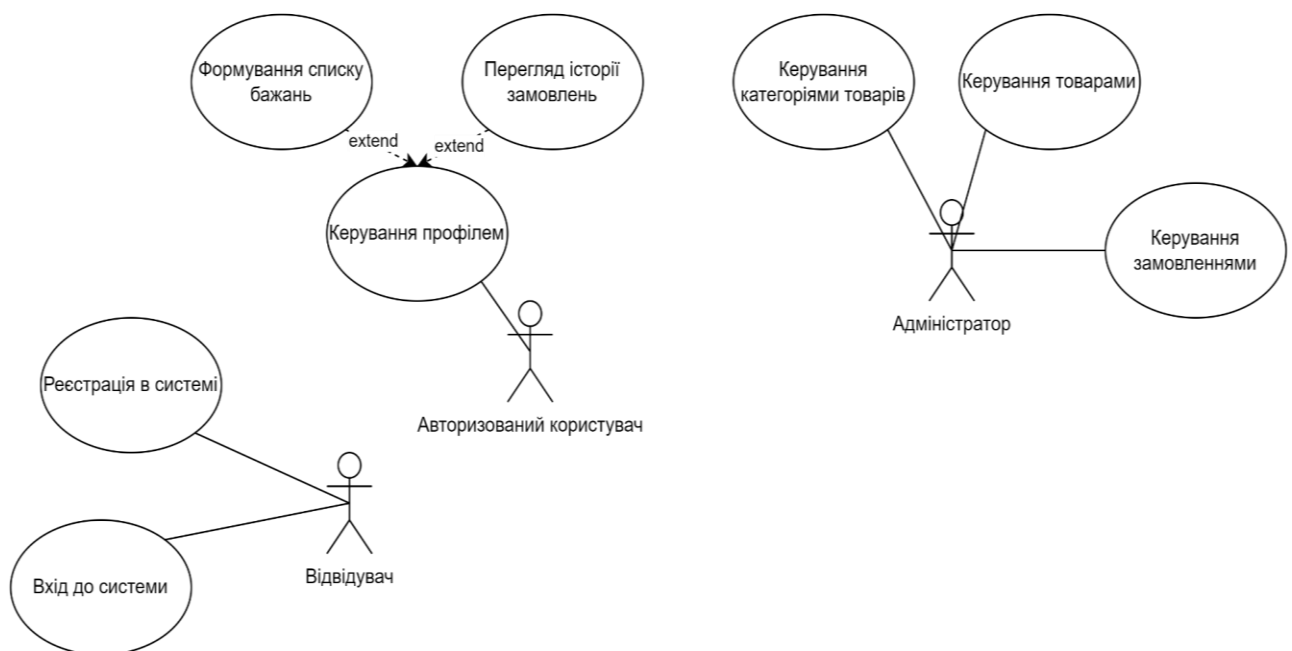


Рисунок 2.2 – Діаграма варіантів використання акторів «Відвідувач», «Авторизований користувач» та «Адміністратор»

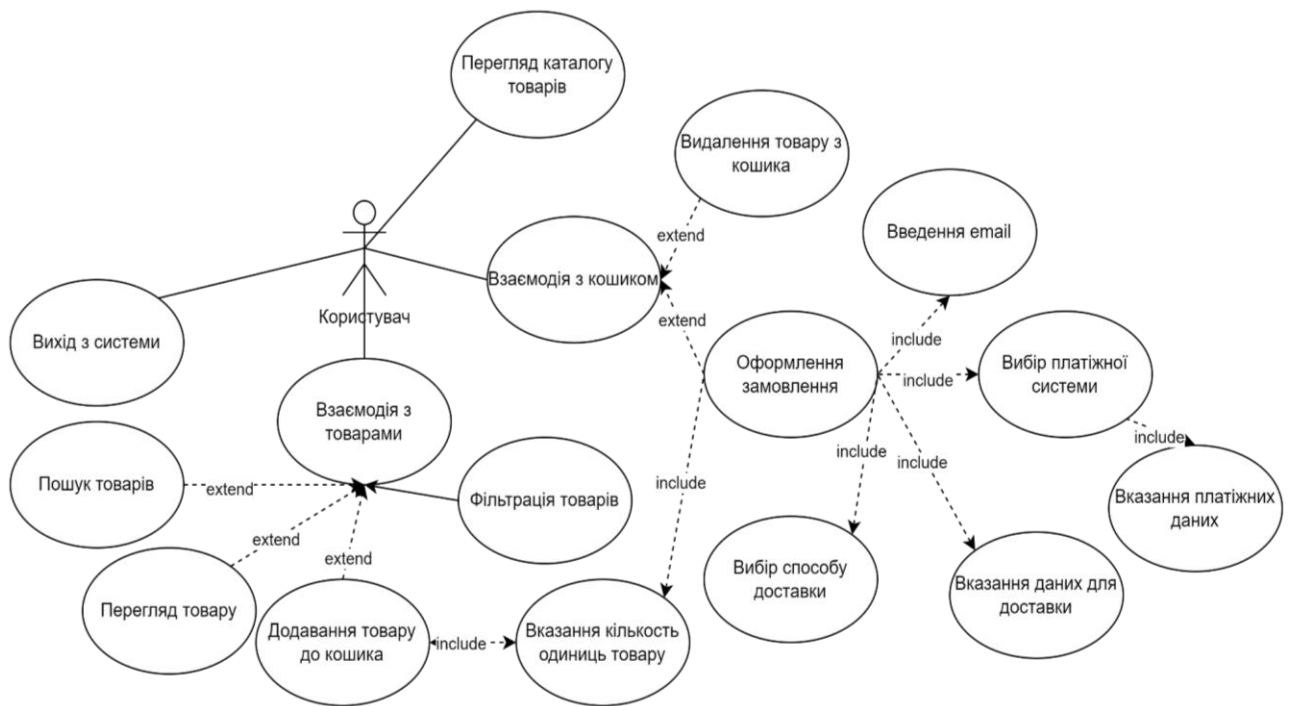


Рисунок 2.3 – Діаграма варіантів використання «Користувач»

2.2.1 Опис варіантів використання

Прецедент «Реєстрація в системі».

Призначення: даний варіант використання надає можливість відвідувачу зареєструватися в системі.

Основний потік подій: даний варіант використання починає виконуватися, коли відвідувач натискає на пункт меню «Create an Account». Система відображає форму реєстрації. Після введення необхідних даних, відвідувач натискає на кнопку «Create an Account», система створює нового користувач та зберігає його дані.

Виняткова ситуація 1: невалідні дані – система відображає повідомлення про помилку, відвідувач може повторно ввести дані.

Виняткова ситуація 2: такий користувач вже зареєстрований – система відображає повідомлення про помилку, відвідувач може змінити дані.

Виняткова ситуація 3: помилка сервера – система відображає відповідне повідомлення.

Прецедент «Вхід до системи».

Призначення: даний варіант використання надає можливість відвідувачу авторизуватися в системі.

Основний потік подій: даний варіант використання починає виконуватися, коли відвідувач натискає на пункт меню «Sign In». Система відображає форму входу до системи. Після введення пошти та паролю, відвідувач натискає на кнопку «Sign In», система відображає сторінку профіля користувача.

Передумова: перед початком виконання даного варіанта використання відвідувач повинен зареєструватися в системі.

Вияткова ситуація 1: невалідні дані – система відображає повідомлення про помилку, відвідувач може повторно ввести дані.

Вияткова ситуація 2: такого користувача не існує – система відображає повідомлення про помилку, відвідувач може змінити дані.

Вияткова ситуація 3: помилка сервера – система відображає відповідне повідомлення.

Прецедент «Вихід з системи».

Призначення: даний варіант використання надає можливість авторизованому користувачу виходити з системи.

Основний потік подій: даний варіант використання починає виконуватися, коли авторизованому користувач натискає на ім'я профіля та обирає пункт меню «Logout». Система видаляє дані про сесію користувача та відображає головну сторінку.

Прецедент «Перегляд каталогу товарів».

Призначення: даний варіант використання надає можливість користувачу переглядати каталог товарів.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач натискає на будь-який пункт навігаційного меню, що пов'язаний з товаром. Система відображає сторінку з товарами згідно обраній категорії.

Прецедент «Взаємодія з товарами».

Призначення: даний варіант використання надає можливість користувачу взаємодіяти з товарами.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач або натискає на будь-який елемент на картці товару, або фільтрує/сортуює товари. Система реагує залежно від обраної дії.

Прецедент «Перегляд товару».

Призначення: даний варіант використання надає можливість користувачу переглядати картку товару.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач натискає на назву товару. Система відкриває сторінку обраного товару. Користувач може ознайомитися з описом, а також взаємодіяти з товаром.

Передумова: перед початком виконання даного варіанта використання користувач повинен знаходитися на сторінці каталогу товарів.

Виняткова ситуація 1: помилка сервера – система відображає відповідне повідомлення та не відкриває сторінку товару.

Прецедент «Фільтрація товарів».

Призначення: даний варіант використання надає можливість користувачу фільтрувати товари за характеристиками.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач обирає один із фільтрів лівобічної панелі. Система відображає сторінку з товарами, які відповідають умовам фільтрації.

Передумова: перед початком виконання даного варіанта використання користувач повинен знаходитися на сторінці каталогу товарів.

Виняткова ситуація 1: помилка сервера – система відображає відповідне повідомлення.

Прецедент «Пошук товарів».

Призначення: даний варіант використання надає можливість користувачу шукати товари за назвою.

Основний потік подій: даний варіант використання починає виконуватися,

коли користувач вводить повну або часткову назву товару в поле пошуку та натискає «Enter» на клавіатурі. Система відображає товари, які відповідають критеріям пошуку.

Вияткова ситуація 1: помилка сервера – система відображає відповідне повідомлення.

Прецедент «Додавання товару до кошика».

Призначення: даний варіант використання надає можливість користувачу додавати обрані товари до кошика.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач обирає характеристики товару, вводить кількість і натискає на кнопку «Add to Cart» в картці або на сторінці товару. Система додає товар до кошика та відображає відповідне повідомлення.

Передумова: перед початком виконання даного варіанта використання користувач повинен знаходитися на сторінці каталогу товарів або на сторінці товару.

Вияткова ситуація 1: товару немає в наявності – система відображає відповідне повідомлення, товар не додається до кошика.

Вияткова ситуація 2: помилка сервера – система відображає відповідне повідомлення, товар не додається до кошика.

Прецедент «Вказання кількості одиниць товару».

Призначення: даний варіант використання надає можливість користувачу вказати кількість одиниць товару для додання до кошика.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач вводить значення в поле «Qty». Система запам'ятовує кількість товару, яку необхідно додати до кошика. Якщо користувач змінює кількість товару з інтерфейсу кошика, то система автоматично перераховує суму замовлення.

Передумова: перед початком виконання даного варіанта використання користувач повинен знаходитися на сторінці каталогу товарів або на сторінці товару.

Прецедент «Взаємодія з кошиком».

Призначення: даний варіант використання надає можливість користувачу взаємодіяти з кошиком.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач натискає на піктограму кошика в верхньому правому кутку інтерфейсу та обирає пункт «View and Edit Cart». Система відображає інтерфейс кошика.

Передумова: перед початком виконання даного варіанта використання користувач повинен додати як мінімум один товар до кошика.

Виняткова ситуація 1: жодного товару не було додано – система відображає повідомлення про те, що жодного товару ще не було додано.

Виняткова ситуація 2: помилка сервера – система відображає відповідне повідомлення.

Прецедент «Видалення товару з кошика».

Призначення: даний варіант використання надає можливість користувачу видаляти товари з кошика.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач натискає на піктограму смітника в картці відповідного товару. Система видаляє товар з кошика та перераховує суму замовлення.

Передумова: перед початком виконання даного варіанта використання користувач повинен додати як мінімум один товар до кошика.

Виняткова ситуація 1: жодного товару не було додано – система відображає повідомлення про те, що жодного товару ще не було додано.

Виняткова ситуація 2: помилка сервера – система відображає відповідне повідомлення, товар не видаляється.

Прецедент «Оформлення замовлення».

Призначення: даний варіант використання надає можливість користувачу оформити замовлення.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач натискає на кнопку «Proceed to Checkout». Система відображає

інтерфейс оформлення замовлення, де користувач вводить email, дані для доставки (ПІБ, адреса), обирає спосіб доставки та оплати. Якщо користувач авторизувався, то більшість даних можна буде підтягнути з профіля. Після заповнення форми, користувач натискає кнопку «Place Order», система обробляє замовлення та зберігає всю інформацію до БД. Користувач, на вказану при оформленні замовлення електронну адресу, отримує електронний лист з підтвердженням замовлення та інформацією для відстеження доставки товару.

Передумова: перед початком виконання даного варіанта використання користувач повинен додати як мінімум один товар до кошика.

Виняткова ситуація 1: внутрішня помилка сервера – система сповіщає про це, і замовлення не оформлюється.

Виняткова ситуація 2: невалідні дані – система відображає повідомлення про помилку, користувач може повторно ввести дані.

Виняткова ситуація 3: сервіс оплати не відповідає – система відображає відповідне повідомлення, користувач може спробувати знов або змінити метод оплати.

Виняткова ситуація 4: недостатньо коштів на рахунку – система відображає відповідне повідомлення, користувач може спробувати знов або змінити реквізити картки.

Виняткова ситуація 5: товару більше немає в наявності – система відображає відповідне повідомлення, користувач може оформити замовлення знов.

Прецедент «Введення email».

Призначення: даний варіант використання надає можливість користувачу вказувати email при оформленні замовлення.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач вводить адресу електронної пошти у відповідне поле форми оформлення замовлення. Система запам'ятовує дані.

Виняткова ситуація 1: невалідні дані – система відображає повідомлення про помилку, користувач може повторно ввести дані.

Прецедент «Вибір платіжної системи».

Призначення: даний варіант використання надає можливість користувачу вибрати платіжну систему.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач обирає одну із платіжних систем з переліку. Система відображає форму введення реквізитів.

Прецедент «Вказання платіжних даних».

Призначення: даний варіант використання надає можливість користувачу ввести або підтягнути платіжні дані.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач вводить або обирає попередньо збережені платіжні дані в відповідні поля форми оформлення замовлення. Система попередньо перевіряє та запам'ятовує дані.

Вияткова ситуація 1: невалідні дані – система відображає повідомлення про помилку, користувач може повторно ввести дані.

Вияткова ситуація 2: неправильні реквізити картки – система відображає повідомлення про помилку, користувач може повторно ввести дані.

Прецедент «Вибір способу доставки».

Призначення: даний варіант використання надає можливість користувачу вибрати спосіб доставки доставки.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач обирає один із способів доставки з переліку. Система запам'ятовує вибір та додає вартість доставки до суми замовлення.

Прецедент «Вказання даних для доставки».

Призначення: даний варіант використання надає можливість користувачу ввести або підтягнути дані для доставки.

Основний потік подій: даний варіант використання починає виконуватися, коли користувач вводить або обирає попередньо збережені дані для доставки (ПІБ, адреса) у відповідні поля форми оформлення замовлення. Система запам'ятовує дані.

Вияткова ситуація 1: невалідні дані – система відображає повідомлення про помилку, користувач може повторно ввести дані.

2.3 Діаграма діяльності

Діаграма діяльності в Unified Modeling Language (UML) є важливим інструментом для моделювання процесів в рамках системи. Основна функція такої діаграми полягає в ілюстрації способів, якими різні активності взаємодіють і залежать одна від одної, а також у зображенні умов, що регулюють переходи між цими активностями [6].

Діаграма діяльності забезпечує детальне зображення бізнес-процесів або алгоритмів. Вона використовує різноманітні символи для представлення дій, рішень та паралелізму в процесах. На відміну від діаграми послідовностей, яка фокусується на взаємодії між об'єктами, діаграма діяльності демонструє потік управління від однієї активності до іншої, підкреслюючи, як різні операції впливають на рух даних і стан системи.

Використання діаграм діяльності може бути дуже ефективним при аналізі бізнес-процесів, де потрібно врахувати численні умовні поведінки та паралельні дії. Наприклад, у комплексних системах, таких як онлайн-банкінг або електронна комерція, діаграма діяльності допоможе визначити можливі затримки у обробці замовлень чи підтвердження платежів, а також оптимізувати потоки процесів для підвищення ефективності. Це включає моделювання сценаріїв, де користувачі мають виконувати рішення або вибір на основі поточного стану системи [10].

Також діаграма діяльності має важливе значення у фазі проектування і тестування. Через свою здатність чітко відображати процеси вона використовується для визначення тестових сценаріїв, які забезпечують покриття всіх потоків процесів. Такий підхід знижує ризики пропуску критичних шляхів процесів під час тестування, забезпечуючи більшу надійність і стабільність системи перед її запуском [6, 13].

Управління паралелізмом та залежностями між діями є ще однією ключовою особливістю діаграми діяльності. Це допомагає розробникам і аналітикам ідентифікувати можливі проблемні зони в архітектурі системи, де паралельне виконання процесів може призвести до конфліктів або взаємних блокувань ресурсів. Завдяки цьому можна заздалегідь спланувати необхідні механізми синхронізації та управління доступом до спільних ресурсів, що є критично важливим для систем з високою конкурентністю та великою кількістю користувачів.

На рисунках 2.4 – 2.6 наведено діаграму діяльності прецеденту «Оформити замовлення».

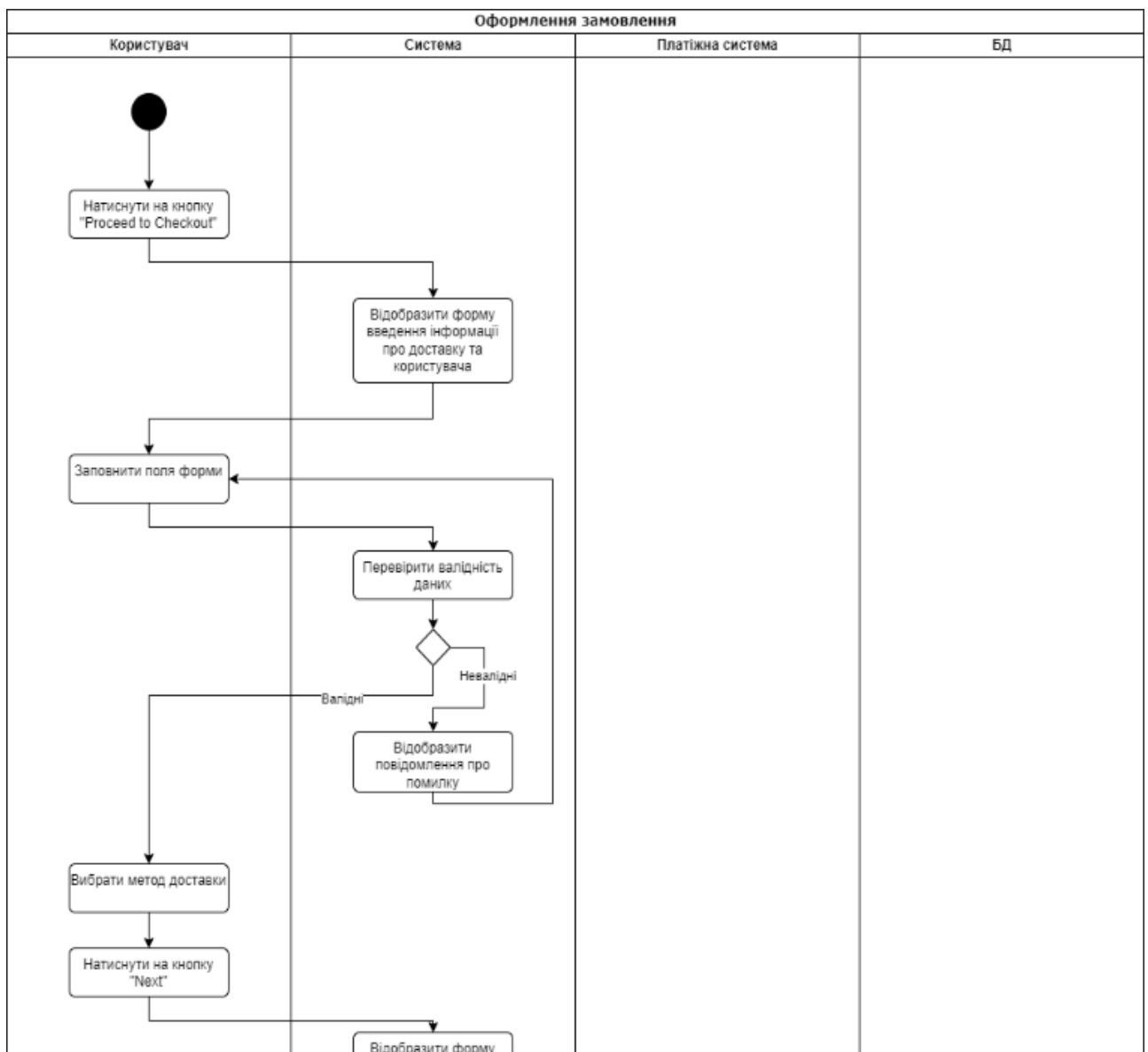


Рисунок 2.4

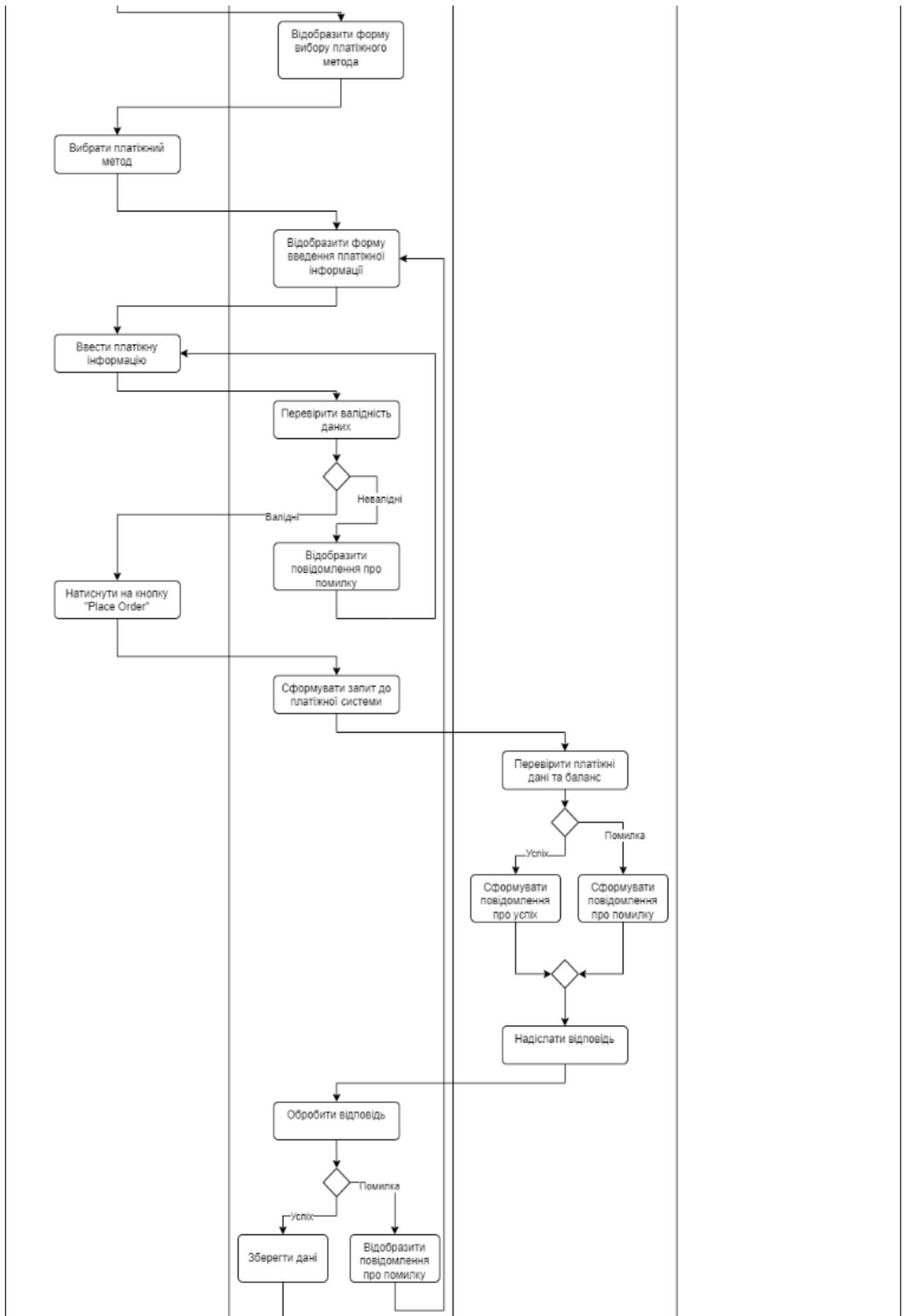


Рисунок 2.5

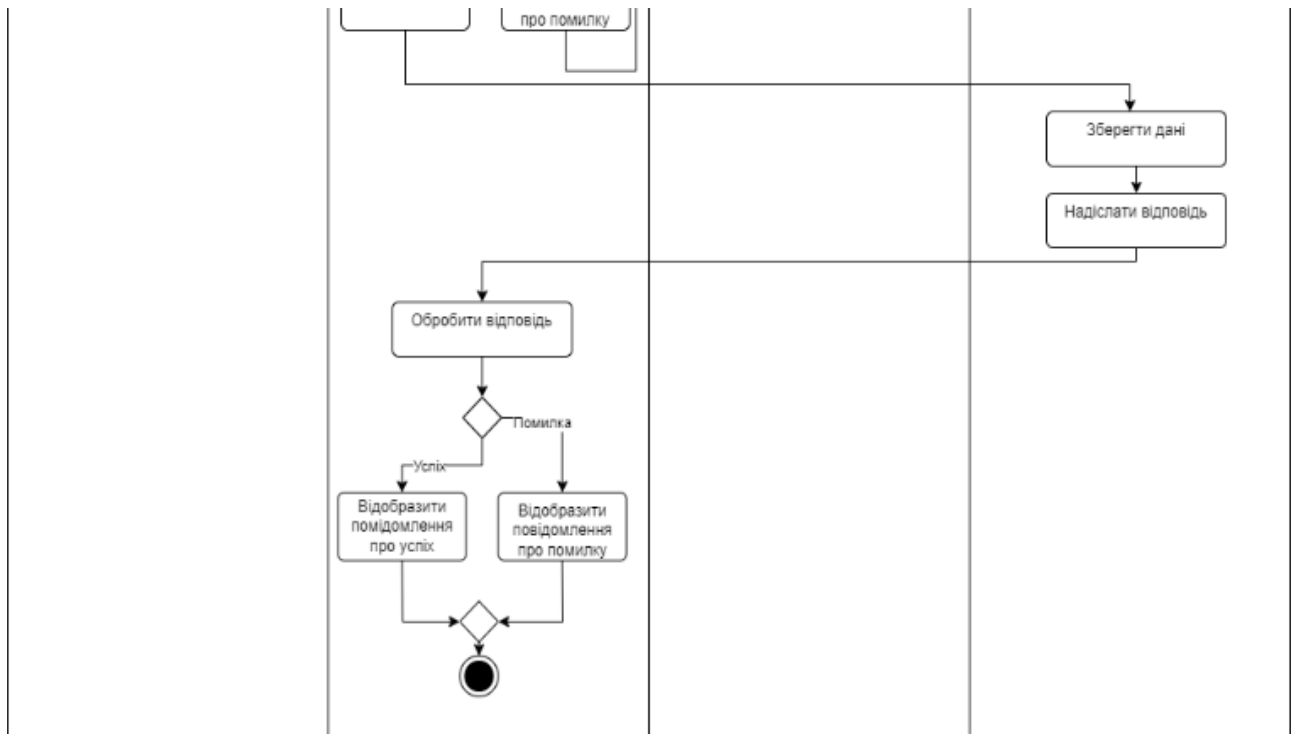


Рисунок 2.6

2.4 Діаграма послідовності

Діаграма послідовності (Sequence Diagram) – це один із типів діаграм взаємодії в уніфікованій мові моделювання UML (Unified Modeling Language). Вона використовується для візуального представлення взаємодії між об'єктами системи протягом певного періоду часу [6].

Діаграма послідовності показує, як об'єкти оперують один з одним шляхом обміну повідомленнями в хронологічному порядку. Вона відображає різні об'єкти, що беруть участь у взаємодії, як прямокутники розташовані горизонтально вздовж верхньої частини діаграми. Вертикальні лінії під кожним об'єктом називаються «лініями життя» (lifelines) і представляють існування об'єкта протягом взаємодії.

На діаграмі послідовності повідомлення, які обмінюються між об'єктами, зображуються стрілками, що з'єднують відповідні лінії життя. Ці стрілки демонструють порядок та напрямок, в якому повідомлення надсилаються від одного об'єкта до іншого [10].

Основні елементи та можливості діаграми послідовності:

- візуалізація порядку обміну повідомленнями між об'єктами;
- відображення об'єктів-учасників взаємодії та їхніх ролей;
- показ створення та знищення об'єктів під час взаємодії;
- представлення потоків керування (ітерації, альтернативні шляхи, паралелізм);
- демонстрація часових співвідношень між повідомленнями.

Діаграми послідовності є корисними для моделювання складних поведінкових сценаріїв, документування вимог, проєктування системи та тестування взаємодії між компонентами. Вони широко використовуються в процесі розробки програмного забезпечення з об'єктно-орієнтованою парадигмою [3, 10].

На рисунку 2.7 описана діаграма послідовності прецеденту «Оформити замовлення».

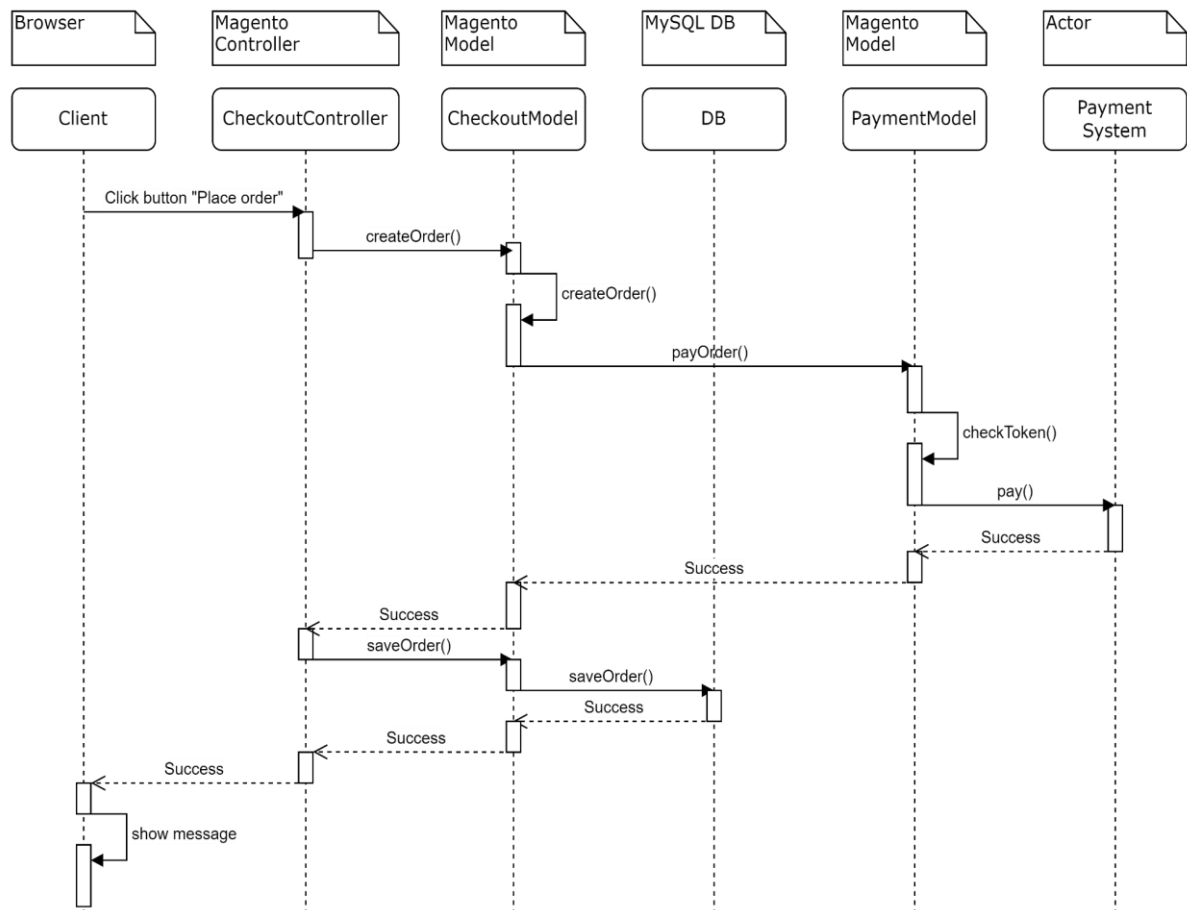


Рисунок 2.7 – Діаграма послідовності

2.5 Діаграма розгортання

Діаграма розгортання (Deployment Diagram) – це один з видів структурних діаграм в уніфікованій мові моделювання UML (Unified Modeling Language), який використовується для візуального представлення архітектури системи з точки зору розміщення програмних компонентів на різних вузлах обчислювальної інфраструктури [10].

Розглянемо основні елементи діаграми розгортання.

- вузол (Node) – це фізичний елемент, який представляє обчислювальний ресурс, такий як сервер, пристрій або вузол кластера (він може містити артефакти, що розгортаються на ньому);
- артефакт (Artifact) – це фізична реалізація програмного компонента або виконуваного модуля, такого як програмна бібліотека, виконуваний файл або вебсервіс;
- зв'язок (Link) – це асоціація між вузлами, що представляє канали комунікації, такі як мережеве з'єднання або інтерфейс.

Діаграма розгортання допомагає візуалізувати розміщення різних компонентів програмного забезпечення на фізичних вузлах і показує, як ці компоненти взаємодіють між собою через мережеві з'єднання або інші види зв'язків [10].

Основні цілі використання діаграми розгортання наступні:

- моделювання фізичної архітектури системи, включаючи обчислювальні ресурси та їх зв'язки;
- візуалізація розподілених компонентів системи та їх розміщення на вузлах;
- планування та документування процесу розгортання системи;
- виявлення потенційних вузьких місць та обмежень у архітектурі розгортання.

Діаграма розгортання є особливо корисною для великих та складних систем, де програмні компоненти можуть бути розподілені на різних фізичних

вузлах або середовищах. Вона допомагає розробникам та архітекторам зрозуміти загальну картину розміщення системи та спланувати ефективно розгортання.

На рисунку 2.8 наведено діаграму розгортання системи.

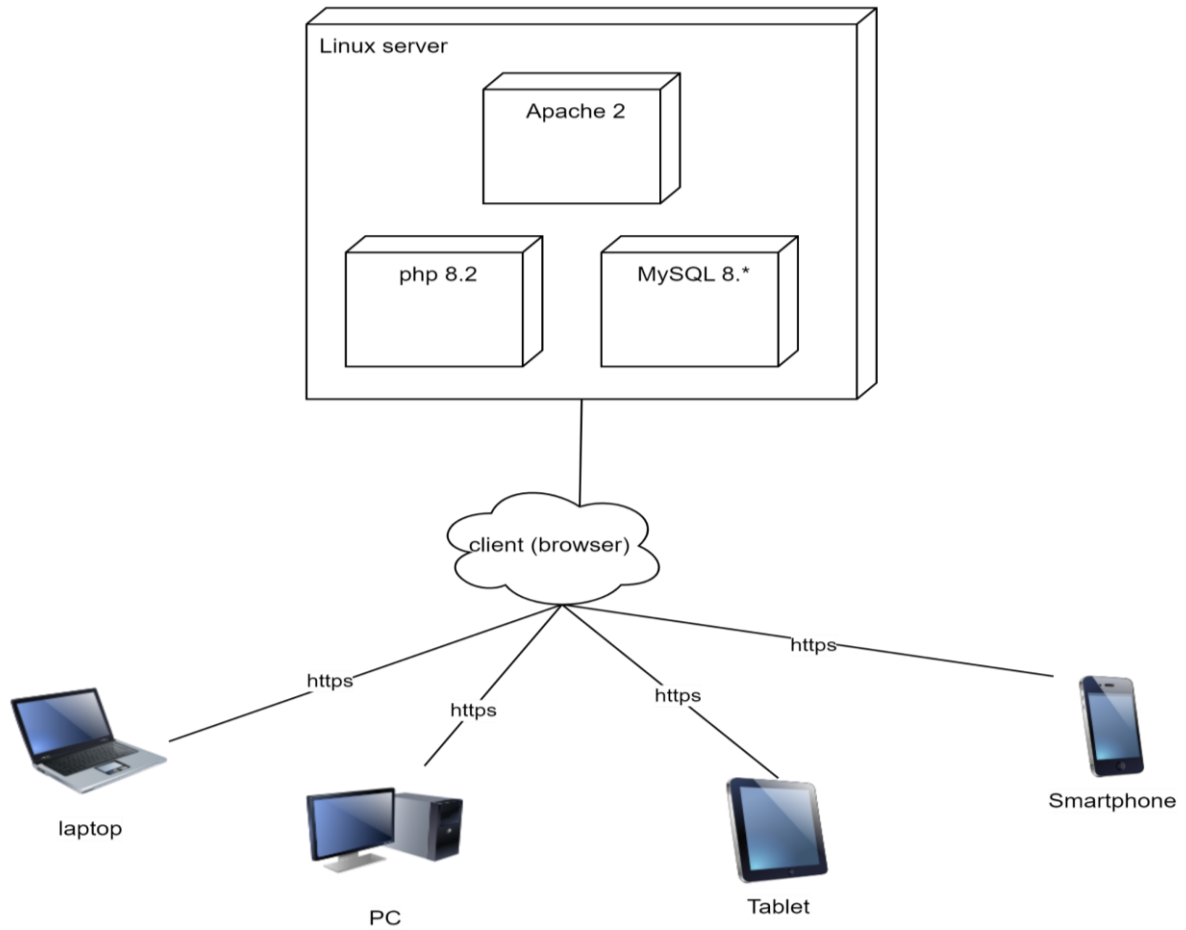


Рисунок 2.8 – Діаграма розгортання

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Опис інструментів розробки

Для реалізації було використано CMS Magento та пошуковий сервер Elasticsearch.

Elasticsearch – це пошуковий сервер із json rest api, що використовує Lucene і написаний на Java [7].

RequireJS – бібліотека, що допомагає покращити швидкість та якість коду, керуючи залежностями між JavaScript-файлами.

KnockoutJS – бібліотека, що допомагає у створенні більш оптимізованих сайтів.

3.2 Основні класи системи

Оскільки основним прецедентом системи є «Оформити замовлення», то основними класами системи будуть ті, що надають можливість взаємодіяти з кошиком, сервісами оплати та доставки, а також збереження даних замовлення.

Основними класами системи є:

- \Magento\Checkout\Model\Session відповідає за управління сесією оформлення замовлення;
- \Magento\Quote\Model\Quote представляє об'єкт кошика покупця;
- \Magento\Quote\Model\Quote\Address відповідає за представлення адрес доставки та білінгу;
- \Magento\Checkout\Model\Type\Onepage керує процесом оформлення замовлення на одній сторінці;
- \Magento\Quote\Model\Quote\Payment відповідає за зберігання інформації про платіжний метод та дані платежу;

- `\Magento\Sales\Model\Order` представляє об'єкт замовлення після його створення;
- `\Magento\Sales\Model\Order\Payment` відповідає за зберігання інформації про платіж для створеного замовлення;
- `\Magento\Sales\Model\Order\Address` відповідає за представлення адрес доставки та білінгу для створеного замовлення;
- `\Magento\Sales\Model\Order\Item` відповідає за представлення окремих товарів у замовленні;
- `\Magento\Sales\Model\Order\Invoice` відповідає за представлення рахунків, пов'язаних із замовленням.

Ці класи відповідають за різні аспекти процесу оформлення замовлення, починаючи від управління кошиком і закінчуючи створенням фактичного об'єкта замовлення та пов'язаних з ним елементів, таких як платежі, адреси та рахунки.

Детально ознайомитися з кодом системи можна в додатку В.

3.3 Підготовка та налаштування оточення

Для того щоб розгорнути систему на платформі Magento, необхідно налаштувати оточення. Процес налаштування включає в себе встановлення сервера, версії PHP, БД, тощо. Розглянемо цей процес детально.

Першим кроком у підготовці оточення є встановлення вебсервера. Magento 2 потребує вебсервер, такий як Apache або Nginx. Залежно від ОС це може бути також і багатоплатформова збірка вебсервера на кшталт OpenServer або XAMPP (рис. 3.1) [11].

Наступним важливим кроком є встановлення PHP версії 8.2 або новішої. Magento 2 вимагає наявності певних розширень PHP, таких як `php-mbstring`, `php-xml` та `php-intl` (рис. 3.2 – 3.3) [2, 3, 13].

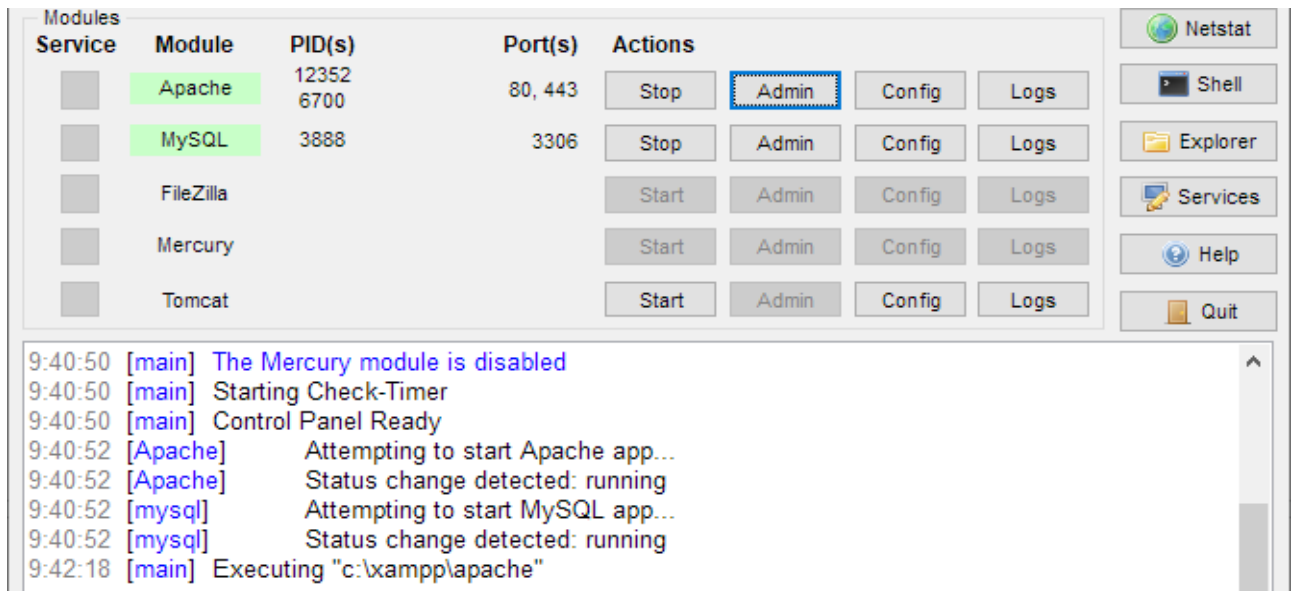


Рисунок 3.1 – Запуск Apache на XAMPP

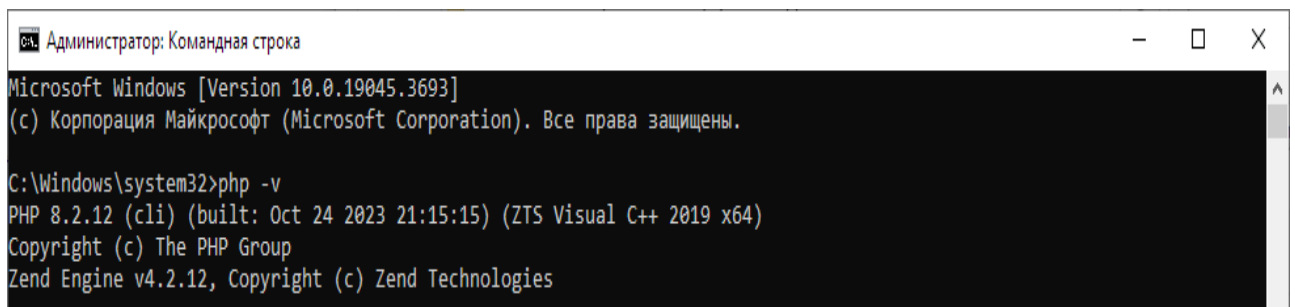


Рисунок 3.2 – Версія PHP

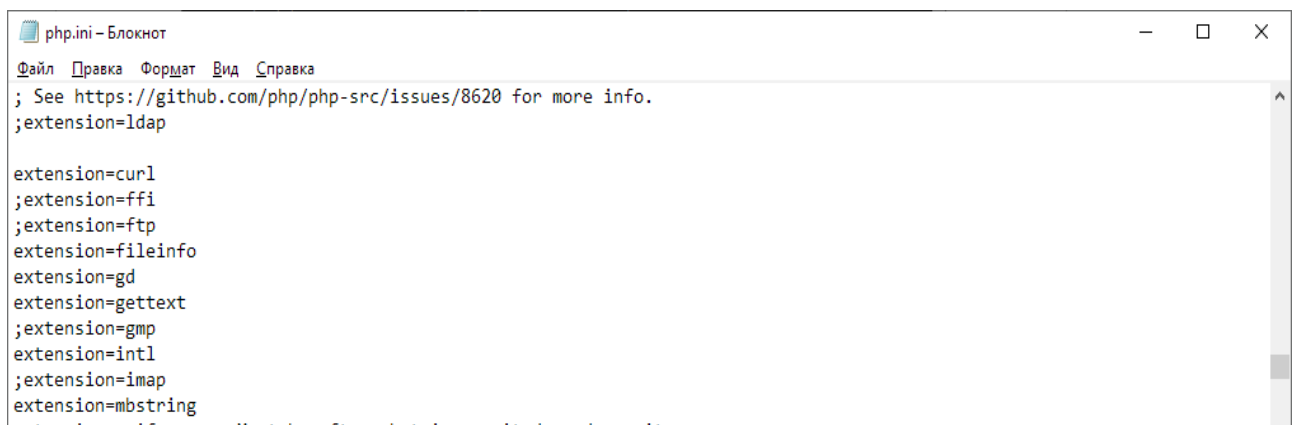


Рисунок 3.3 – Налаштування php.ini

Magento 2 потребує реляційну базу даних, таку як MySQL або MariaDB [9]. Необхідно встановити одну з цих СУБД і створити порожню базу даних для використання з Magento 2 (рис. 3.4 – 3.5).

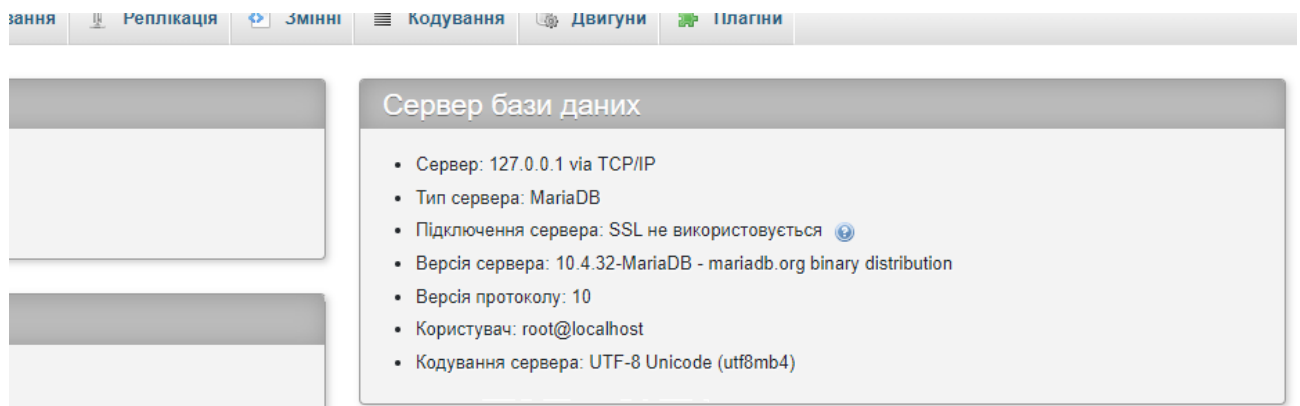


Рисунок 3.4 – Сервер БД

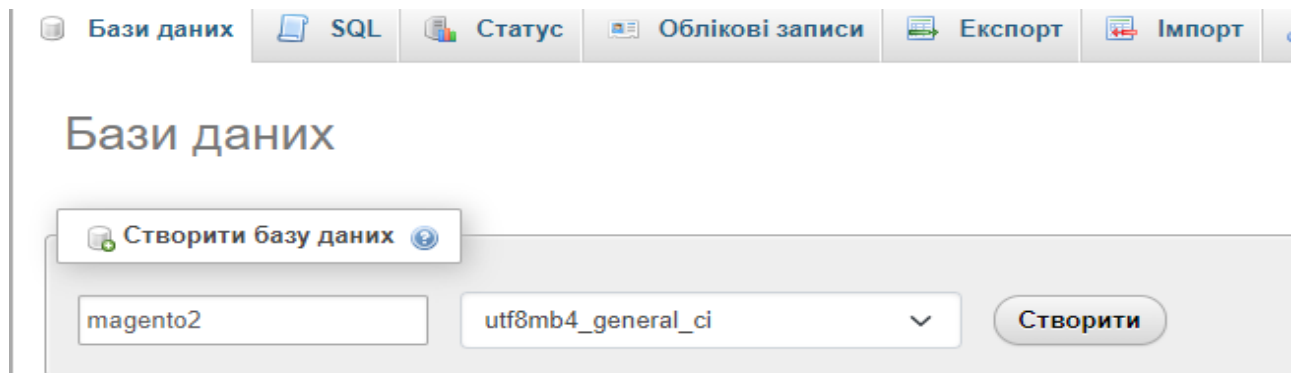


Рисунок 3.5 – Створення БД

Після встановлення вебсервера, РНР та бази даних, необхідно налаштувати конфігурацію вебсервера. Це включає налаштування шляху до кореневого каталогу Magento 2, обробку `.htaccess` файлів (для Apache), права доступу та інші параметри (рис. 3.6) [5].

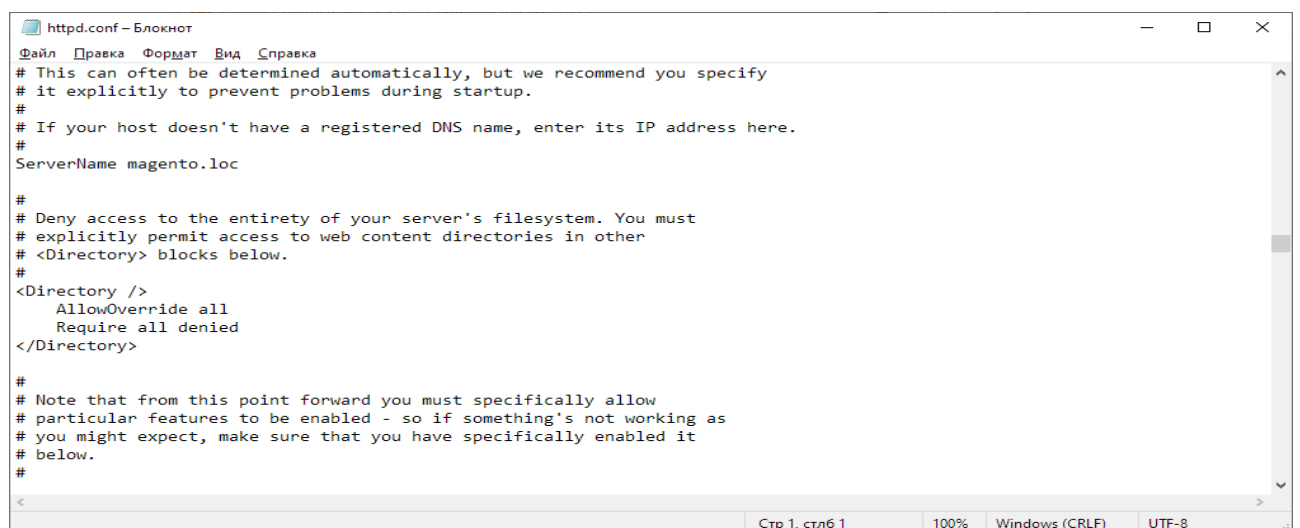


Рисунок 3.6 – Налаштування конфігурації вебсервера

Magento 2 використовує Composer для керування залежностями, тому наступним кроком є встановлення Composer [1] (рис. 3.7).

```
C:\xampp\htdocs\magento2>composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Warning: The lock file is not up to date with the latest changes in composer.json. You may be getting outdated dependencies. It is recommended that you run `composer update` or `composer update <package name>`.
Package operations: 199 installs, 0 updates, 0 removals
 - Downloading psr/log (1.1.4)
 - Downloading symfony/polyfill-ctype (v1.27.0)
 - Downloading symfony/polyfill-mbstring (v1.27.0)
 - Downloading symfony/polyfill-php80 (v1.27.0)
 - Downloading symfony/deprecation-contracts (v3.3.0)
 - Downloading symfony/finder (v5.4.21)
 - Downloading symfony/polyfill-intl-normalizer (v1.27.0)
 - Downloading symfony/polyfill-intl-grapheme (v1.27.0)
 - Downloading symfony/string (v5.4.22)
 - Downloading psr/container (1.1.2)
 - Downloading symfony/service-contracts (v2.5.2)
 - Downloading symfony/polyfill-php73 (v1.27.0)
 - Downloading symfony/console (v5.4.24)
 - Downloading composer/ca-bundle (1.3.6)
 - Downloading composer/pcr (3.1.0)
 - Downloading composer/class-map-generator (1.0.0)
```

Рисунок 3.7 – Встановлення Composer

Оскільки наш сервер використовує спеціальну доменну адресу для доступу до Magento 2, необхідно налаштувати файл «hosts» для відображення цієї адреси на відповідний IP-сервера (рис. 3.8).

```
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10      x.acme.com        # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1            localhost
#   ::1                  localhost

127.0.0.1 magento.loc
```

Рисунок 3.8 – Файл hosts

Одним із ключових компонентів, який потрібно встановити для Magento 2, є Elasticsearch – потужний пошуковий движок. Elasticsearch забезпечує швидкий та релевантний пошук для магазину Magento 2. Встановлення Elasticsearch включає завантаження відповідного пакету, налаштування конфігураційних файлів та інтеграцію з Magento 2 (рис. 3.9) [7].

```

"name" : "WIN4-218-1623",
"cluster_name" : "elasticsearch",
"cluster_uuid" : "qnbSiUMHSpmIXIpAwfW0fA",
"version" : {
  "number" : "7.12.0",
  "build_flavor" : "default",
  "build_type" : "zip",
  "build_hash" : "78722783c38caa25a70982b5b042074cde5d3b3a",
  "build_date" : "2021-03-18T06:17:15.410153305Z",
  "build_snapshot" : false,
  "lucene_version" : "8.8.0",
  "minimum_wire_compatibility_version" : "6.8.0",
  "minimum_index_compatibility_version" : "6.0.0-beta1"
}

```

Рисунок 3.9 – Параметри Elasticsearch

Після виконання всіх цих кроків, включаючи встановлення Elasticsearch, оточення готове для встановлення Magento 2 за допомогою командного рядка або вебінсталятора [7, 9, 11].

З файлом оточення можна ознайомитися в додатку А.

3.4 Тестування проєкту

Unit-тестування відіграє ключову роль у забезпеченні якості та стабільності програмного забезпечення, особливо у великих і складних системах, таких як Magento 2. Це процес тестування окремих модулів або компонентів, де кожен блок коду, наприклад функція або метод, перевіряється на різні вхідні дані та очікувані результати. Основна мета полягає в тому, щоб ізолювати кожен частину коду та перевірити її незалежно, гарантуючи, що вона працює належним чином.

Magento 2 має потужну систему unit-тестування, засновану на фреймворку PHPUnit [1, 3, 13]. Unit-тестування забезпечує раннє виявлення помилок, полегшує підтримку та рефакторинг коду, а також слугує документацією, демонструючи очікувану поведінку класів і методів. У складних системах, таких

як Magento 2, наявність якісних unit-тестів є критично важливою для забезпечення стабільності та якості програмного забезпечення.

Розглянемо приклад unit-тестування для класу `\Magento\Catalog\Model\Product\Option\Type\File` з ядра Magento 2. Цей клас відповідає за обробку опцій завантаження файлів для продуктів у магазині. Ось як виглядає unit-тест для методу `validateUserValue` цього класу (рис. 3.10). Детально ознайомитися з тестами можна в додатку Б.

```
class FileTest extends TestCase
{
    public function testValidateUserValue($value, $errors)
    {
        $optionMock = $this->getMockBuilder(\Magento\Catalog\Model\Product\Option::class)
            ->disableOriginalConstructor()
            ->getMock();

        $requestMock = $this->createMock(Http::class);
        $requestMock->expects($this->any())
            ->method('getServer')
            ->willReturn([]);

        $model = new File($requestMock);
        $model->setOption($optionMock);

        $this->assertEquals($errors, $model->validateUserValue($value));
    }
}
```

Рисунок 3.10 – Unit-тестування

3.5 Керівництво користувача

3.5.1 Рівень підготовки користувача

Ефективна робота з Magento 2 вимагає від користувачів базових комп'ютерних навичок та уважного вивчення інструкцій. Ретельне ознайомлення з керівництвом користувача допоможе опанувати функціонал платформи та забезпечує зручність взаємодії з нею.

Рівень необхідної підготовки залежить від ролі користувача – адміністратора, розробника, дизайнера чи власника магазину. Проте базове розуміння концепцій електронної комерції, вебтехнологій та особливостей Magento 2 є ключовим для всіх категорій користувачів.

3.5.2 Реєстрація/вхід

На головній сторінці магазину натисніть на посилання «Create an Account» в верхньому меню. Це перенаправить Вас на сторінку реєстрації (рис. 3.11).

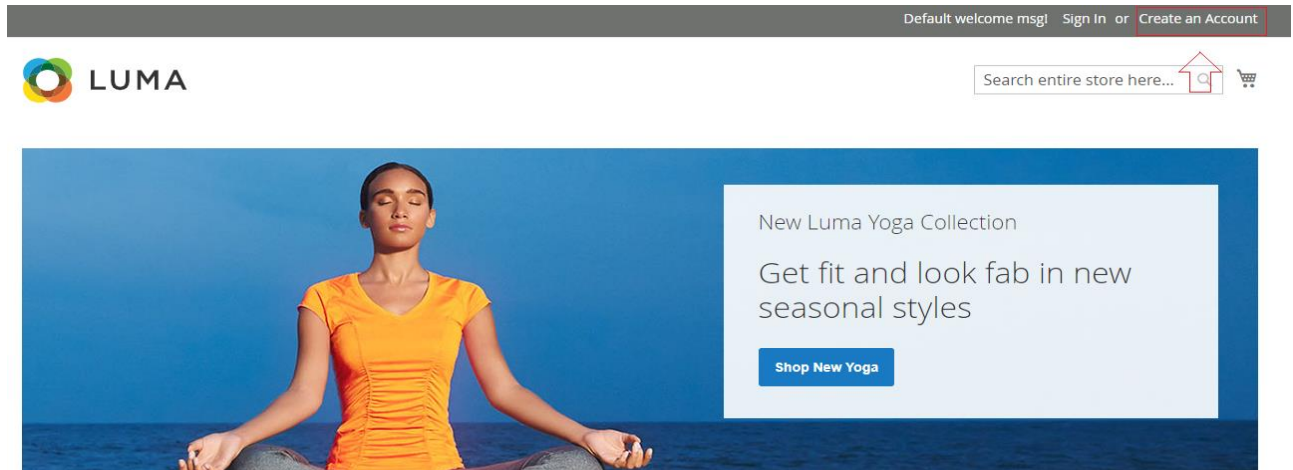


Рисунок 3.11 – Верхнє меню

На сторінці реєстрації Ви побачите форму, де потрібно ввести особисті дані, таку як ім'я, прізвище, електронна адреса та бажаний пароль. Обов'язково ознайомтеся з політикою конфіденційності та умовами обслуговування і поставте відмітку, якщо ви їх приймаєте (рис. 3.12).

Personal Information

First Name *

Last Name *

Sign Up for Newsletter

Sign-in Information

Email *

Password *

Password Strength: Very Strong

Confirm Password *

* Required Fields

[Create an Account](#)

Рисунок 3.12 – Форма реєстрації

Після заповнення всіх необхідних полів натисніть кнопку «Create an Account». Ваша реєстрація буде успішною, і ви отримаєте повідомлення про підтвердження на вказану електронну адресу (рис. 3.13).

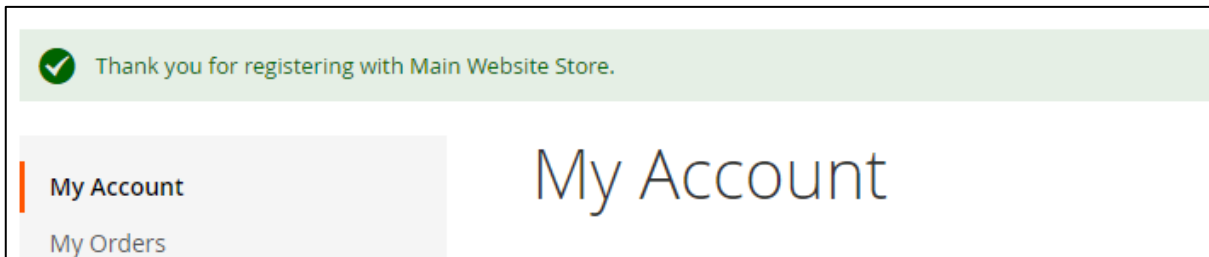


Рисунок 3.13 – Успішна реєстрація

Якщо ви вже зареєстровані, то перейдіть на сторінку входу, натиснувши на посилання «Sign In» в верхньому меню (рис. 3.14).

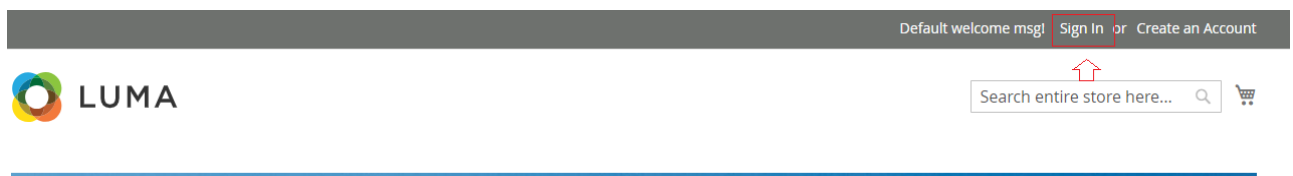


Рисунок 3.14 – Пункт «Sign In»

На сторінці входу введіть свою електронну адресу та пароль у відповідні поля та натисніть кнопку «Sign In» (рис. 3.15).

Registered Customers

If you have an account, sign in with your email address.

Email *

Password *

[Sign In](#) [Forgot Your Password?](#)

Рисунок 3.15 – Форма входу до системи

Якщо ваші облікові дані правильні, ви будете успішно авторизовані і перенаправлені на сторінку облікового запису, де можна керувати замовленнями, адресами доставки та іншими налаштуваннями профілю (рис. 3.16).

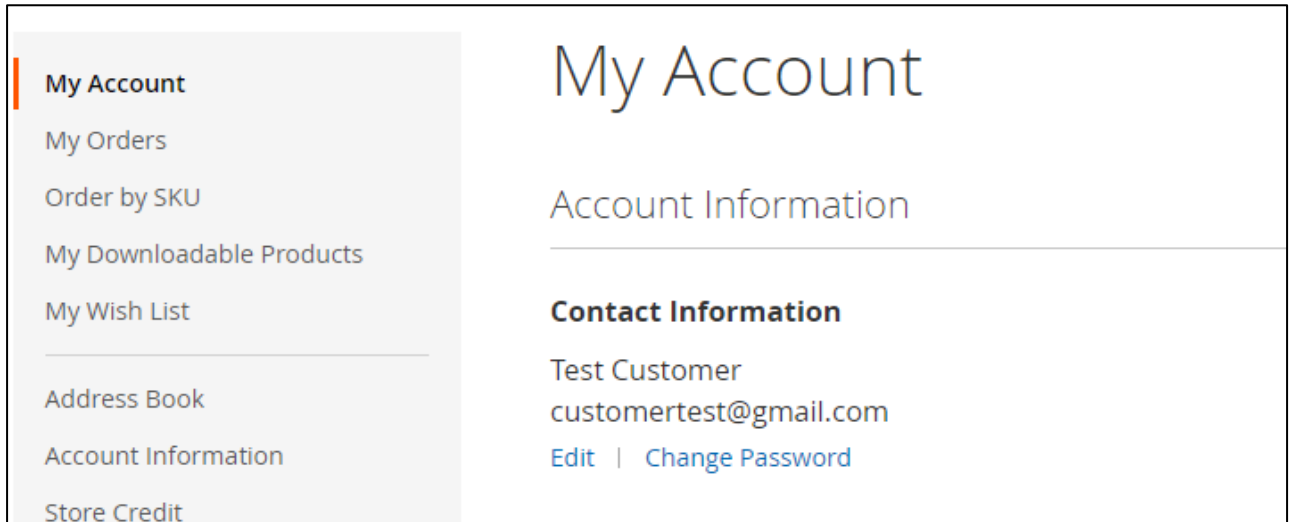


Рисунок 3.16 – Обліковий запис користувача

Якщо ви забули пароль, натисніть на посилання «Forgot your password?» під формою входу. Ви будете перенаправлені на сторінку відновлення пароля, де потрібно ввести свою електронну адресу. На вказану адресу буде надіслано інструкції для скидання пароля.

Після успішної реєстрації або входу ви матимете доступ до різних функцій і привілеїв, таких як перегляд історії замовлень, зберігання адрес доставки, відстеження замовлень та отримання персоналізованих знижок і пропозицій від магазину.

3.5.3 Процес оформлення замовлення

Процес оформлення замовлення в Magento 2 починається з вибору потрібного продукту. Перейдіть на сайт магазину та знайдіть бажаний продукт за допомогою пошуку, переглядаючи категорії або список продуктів. Натисніть

на назву або зображення продукту, щоб перейти на його сторінку з детальним описом (рис. 3.17 – 3.18).

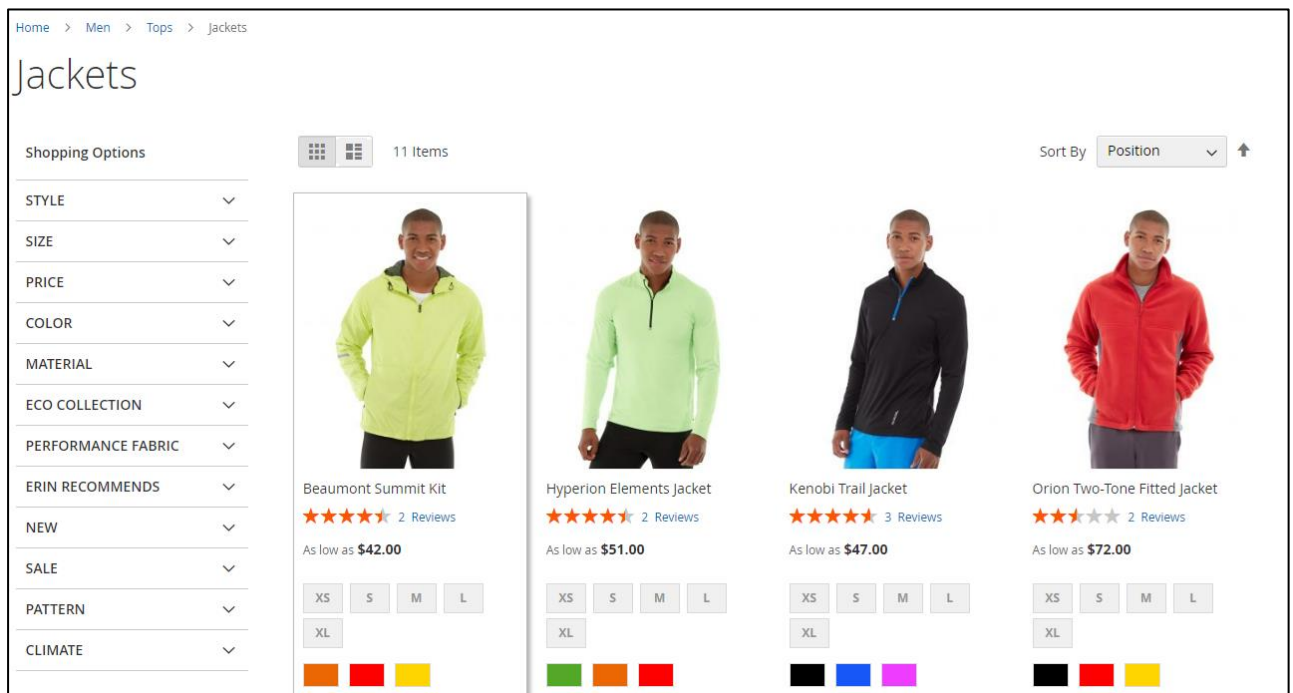


Рисунок 3.17 – Каталог продуктів

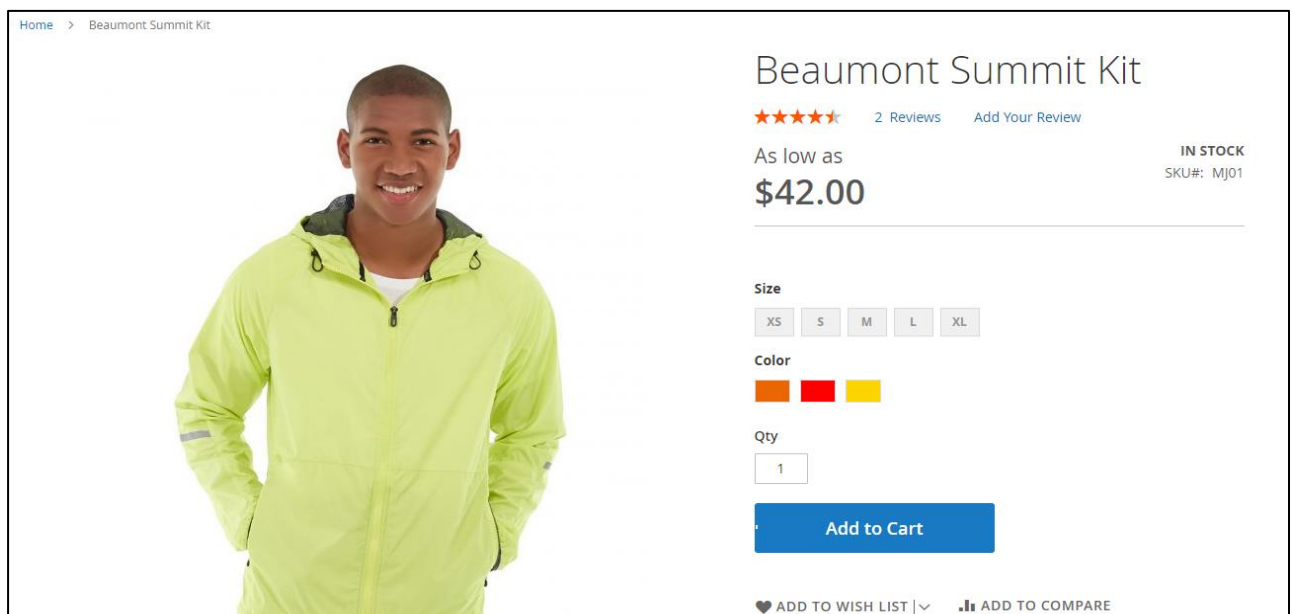


Рисунок 3.18 – Картка продукту

На сторінці продукту ознайомтеся з його характеристиками, ціною, наявністю та іншою інформацією. За потреби, виберіть необхідні опції продукту, такі як колір, розмір або додаткові аксесуари. Встановіть бажану кількість

одиниць продукту та натисніть кнопку «Add to Card» для додавання продукту до вашого кошика покупок (рис. 3.19).

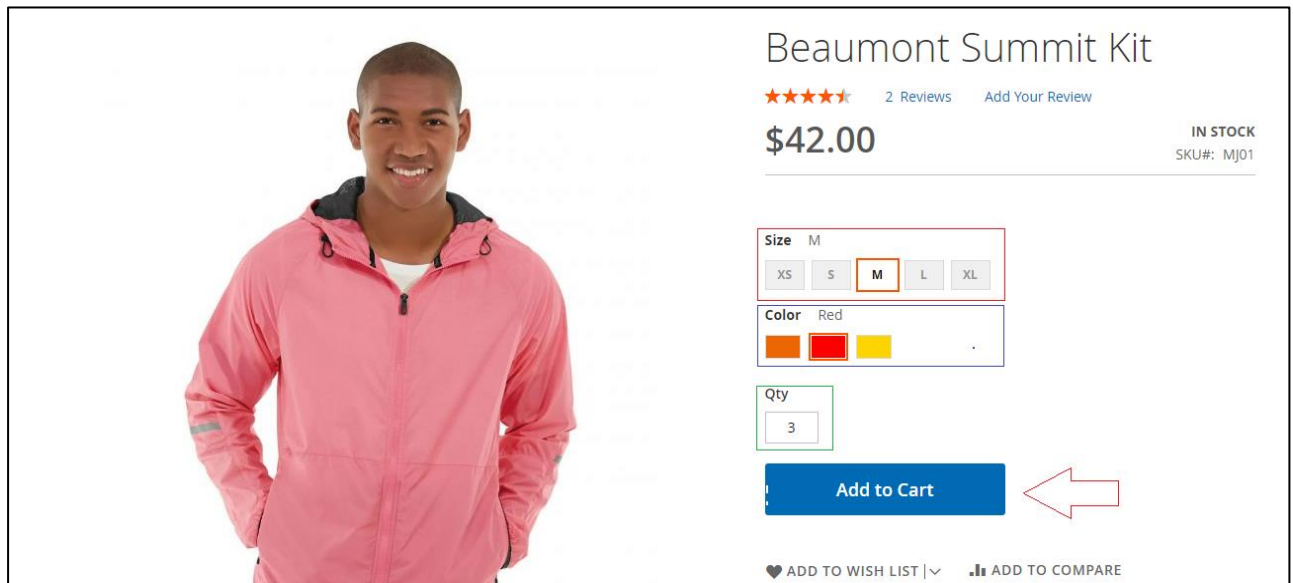


Рисунок 3.19 – Вибір опцій продукта

Після додавання продукту до кошика з'явиться спливаюче вікно з підсумковою інформацією про поточний вміст кошика. Натисніть «View and Edit Card» або «Proceed to Checkout» для переходу до оформлення замовлення (рис. 3.20).

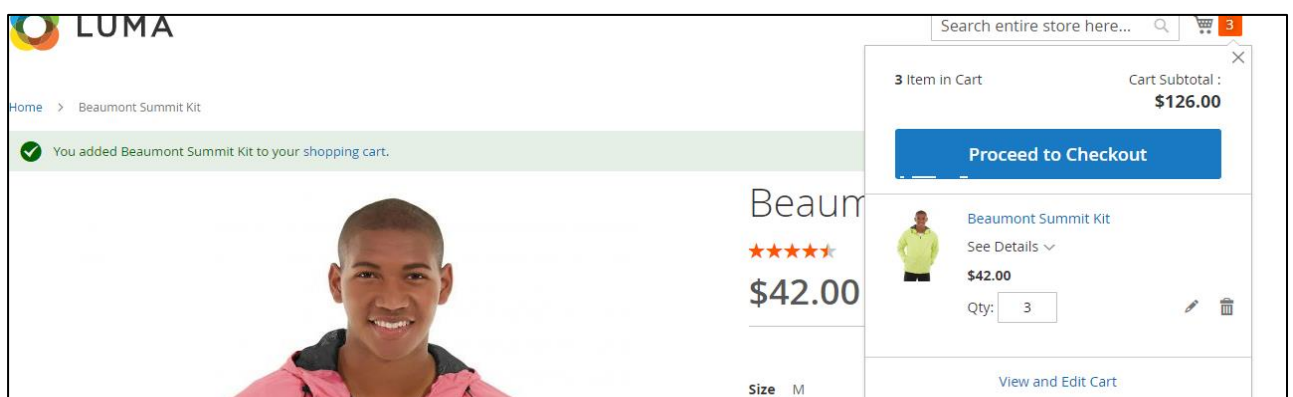


Рисунок 3.20 – Вибір опцій продукта

На сторінці кошика ви побачите список всіх доданих продуктів із зазначенням їх цін, кількості та загальної вартості. На цьому етапі ви можете змінити кількість або видалити непотрібні продукти з кошика (рис. 3.21).

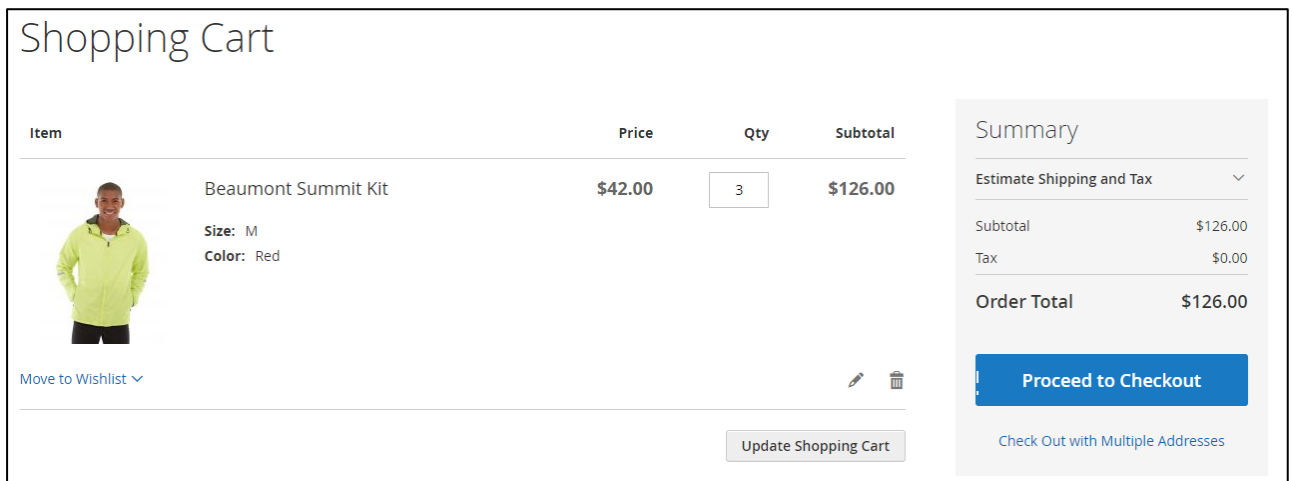


Рисунок 3.21 – Кошик

Після перевірки вмісту кошика натисніть кнопку «Proceed to Checkout» для початку процесу оформлення. Якщо ви вже маєте обліковий запис у магазині, введіть свої облікові дані для входу. В іншому випадку, вам буде запропоновано створити новий обліковий запис або продовжити як гість (рис. 3.22).

Рисунок 3.22 – Введення даних користувача

Далі введіть або підтвердіть свою адресу доставки та адресу виставлення рахунку (якщо вони відрізняються). Виберіть бажаний спосіб доставки та введіть будь-яку додаткову інформацію для доставки, якщо це потрібно (рис. 3.23 – 3.24).

Наступним кроком є вибір способу оплати (банківська картка, електронний гаманець, оплата при отриманні тощо) та введення відповідних платіжних даних (рис. 3.25).

Street Address *

Sobornaya st. 164

Country *

United States

State/Province *

Alabama

City *

Montgomery

Рисунок 3.23 – Введення адреси доставки

Shipping Methods

\$0.00 Table Rate Best Way

\$15.00 Fixed Flat Rate

Next

Рисунок 3.24 – Вибір способу доставки

Check / Money order

Cash On Delivery

Bank Transfer Payment

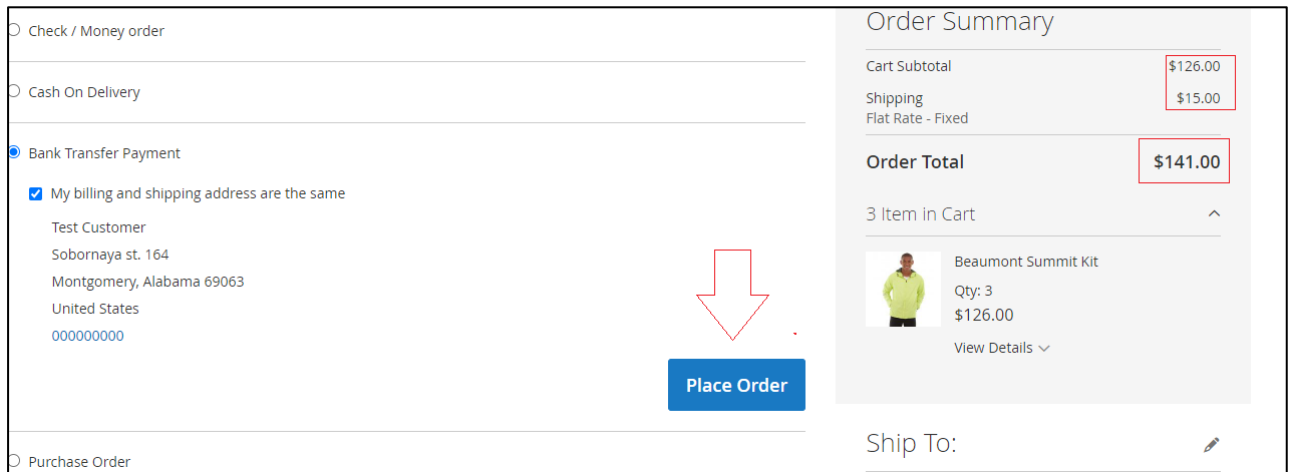
My billing and shipping address are the same

Test Customer
Sobornaya st. 164
Montgomery, Alabama 69063
United States
000000000

Place Order

Рисунок 3.25 – Вибір методу оплати

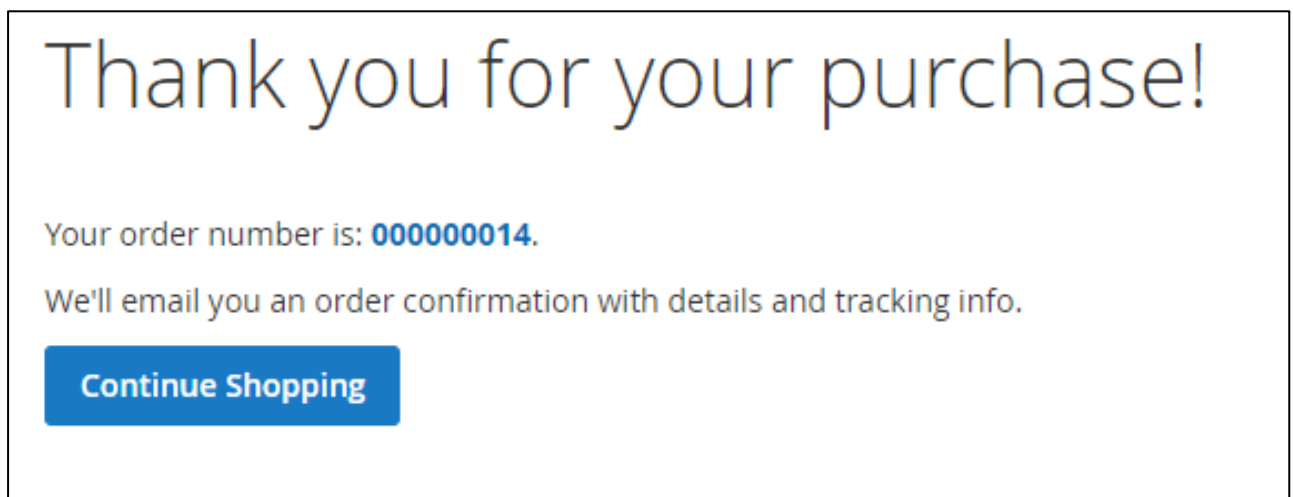
Перевірте підсумкову інформацію замовлення, включно з загальною вартістю, витратами на доставку та податками. Прийміть умови угоди та політику конфіденційності магазину та натисніть кнопку «Place Order» для завершення процесу оформлення (рис. 3.26).



The screenshot displays the checkout process. On the left, there are three payment options: 'Check / Money order', 'Cash On Delivery', and 'Bank Transfer Payment' (which is selected). Below the selected option, there is a checkbox for 'My billing and shipping address are the same' and a shipping address form with the following details: Test Customer, Sobornaya st. 164, Montgomery, Alabama 69063, United States, and zip code 00000000. A red arrow points to the 'Place Order' button. On the right, the 'Order Summary' section shows: Cart Subtotal (\$126.00), Shipping Flat Rate - Fixed (\$15.00), and Order Total (\$141.00). Below this, it indicates '3 Item in Cart' and lists one item: Beaumont Summit Kit, Qty: 3, \$126.00. At the bottom right, there is a 'Ship To:' field with an edit icon.

Рисунок 3.26 – Підтвердження замовлення

Після успішного оформлення замовлення ви отримаєте підтвердження з номером вашого замовлення та докладною інформацією про ваше замовлення (рис. 3.27).



The screenshot shows a confirmation message: 'Thank you for your purchase!'. Below this, it states 'Your order number is: 00000014.' and 'We'll email you an order confirmation with details and tracking info.' At the bottom, there is a blue button labeled 'Continue Shopping'.

Рисунок 3.27 – Оформлене замовлення

Також ви можете відслідковувати статус виконання замовлення в особистому кабінеті (рис. 3.28).

Order #	Date	Order Total	Status	Action
000000014	5/9/24	\$141.00	Processing	View Order Reorder

1 Item Show per page

Рисунок 3.28 – Перегляд статусу замовлення

Відтепер замовлення буде опрацьовуватись магазином, і ви отримаєте сповіщення на кожному етапі його обробки та доставки.

ВИСНОВКИ

У ході виконання даної роботи було розроблено торговий майданчик, який реалізовано за допомогою CMS Magento 2. Реалізована система електронної комерції забезпечує зручний інтерфейс для перегляду каталогу товарів, формування замовлень, вибору способів доставки та оплати. Використання Magento та Elasticsearch дозволяє створити масштабовану та продуктивну платформу для ведення електронної комерції. Ретельне тестування системи забезпечує її стабільність та відповідність вимогам. Застосування UML у процесі розробки сприяє кращому розумінню вимог, архітектури та процесів системи, полегшуючи її подальшу підтримку та розширення.

У відповідності з метою кваліфікаційної роботи було розроблено торговий майданчик з використаннями наступних технологій:

- Magento 2 для реалізації клієнтської та серверної частини;
- Elastic Search для пошуку.

У відповідності з поставленими задачами були виконані наступні етапи створення системи:

- сформовані вимоги до системи (функціональні та нефункціональні), наведено огляд аналогів та їх порівняння;
- спроектована та побудована структура системи (побудовані UML-діаграми: прецедентів, діяльності, послідовності та розгортання; надано опис кожного з прецедентів системи);
- реалізовано торговий майданчик (наведена інструкція по розгортанню середовища, надано покрокове керівництво користувача та вказані основні класи системи);
- протестована робота системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Adobe Commerce Developer Documentation. URL: <https://developer.adobe.com/commerce/docs/> (дата звернення: 04.04.2024).
2. Adobe Commerce Merchant Documentation. URL: <https://experienceleague.adobe.com/en/docs/commerce-admin/user-guides/home> (дата звернення: 02.04.2024).
3. Bierer D. PHP 8 Programming Tips, Tricks and Best Practices: A practical guide to PHP 8 features, usage changes, and advanced programming techniques. Birmingham : Packt Publishing, 2021. 528 p.
4. BigCommerce. URL: <https://www.bigcommerce.com/> (дата звернення: 02.03.2024).
5. Burge S. Magento 2 Explained: Your Step-by-Step Guide to Magento 2. Michigan : Independently published, 2018. 335 p.
6. Dennis A., Wixom B., Tegarden D. Systems Analysis and Design: An Object-Oriented Approach with UML. Hoboken : Wiley, 2020. 544 p.
7. Konda M. Elasticsearch in Action, Second Edition. Shelter Island : Manning, 2023. 592 p.
8. Shopify. URL: <https://www.shopify.com/> (дата звернення: 02.03.2024).
9. Stech C. Magento 2 Handbuch: Magento Open Source 2.3.2. Norderstedt : Books on Demand, 2019. 492 p.
10. Unhelkar B. Software Engineering with UML. Boca Raton : Auerbach Publications, 2020. 426 p.
11. Vishwakarma V., Vishwakarma R., Vishwakarma A. D. Adobe Commerce: eCommerce Marketing Platform Fundamental of Magento 2.0 Setup and Development. Birmingham : Packt Publishing, 2022. 237 p.
12. WooCommerce. URL: <https://woocommerce.com/> (дата звернення: 02.03.2024).

13. Zandstra M. PHP 8 Objects, Patterns, and Practice: Mastering OO Enhancements, Design Patterns, and Essential Development Tools. London : Apress, 2021. 858 p.

ДОДАТОК А

Файл оточення

```
<?php
return [
    'backend' => [
        'frontName' => 'admin'
    ],
    'cache' => [
        'graphql' => [
            'id_salt' => '6LK3jXmB793yHxd0vuVNjda6exKgC5id'
        ],
        'frontend' => [
            'default' => [
                'id_prefix' => 'ca3_'
            ],
            'page_cache' => [
                'id_prefix' => 'ca3_'
            ]
        ],
        'allow_parallel_generation' => false
    ],
    'queue' => [
        'consumers_wait_for_messages' => 1
    ],
    'remote_storage' => [
        'driver' => 'file'
    ],
    'crypt' => [
        'key' => '8e3a2b2283bba9b05de40b681f2ebe11'
    ],
    'db' => [
        'table_prefix' => '',
        'connection' => [
            'default' => [
                'host' => 'localhost',
```



```

'dbname' => 'magento2',
'username' => 'root',
'password' => '',
'model' => 'mysql4',
'engine' => 'innodb',
'initStatements' => 'SET NAMES utf8;',
'active' => '1',
'driver_options' => [
    1014 => false
]
]
],
'resource' => [
    'default_setup' => [
        'connection' => 'default'
    ]
],
'x-frame-options' => 'SAMEORIGIN',
'MAGE_MODE' => 'default',
'session' => [
    'save' => 'files'
],
'lock' => [
    'provider' => 'db'
],
'directories' => [
    'document_root_is_pub' => true
],
'elasticsearch7' => [
    'hostname' => 'localhost',
    'port' => '9200',
    'username' => '',
    'password' => '',
    'enableDefaultIndices' => true,
],
];

```

ДОДАТОК Б

Тестування

```
<?php
namespace Magento\Catalog\Test\Unit\Model\Product\Option\Type;

use Magento\Catalog\Model\Product\Option\Type\File;
use Magento\Framework\App\Request\Http;
use PHPUnit\Framework\TestCase;

class FileTest extends TestCase
{
    /**
     * @dataProvider validateUserValueDataProvider
     */
    public function testValidateUserValue($value, $errors)
    {
        $optionMock = $this->getMockBuilder(\Magento\Catalog\Model\Product\Option::class)
            ->disableOriginalConstructor()
            ->getMock();

        $requestMock = $this->createMock(Http::class);
        $requestMock->expects($this->any())
            ->method('getServer')
            ->willReturn([]);

        $model = new File($requestMock);
        $model->setOption($optionMock);

        $this->assertEquals($errors, $model->validateUserValue($value));
    }

    public function validateUserValueDataProvider()
    {
        return [
            'valid_value' => [
```

```
    ['file' => ['error' => 0]],  
    true,  
  ],  
  'invalid_value' => [  
    ['file' => ['error' => 2]],  
    false,  
  ],  
];  
}  
}
```

ДОДАТОК В

Посилання на Git

URL: https://bitbucket.org/s_var_og/ecommerce/src/main/.