

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

«Вебсервіс для продажу ветеринарних препаратів»

Назва теми

Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.2201113.01.18.ПЗ

Виконав студент IV курсу групи ПЗ-22-1



Підпис

Назар ФІЛОНЮК

Ім'я, прізвище

Керівник канд. техн. наук, доцент
Науковий ступінь, звання



Підпис

Оксана ЯШИНА

Ім'я, прізвище

Нормоконтролер канд. техн. наук, доцент




Підпис

Юрій ФОРКУН

Ім'я, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення



Підпис

Леонід БЕДРАТЮК

Ім'я, прізвище

4 червня 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри іпз

Л. П. Бедратюк 

02 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Філонюку Назару Юрійовичу

Прізвище, ім'я, по батькові студента

1. Тема роботи Вебсервіс для продажу ветеринарних препаратів

Керівник роботи Яшина Оксана Миколаївна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. №7.

2. Строк подання студентом роботи на кафедру 01.06.26 р.

3. Вихідні дані до роботи Методичні матеріали до кваліфікаційної роботи бакалавра

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

Аналіз предметної області, визначення її функціональних та нефункціональних вимог предметної області, постановка задачі, проектування програмного забезпечення, програмна реалізація, проведення тестування інтернет-магазину.

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

Три креслення:

1. Діаграма варіантів використання для зареєстрованого користувача, діаграма станів

2. Діаграма _____ варіантів _____ використання _____ для _____

3. Діаграма станів _____

3. Діаграма станів.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Форкун Ю. В., доцент кафедри ІПЗ	05.05.26 	25.05.26 
Антиплагіат	Форкун Ю. В., доцент кафедри ІПЗ	05.05.26 	25.05.26 

7. Дата видачі завдання « 02 » січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою кваліфікаційної роботи (КвР), визначення та узгодження індивідуальних тем КвР	01.12 – 31.12.2025	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2026	
3 Проектування програмного забезпечення	01.02 – 28.02.2026	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2026	
5 Тестування програмного забезпечення	11.04 – 30.04.2026	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2026	
7 Попередній захист КвР	травень 2026 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2026	
9 Здача КвР на кафедру; підготовка КвР для розміщення у репозиторії ХНУ; підготовка до захисту та захист КвР	з 01.06.2026	

Студент


Підпис

Назар ФІЛОНЮК

Ініціали, прізвище

Керівник роботи


Підпис

Оксана ЯШИНА

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Вебсервіс для продажу ветеринарних препаратів».

Автор роботи: Філонюк Назар Юрійович.

Керівник роботи: Яшина Оксана Миколаївна.

Пояснювальна записка: 80 с., 26 рис., 11 табл., 2 дод., 40 джерел.

Графічна частина: 3 креслення у форматі А3.

ВЕБРЕСУРС, ВЕБЗАСТОСУНОК, ВЕБРЕСУРС, ІНТЕРНЕТ-МАГАЗИН, ВЕБ,
ЕЛЕКТРОННА КОМЕРЦІЯ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ,
ПРОДАЖ ВЕТЕРИНАРНИХ ПРЕПАРАТІВ.

Мета кваліфікаційної роботи: здійснити проектування та розробку вебсервісу для продажу ветеринарних препаратів.

У ході реалізації проекту було здійснено аналіз предметної області, визначено завдання для проведення кваліфікаційної роботи, в кінці зроблено висновки.

Вебзастосунок розроблявся за допомогою мови програмування JavaScript.

Результатом кваліфікаційної роботи став вебресурс для продажу ветеринарних товарів, що дозволяє користувачам легко використовувати даний ресурс для здійснення замовлення та покупок, маючи доступ до мережі Інтернет.





27.05.2026

Дата


Підпис

ЗМІСТ

Вступ	7
1 Дослідження предметної області та постановка задачі	10
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	10
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	14
1.3 Визначення функціональних та нефункціональних вимог	21
2 Проектування вебсервісу	29
2.1 Проектування архітектури та структури вебсервісу	29
2.2 Проектування логічної моделі бази даних	32
2.3 Проектування інтерфейсу користувача	38
2.4 Аналіз та вибір технологій і методів реалізації вебсервісу	43
3 Програмна реалізація та тестування	46
3.1 Реалізація бази даних вебсервісу	46
3.2 Реалізація модулів вебсервісу	51
3.3 Реалізація інтерфейсу користувача. Інструкція користувача	55
3.4 Тестування вебсервісу	58
Висновки	62
Перелік джерел посилання	64
Додаток А Презентаційні матеріали	67
Додаток Б Технічне завдання	75

					КвРІПЗ.2201113.01.18.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	Вебсервіс для продажу ветеринарних препаратів Відомість документів	Лім.	Арк.	Аркушів	
Виконала		Філонюк Н.Ю.		25.05				6	76
Керівник		Яшина О.М.		25.05				ХНУ, ІПЗ-22-1	
Н. контр.		Форкун Ю.В.		25.05					
Зав. каф.		Бедратюк Л.П.		25.05					

ВСТУП

Актуальність розробки вебсервісу для торгівлі ветеринарними препаратами, а разом з тим спеціалізованого вебмодуля для ветеринарного інтернет-магазину сьогодні виходить далеко за межі звичайної автоматизації продажів. Ми живемо в епоху «гуманізації» домашніх тварин, де власник сприймає kota чи собаку як повноцінного члена родини, а отже, очікує від сервісу такої ж точності та швидкості, як у найкращих медичних клініках чи сучасних маркетплейсах.

Замість того, щоб бути просто цифровою вітриною, вебмодуль стає інтелектуальним ядром, яке з'єднує склад, клієнта та специфічні медичні вимоги. У традиційних магазинах замовлення часто обробляються хаотично через месенджери або таблиці, що неминуче призводить до втрати клієнтів у моменти пікових навантажень. Спеціалізоване рішення дозволяє перетворити цей «клубок» процесів на чіткий конвеєр, де кожен етап — від резервування лікувального корму до контролю температурного режиму при доставці вакцин - стає прозорим і контрольованим.

Ветеринарна сфера має унікальний набір критичних параметрів, які звичайні CRM-системи часто ігнорують. Наприклад, багато препаратів мають суворі терміни придатності та потребують дотримання принципу FIFO (першим виходить той товар, у якого раніше закінчується термін). Розробка модуля дозволяє впровадити автоматичну перевірку цих термінів ще на етапі формування кошика, запобігаючи продажу простроченої продукції, що для ветаптеки є не лише репутаційним, а й юридичним ризиком. Крім того, інтеграція функціоналу для перевірки ветеринарних рецептів безпосередньо в інтерфейсі обробки замовлення робить процес легітимним та безпечним для обох сторін.

Сучасний ринок зоотоварів тримається на регулярних покупках. Якщо система «знає», що клієнт купив 12-кілограмовий мішок корму для лабрадора,

					КВРІПЗ.2201113.01.18.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		7

вона може точно розрахувати час, коли запаси вичерпаються. Актуальність вебмодуля полягає у здатності накопичувати ці дані та автоматично пропонувати повторне замовлення саме тоді, коли воно потрібне. Це створює ефект «турботливого помічника», а не просто магазину. Такий підхід радикально знижує вартість утримання клієнта (Retention Rate), що в умовах дорогої реклами є ключовим фактором виживання бізнесу.

Без власного модуля обробки замовлень бізнес швидко впирається в «скляну стелю». Кількість помилок при ручному введенні даних зростає пропорційно кількості замовлень, а швидкість реакції менеджерів падає. Впровадження вебрішення дозволяє масштабувати продажі в десятки разів без значного розширення штату. В умовах, коли великі гравці ринку пропонують доставку «день у день», наявність інтегрованого модуля стає єдиним способом для середнього та малого ветеринарного бізнесу залишатися релевантним і не втратити свою частку ринку під натиском гігантів.

Отже, актуальність такої розробки виявляється у таких позитивних моментах:

- масштабованість та конкурентоспроможність;
- побудова довгострокової лояльності через дані;
- специфіка ветеринарного ритейлу як драйвер розробки;
- стратегічна трансформація бізнесу.

Зважаючи на актуальність використання вебзастосунків у різних галузях життєдіяльності людини, метою кваліфікаційної роботи є проведення проектування і розроблення вебмодуля для обробки замовлень у ветеринарному онлайн-магазині.

Враховуючи визначену мету варто визначити наступні завдання задля її досягнення, а саме:

- проаналізувати область продажу ветеринарних товарів, визначити особливості та межі використання функціоналу;
- проаналізувати існуючі вебмодулі для реалізації та обробки замовлень на ресурсах із продажу ветеринарних товарів;

					КвРІПЗ.2201113.01.18.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

- встановити плюси та мінуси існуючих розробок;
- здійснити постановку задачі із подальшою розробкою технічного завдання;
- вибрати тип архітектури для розробки програмного забезпечення;
- проаналізувати та вибрати інструментарій, засоби та технології розроблення програмного забезпечення;
- здійснити реліз та тестування розробленого програмного забезпечення.

					КВРІПЗ.2201113.01.18.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		9

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Виконання вебмодулю - це не просто перелік дій, а демонстрація вашої здатності проходити повний життєвий цикл розробки програмного забезпечення (SDLC). У контексті створення вебмодуля для ветеринарного магазину діяльність охоплює шлях від глибокого аудиту бізнес-процесів до технічної реалізації критичних функцій.

1. Аналітичний етап та обстеження об'єкта.

Першим етапом робіт став ґрунтовний аналіз поточної операційної діяльності бази практики. Вивчення того, як менеджери обробляють замовлення вручну, дозволило виявити критичні вразливості: затримки в перевірці рецептів та помилки при списанні товарів з обмеженим терміном придатності. На основі цих спостережень було сформовано технічне завдання, яке перетворює бізнес-проблеми на функціональні вимоги до вебмодуля. Окрему увагу було приділено дослідженню ринку ветеринарних препаратів для розуміння специфіки зберігання та відпуску специфічних категорій товарів.

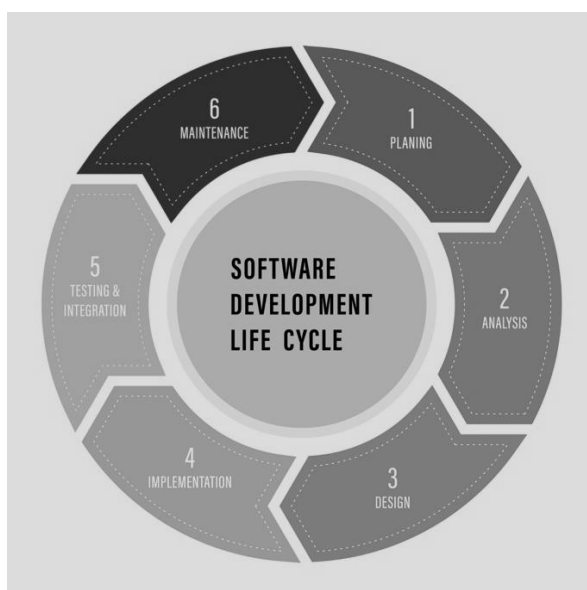


Рисунок 2.1 – Життєвий цикл програмного продукту

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		10

2. Проектування архітектури та бази даних.

Наступним кроком стало створення логічної структури майбутньої системи. Робота включала проектування схеми бази даних, де особливий акцент було зроблено на зв'язках між користувачами, їхніми улюбленцями та історією медичних призначень. Було розроблено архітектуру взаємодії між клієнтською частиною (Frontend) та серверною логікою (Backend), що забезпечує високу швидкість відгуку навіть при великій кількості одночасних запитів. На цьому етапі також було створено прототипи інтерфейсу (Wireframes), орієнтовані на мінімізацію кількості кліків менеджера при обробці одного замовлення.

3. Програмна реалізація та інтеграція.

Основний обсяг робіт припадає на безпосереднє написання коду. Було реалізовано модуль авторизації з різними рівнями доступу, систему динамічної фільтрації товарів за ветеринарними характеристиками та функціонал для завантаження й перевірки цифрових рецептів. Важливою частиною робіт стала інтеграція з зовнішніми сервісами через API: налаштування автоматичної генерації накладних для логістичних операторів та підключення платіжних шлюзів. Весь процес супроводжувався використанням систем контролю версій, що дозволяло зберігати цілісність коду та відстежувати прогрес розробки.

4. Тестування, безпека та підготовка документації.

Завершальний етап включав проведення комплексного тестування розробленого модуля. Було виконано модульне тестування окремих функцій, а також стрес-тестування системи на здатність витримувати пікові навантаження. Окремим блоком робіт стало налаштування безпеки: впровадження шифрування персональних даних та перевірка системи на стійкість до типових вебвразливостей. Результатом практики стала підготовка технічної документації для розробників та короткого посібника користувача для менеджерів магазину, що забезпечує легке впровадження продукту в реальну роботу підприємства.

Виконані роботи дозволяють не лише створити працездатний програмний продукт, а й довести його економічну доцільність. Кожен етап - від аналізу до

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

деплою - став частиною кваліфікаційної роботи, підтверджуючи її практичну цінність та технічну обґрунтованість.

Предметна область охоплює сферу електронної комерції у вузькоспеціалізованому сегменті товарів для тварин, що поєднує в собі ознаки класичного ритейлу та фармацевтичної діяльності. Основна особливість цієї галузі полягає в тому, що кінцевий споживач, тобто тварина та замовник, тобто власник розділені, що вимагає від системи здатності оперувати даними обох суб'єктів. На відміну від звичайних магазинів, ветеринарний маркетплейс працює з товарами підвищеного рівня відповідальності - лікувальними дієтами та медикаментами, що накладає додаткові зобов'язання щодо контролю якості, термінів придатності та верифікації дозволів на покупку рецептів.

Структура предметної області базується на складній багатосторонній моделі даних. В її центрі знаходиться не просто замовлення, а трикутник взаємодії *Клієнт - Улюбленець - Товар*. Кожна одиниця товару в системі має розширений набір атрибутів, таких як фармакологічна група, активна речовина та сумісність із певними видами тварин.

Важливою структурною ланкою є інтеграція з ветеринарними базами даних, що дозволяє пов'язувати історію покупок із конкретними фізіологічними потребами тварини (вага, вік, хронічні захворювання). Організаційна структура також включає складські одиниці з особливими режимами зберігання та логістичні ланцюжки, що мають підтримувати цілісність пакування та специфічні умови транспортування.

Функціональна складова системи визначається динамічним життєвим циклом замовлення, який значно складніший за стандартну модель «оплата - відвантаження». Ключовою особливістю є функція валідації спеціалізованих запитів, де система автоматично перевіряє наявність підтверджувальних документів для рецептурних препаратів.

Інша функціональна риса - це механізм предиктивного аналізу попиту. Система повинна автоматично розраховувати цикли споживання (наприклад, тривалість використання пакування корму або курсу лікування паразитів) і

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

генерувати персоналізовані пропозиції для клієнта. Це створює замкнений цикл обслуговування, де функція обробки замовлень плавно переходить у функцію підтримки здоров'я тварини. Крім того, функціонал передбачає складну систему звітності, яка дозволяє відстежувати рух кожної серії препарату, забезпечуючи повну прозорість діяльності згідно з державними нормами контролю за обігом ветеринарних засобів.

Таким чином, аналіз предметної області демонструє, що вебмодуль для обробки замовлень ветаптеки є високотехнологічним рішенням, яке має інтегрувати медичні вимоги у комерційні процеси. Це вимагає створення гнучкої архітектури, здатної адаптуватися до мінливих потреб власників тварин та суворих стандартів ветеринарної медицини.

Інформаційне забезпечення - це сукупність єдиної системи класифікації та кодування інформації, уніфікованих систем документації та масивів інформації, що використовуються в системі. У нашому випадку воно організоване так, щоб забезпечити безперебійний цикл відвідування магазину та лікування тварини.

Вся інформація в модулі поділяється на три потоки: вхідна, вихідна та внутрішня інформація.

Вхідна інформація містить дані про клієнта, параметри домашнього улюбленця, такі як вид, вага, вік, цифрові копії рецептів, а також дані від постачальників про нові надходження та терміни придатності препаратів.

Внутрішня нормативно-довідкова інформація - це каталоги ветеринарних засобів, класифікатори хвороб, довідники дозувань, а також статусна модель замовлень.

Вихідна інформація – це сформовані ТТН, рахунки-фактури, листи-підтвердження для клієнтів, аналітичні звіти про реалізацію та акти списання залишків.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

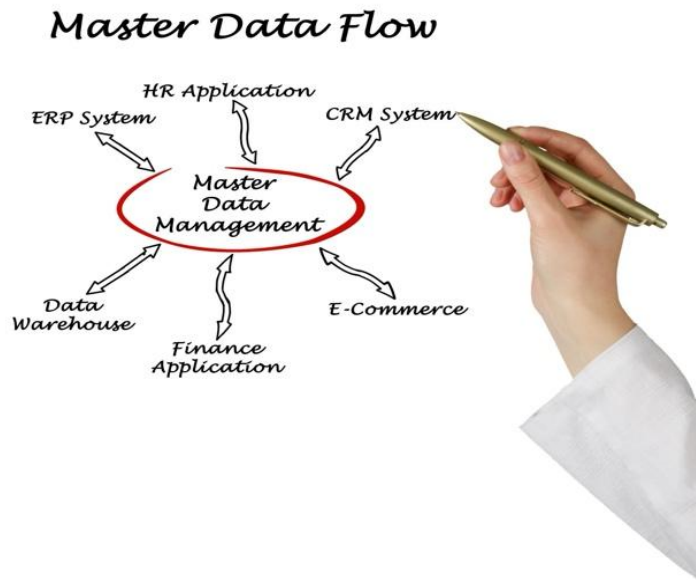


Рисунок 1.2 - Організація бази даних

Інформаційна модель будується на принципах реляційної цілісності. Це означає, що кожне замовлення жорстко прив'язане не лише до покупця, а й до конкретної тварини в його профілі. Це дозволяє системі «розуміти» контекст: якщо купується препарат для котів, а у профілі вказано лише собаку, система автоматично пропонує менеджеру уточнити деталі замовлення. Такий рівень інформаційної підтримки мінімізує помилки, які в ветеринарії можуть бути фатальними.

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Для обґрунтування розробки власного модуля необхідно порівняти існуючі рішення. На сьогодні ринок пропонує три основні типи систем, кожна з яких має суттєві обмеження для ветеринарного ритейлу.

Порівняльна характеристика технологічних рішень наводиться таблиці 1.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14

Таблиця 1.1 – Порівняльна характеристика технологічних рішень

Тип системи	Приклади	Переваги	Недоліки для ветаптеки
Універсальні CRM (SaaS)	Bitrix24, Zoho	Потужні інструменти продажів, воронки.	Повна відсутність профілів тварин та контролю за рецептами.
Спеціалізовані медичні ІС	VetManager, ezyVet	Глибокий медичний облік, карти пацієнтів.	Слабкий функціонал саме інтернет-магазину (кошик, логістика, ТТН).
Е-commerce платформи	Shopify, Magento	Найкращий UX для покупок, SEO-оптимізація.	Складність у впровадженні специфічної ветеринарної логіки (терміни придатності серій).

Аналіз показує, що існуючі аналоги або занадто «загальні» (не розуміють специфіки ліків), або занадто «клінічні» (не зручні для швидкої торгівлі). Більшість готових рішень вимагають дорогої кастомізації для того, щоб просто додати поле «дата наступної вакцинації» або реалізувати завантаження фото рецепта з мобільного телефону.

Вебмодуль займає унікальну нішу «Retail-Vet Tech» та поєднує швидкість і зручність сучасного інтернет-магазину з суворим контролем ветеринарних стандартів. Це дозволяє отримати інструмент, який не просто «продає», а веде повноцінний супровід клієнта та його улюбленця.

Сучасний ринок електронної комерції у сфері зоотоварів та ветеринарії можна умовно поділити на три основні категорії: великі маркетплейси

загального призначення, вузькоспеціалізовані локальні ветеринарні аптеки та масштабні міжнародні платформи. Кожен із цих підходів має свої архітектурні особливості.

Великі вітчизняні зоомагазини, наприклад, аналоги Pethouse або MasterZoo пропонують високий рівень користувацького досвіду (UX), швидке завантаження сторінок та надійну архітектуру. Їхньою головною перевагою є відлагоджений процес оформлення замовлення, інтеграція з популярними платіжними системами та службами доставки. Проте, з точки зору інженерії програмного забезпечення для специфічної ніші, вони мають суттєвий недолік. Їхня база даних та система фільтрації оптимізовані під загальні товари. Користувачам складно підібрати ліки за діючою речовиною, точним дозуванням на кілограм ваги тварини або за специфічними протипоказаннями, оскільки реляційна модель їхніх баз даних не передбачає таких зв'язків для кожної одиниці товару.

З іншого боку, існують спеціалізовані онлайн-ветераптеки. Вони володіють правильним асортиментом та професійною категоризацією товарів (антигельмінтні, вакцини, антибіотики тощо). Незважаючи на глибину предметної області, більшість таких рішень побудовані на застарілих монолітних архітектурах або коробкових CMS-системах. Це призводить до низької продуктивності при масштабуванні, вразливостей у безпеці та відсутності сучасного Single Page Application (SPA) підходу для мобільних пристроїв. Крім того, у них не реалізовані складні бізнес-правила, як-от перевірка наявності рецепта від лікаря перед додаванням товару до кошика.

Як еталонне рішення на міжнародному ринку варто виділити медичний розділ американської платформи Chewy. Ця система базується на мікросервісній архітектурі, де реалізовано високошвидкісний повнотекстовий пошук, зазвичай на базі Elasticsearch, можливість створення профілю улюбленця з історією хвороб та автоматизований модуль перевірки рецептів. Головним недоліком таких еталонних систем є їхня регіональна замкнутість, складність адаптації під українське законодавство щодо обігу ветеринарних

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		16

препаратів та відсутність відкритого API для локальних інтеграцій. У таблиці 2 подано порівняльну характеристику існуючих рішень.

Таблиця 2.1 - Порівняльна характеристика існуючих програмних рішень

Категорія платформи	Основні архітектурні та бізнес-переваги	Ключові недоліки та обмеження
Універсальні зоомагазини	Висока швидкодія, продуманий UI/UX, надійна інтеграція з платіжними шлюзами, висока відмовостійкість.	Відсутність спеціалізованої медичної фільтрації, неможливість валідації рецептурних ліків.
Локальні ветераптеки	Точна категоризація товарів за медичним призначенням, наявність професійних описів препаратів.	Застарілий технологічний стек, низька продуктивність, відсутність адаптивного дизайну, складнощі з масштабуванням бази даних.
Міжнародні гіганти (Chewy)	Сучасна мікросервісна архітектура, профілі тварин, автоматична перевірка рецептів, система підписок на ліки.	Недоступність для локального ринку, закритість системи, невідповідність місцевим протоколам доставки та оплати.

Підсумовуючи, можна зазначити, що на локальному ринку існує дефіцит рішень, які б поєднували сучасний технологічний стек (швидкість, безпеку, масштабованість) із глибоким розумінням предметної області ветеринарії

(робота з рецептами, специфічні фільтри). Саме цю прогалину має закрити вебсервіс, що розробляється в рамках даної кваліфікаційної роботи.

Загалом аналіз повинен спиратися на реальні системи та конкретні програмні рішення, що продемонстровано на рисунках 1.1-1.5.

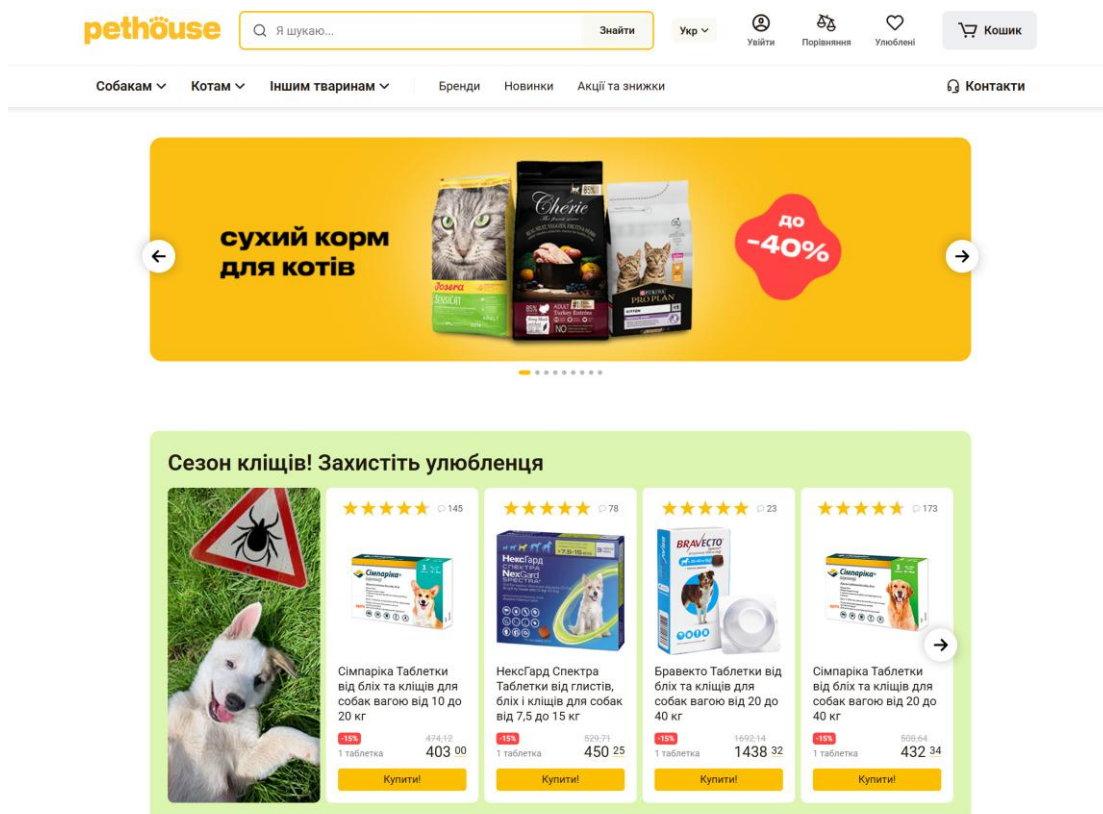


Рисунок 1.1 - Pethouse.ua

Ця платформа є яскравим представником універсальних вітчизняних зоомагазинів. Архітектурно сайт побудований з акцентом на швидкість завантаження каталогу та зручний кошик, що забезпечує високу конверсію. Проте ветеринарний розділ реалізований як звичайна категорія товарів. Тут відсутні спеціалізовані сутності в базі даних для медичних препаратів, тому користувач не може відфільтрувати ліки за симптомами, діючою речовиною чи віковими обмеженнями тварини. Система не передбачає механізмів авторизації ветеринарних рецептів.

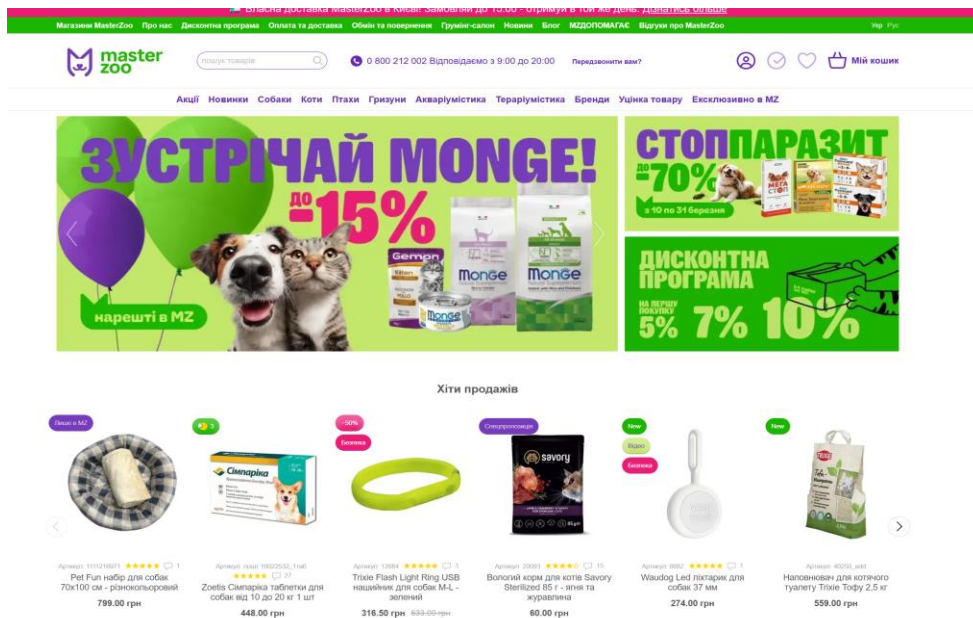


Рисунок 1.2 - MasterZoo.ua

Платформа представляє велику омніканальну мережу. З інженерної точки зору перевагою системи є складна інтеграція з офлайн-складами (ERP-системами), що дозволяє відображати актуальні залишки в реальному часі. У них є виділений розділ «Ветаптека», однак медична логіка залишається спрощеною. Як і в попередньому випадку, архітектура не оптимізована під специфіку фармацевтики: немає інструментів для розрахунку дозування на основі ваги улюбленця або попереджень про несумісність препаратів.

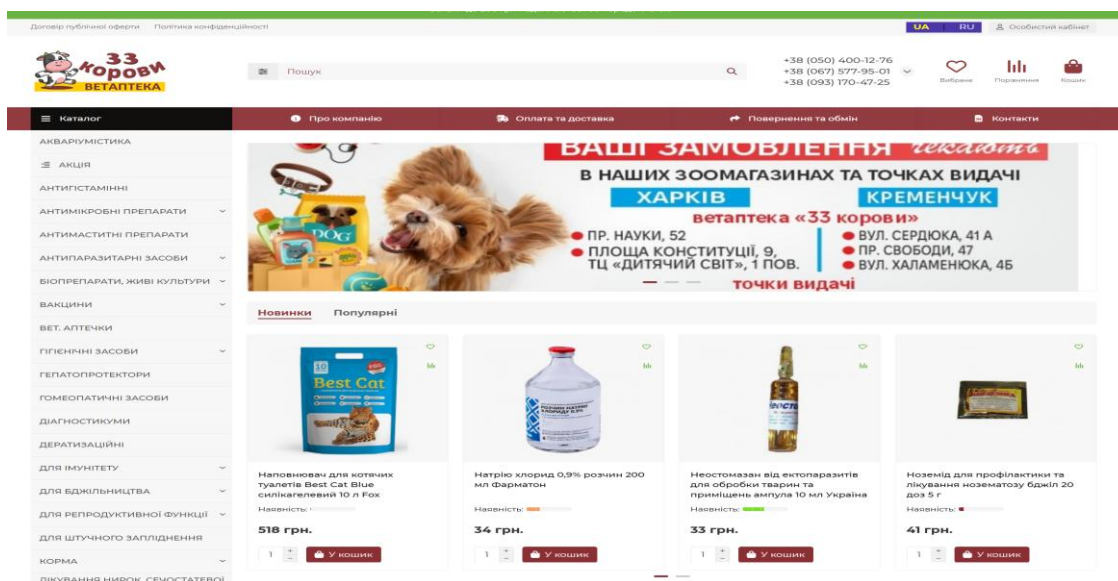


Рисунок 1.3 - 33korumy.in.ua

									Арк.
									19
Зм.	Арк	№ докум.	Підпис	Дата					

КВРПЗ.200169.01.19.ПЗ

Один із найвідоміших спеціалізованих українських ресурсів, що функціонує саме як ветеринарна аптека. З точки зору бізнес-логіки тут чудово опрацьована категоризація: від вакцин до хірургічних інструментів. Однак програмна реалізація має ознаки застарілого монолітного підходу. Інтерфейс користувача не є адаптивним Single Page Application (SPA), що погіршує досвід на мобільних пристроях. Крім того, відсутність сучасних механізмів кешування та оптимізації пошукових запитів створює проблеми з продуктивністю під час пікових навантажень.

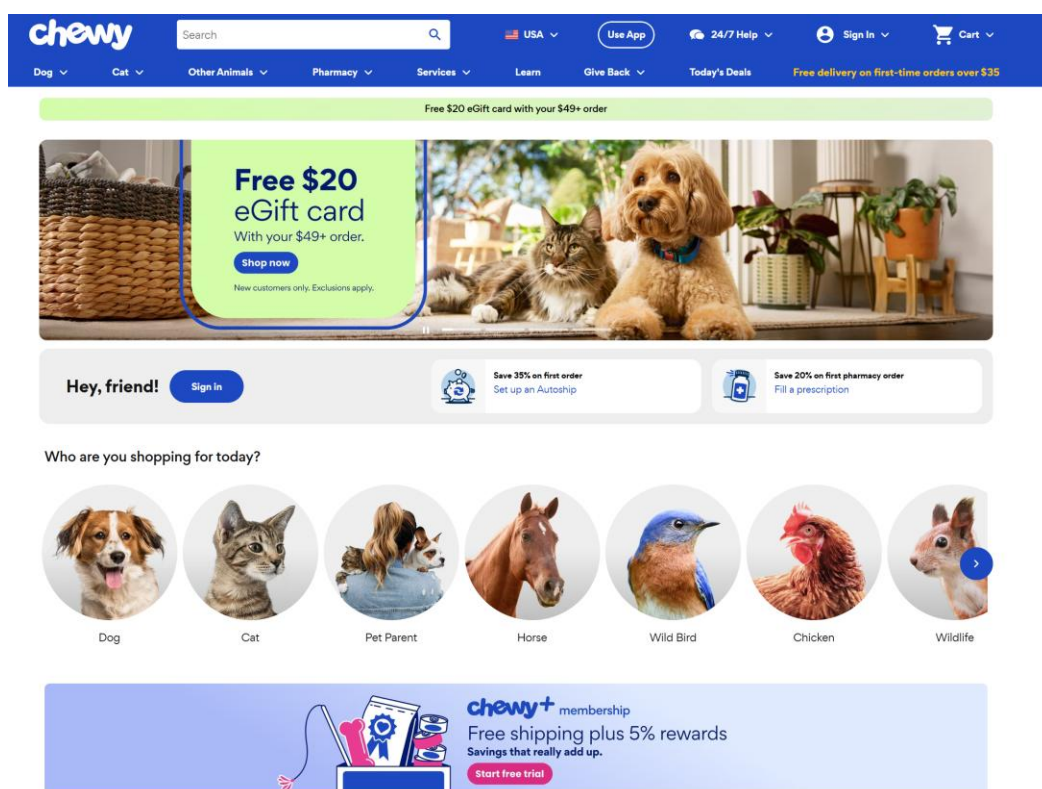


Рисунок 1.4 - Chewy.com (США)

Міжнародний лідер електронної комерції для тварин, чий розділ «Chewy Pharmacy» є еталоном для розробки подібних сервісів. Система побудована на мікросервісній архітектурі, що гарантує максимальну відмовостійкість. Головна перевага - наявність сутності «Профіль улюбленця» (Pet Profile), з якою пов'язані електронні рецепти від ветеринарів. Система автоматично валідує рецепт перед тим, як дозволити транзакцію. Недоліком є неможливість прямого

									Арк.
Зм.	Арк	№ докум.	Підпис	Дата					20

використання їхнього архітектурного шаблону в українських реаліях через різницю в законодавстві та закритість вихідного коду.

Вузькоспеціалізована американська онлайн-ветаптека 1800petmeds.com (США), яка фокусується виключно на здоров'ї тварин. Їхня база даних спроектована з урахуванням медичних довідників: кожен препарат має чіткі зв'язки з побічними ефектами, дозуваннями та аналогами. Платформа використовує розширені алгоритми рекомендацій для нагадування про необхідність повторної покупки хронічних ліків або регулярної обробки від паразитів. Основний мінус для вітчизняного ринку - відсутність локалізації та інтеграції з українськими медичними інформаційними системами.

1.3 Визначення функціональних та нефункціональних вимог

Аналіз вимог є критичним етапом, оскільки він визначає межі проєкту та критерії його якості. Для ветеринарного вебмодуля вимоги мають враховувати як бізнес-логіку продажів, так і специфіку роботи з медичними даними.

Функціональні вимоги визначають, що саме система повинна робити для користувачів. До таких вимог належать:

- управління профілями тварин, тобто можливість створення карток для кожної тварини клієнта з вказанням виду, ваги, віку та алергій для автоматичної перевірки сумісності з товарами;
- валідація рецептурних замовлень через блокування переходу до оплати для товарів категорії *Ветеринарні препарати* до моменту завантаження клієнтом фото рецепта;
- інтеграція з ПРРО, тобто автоматична фіскалізація чеків через сервіси, наприклад, Checkbox у момент підтвердження оплати замовлення;
- автоматизація логістики через генерацію експрес-накладних (ТТН) через АРІ «Нової Пошти» або «Укрпошти» безпосередньо з адмін-панелі;

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		21

– контроль залишків за термінами (FEFO), тобто система повинна автоматично пропонувати до відвантаження одиниці товару з найближчою датою закінчення терміну придатності.

– система сповіщень через автоматичне надсилання статусів замовлення через SMS, Email або месенджери (Viber/Telegram).

Нефункціональні вимоги визначають, як система повинна працювати (якість, надійність, безпека).

До таких вимог належать:

- продуктивність та масштабованість;
- безпека та конфіденційність;
- надійність та доступність.

Продуктивність та масштабованість реалізується через час відгуку та навантаження:

- час відгуку, коли швидкість відповіді сервера на запити користувача не повинна перевищувати 200 мс для 95% операцій.
- навантаження, коли система має підтримувати одночасну роботу щонайменше 50 активних менеджерів та 500 одночасних сесій покупців без деградації продуктивності.

Безпека та конфіденційність:

- шифрування через обов'язкове використання протоколу HTTPS для всіх з'єднань та шифрування персональних даних у базі даних;
- авторизація через розмежування прав доступу - комірник не повинен бачити фінансову звітність, а клієнт - дані інших користувачів.

Надійність та доступність:

- доступність, тобто система повинна бути доступною 99.9% часу (не більше 9 годин простою на рік);
- відновлення, тобто час повного відновлення системи з бекапу не повинен перевищувати 2 години.

Зручність використання:

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		22

- адаптивність, тобто інтерфейс повинен коректно відображатися на десктопах, планшетах та смартфонах;
- інклюзивність, тобто контрастність тексту та розміри елементів керування мають відповідати стандартам для зручності користувачів із вадами зору.

Доцільно показати, як нефункціональні вимоги впливають на технічні рішення, що детально показано у таблиці 3.

Таблиця 3.1 – Матриця відповідності вимог

Вимога	Технічне рішення
Висока швидкість	Використання Node.js (Event Loop) та індексація БД PostgreSQL.
Безпека даних	Впровадження JWT-токенів та Middleware валідації запитів.
Надійність	Автоматичне резервне копіювання на хмарне сховище щоночі.

Чіткий розподіл на функціональні та нефункціональні вимоги дозволяє уникнути хаосу при розробці та забезпечити базу практики продуктом, який не тільки виконує свої завдання, але й є безпечним та приємним у використанні.

Таблиця 4.1 - Порівняльна таблиця характеристик вимог

Категорія	Опис вимоги	Пріоритет
Функціональна	Створення профілю тварини з історією хвороб	Високий
Функціональна	Фільтрація ліків за діючою речовиною та дозуванням	Високий
Функціональна	Валідація завантажених ветеринарних рецептів	Середній

Нефункціональна	Час безперебійної роботи (Uptime) не менше 99.5%	Високий
Нефункціональна	Підтримка стандартів захисту персональних даних (GDPR)	Високий
Нефункціональна	Використання мікросервісного або модульного підходу	Середній

Для встановлення та чіткої візуалізації визначених вимог можна використовувати інструментарій UML. На рисунку 2.3 показано діаграму варіантів використання для адміністратора.



Рисунок 2.3 – Діаграма варіантів використання для адміністратора

Рисунок 2.4 демонструє діаграму варіантів використання для зареєстрованого та незареєстрованого користувача

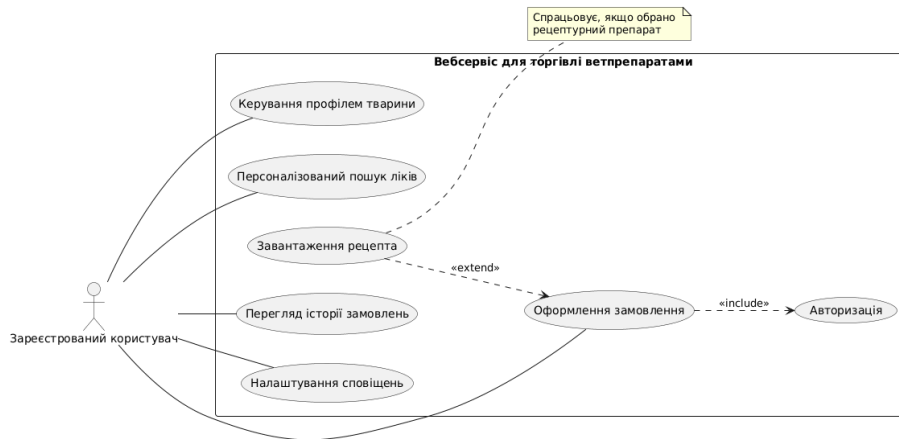


Рисунок 2.3 – Діаграма варіантів використання для зареєстрованого користувача

Щоб краще відобразити процес моделювання та аналізу вимог можна використовувати діаграму станів (рисунок 2.4).

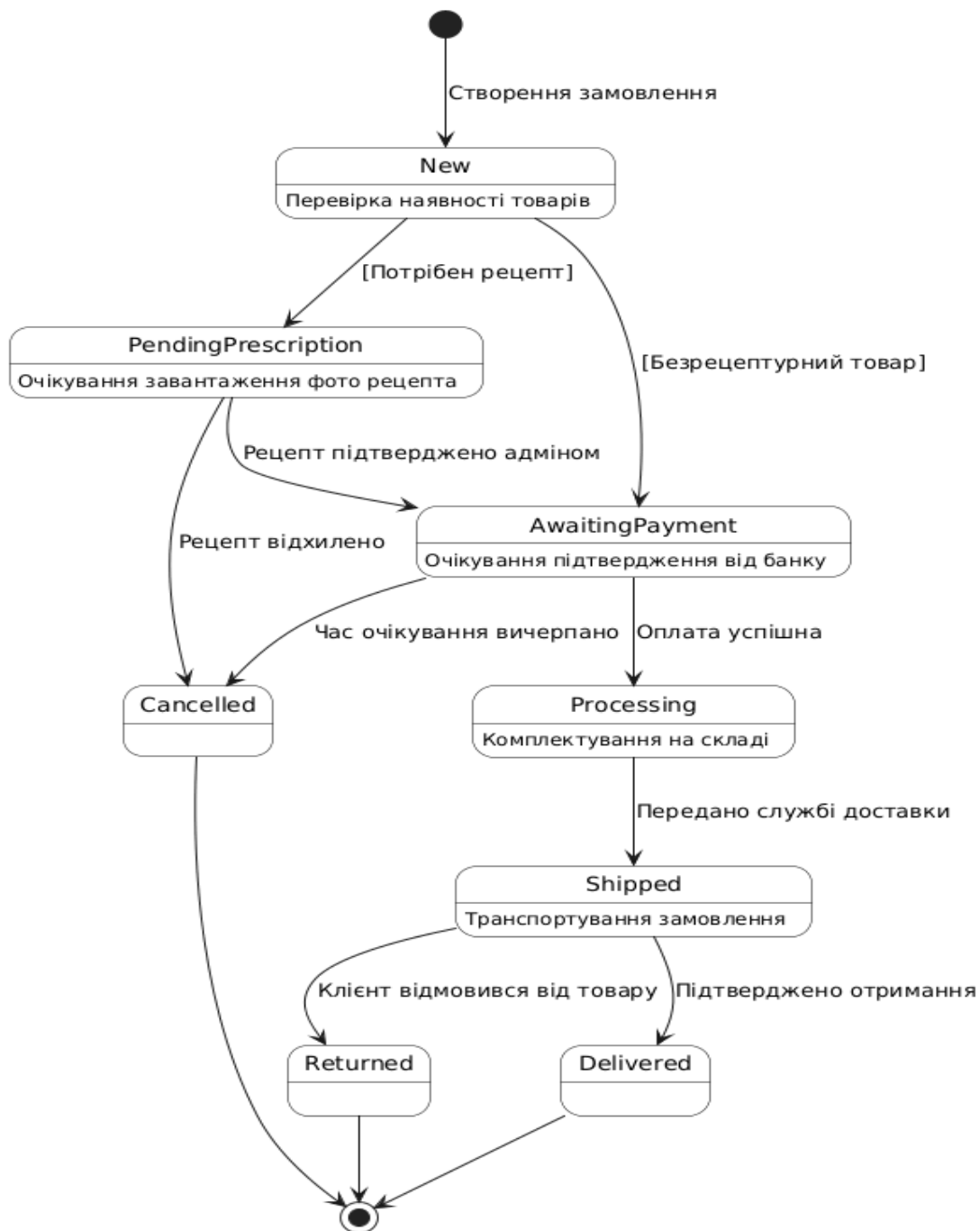


Рисунок 2.4 – Діаграма станів

Отже, діаграми варіантів використання є ключовим інструментом візуалізації вимог. Вона демонструє межі системи та взаємодію користувача з її функціональними модулями. Для зареєстрованого користувача, тобто клієнта ветеринарного вебсервісу логіка взаємодії будується навколо трьох хабів: керування профілем тварини, процес покупки та робота з медичними даними.

Діаграми станів відображають життєвий цикл конкретного об'єкта, стан якого змінюється під впливом зовнішніх подій. Для розроблюваного проєкту найбільш змістовним об'єктом є Замовлення (Order), оскільки воно має складну логіку переходів, включаючи перевірку рецептів та оплату.

2.4. Висновки. Постановка задачі

Виходячи із всього описаного вище, можна визначити, що формування вимог - це фундамент проєкту, оскільки саме на них ви будете посилалися під час тестування системи в кінці роботи. У програмній інженерії вимоги поділяють на функціональні, тобто що система робить та нефункціональні - як вона це робить.

Функціональні вимоги визначають конкретні сервіси та можливості, які система повинна надавати користувачам різних ролей (клієнт, адміністратор, ветеринарний консультант).

Для ветеринарного вебсервісу критично важливою є реалізація модуля управління каталогом, який дозволяє не лише переглядати товари, а й фільтрувати їх за специфічними атрибутами: видом тварини, віковою групою, діючою речовиною та лікарською формою. Система повинна підтримувати механізм створення та редагування Профілю тварини, де зберігаються дані про вагу та алергії улюбленця для автоматичної перевірки сумісності обраного препарату.

Модуль оформлення замовлення має включати функціонал завантаження скан-копій або фото ветеринарних рецептів для препаратів суворого обліку. Адміністративна панель повинна забезпечувати керування складськими залишками, генерацію звітів про продажі та модерацію відгуків. Важливим елементом є система сповіщень, яка автоматично нагадує користувачеві про необхідність повторної обробки тварини від паразитів через певний проміжок часу.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		27

Нефункціональні вимоги описують обмеження та якісні характеристики системи. З точки зору продуктивності, час відклику сервера на запити пошуку товарів не повинен перевищувати 500 мс при одночасній роботі до 100 активних сесій. Надійність системи має забезпечуватися регулярним резервним копіюванням бази даних та механізмами відновлення після збоїв.

Щодо безпеки, вебсервіс зобов'язаний використовувати протокол HTTPS для шифрування трафіку, а автентифікація користувачів має здійснюватися за допомогою безпечних токенів, наприклад, JWT, щоб уникнути витоку персональних даних власників тварин. Масштабованість архітектури повинна дозволяти легке додавання нових модулів, наприклад, системи онлайн-консультацій без необхідності повного переписування ядра системи. Нарешті, інтерфейс має відповідати принципам адаптивності, коректно відображаючись на пристроях із різною роздільною здатністю екрана.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		28

2 ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1 Проектування архітектури та структури вебзастосунку

Проектування архітектури є центральною частиною розробки програмного забезпечення, оскільки саме на цьому етапі визначається здатність системи до масштабування, підтримки та високої доступності. Для ветеринарного вебсервісу найбільш доцільним є використання багатoshарової сервіс-орієнтованої архітектури з чітким розділенням відповідальності за принципами Clean Architecture.

Концептуальна структура системи полягає у тому, що в основі архітектурного рішення лежить принцип розділення фронтенд-частини та бекенд-логіки через API-інтерфейс. Клієнтська частина реалізується як Single Page Application (SPA), що забезпечує безперервну взаємодію з користувачем без перезавантаження сторінок. Концепція SPA базується на розробці вебзастосунку, який завантажує лише один HTML-документ і динамічно оновлює його вміст у відповідь на дії користувача. На відміну від традиційного підходу, де кожна взаємодія з інтерфейсом призводить до запиту нової сторінки від сервера, SPA використовує JavaScript для перемальовування окремих частин сторінки. Це дозволяє досягти високої плавності інтерфейсу, що за відчуттями нагадує роботу з нативним мобільним додатком, де переходи між розділами відбуваються миттєво та без видимих пауз на перезавантаження.

Процес навігації в межах такого застосунку реалізується за допомогою механізму клієнтської маршрутизації. Коли користувач ініціює перехід, спеціальний програмний модуль перехоплює подію зміни URL-адреси та самостійно визначає, який набір компонентів має бути відображений на екрані. При цьому сервер залишається осторонь процесу формування візуальної частини й виступає лише як джерело «сирих» даних. Такий розподіл відповідальності робить архітектуру чистою: фронтенд відповідає за

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		29

представлення та логіку відображення, а бекенд зосереджується виключно на бізнес-логіці та безпеці даних.

Ефективність взаємодії з сервером досягається завдяки асинхронним запитам, які передають дані у форматі JSON. Це суттєво зменшує обсяг трафіку, оскільки браузер більше не отримує повторювані елементи розмітки, такі як шапка сайту або навігаційні блоки, при кожному кліку. У контексті ветеринарного сервісу це забезпечує стабільну роботу навіть при завантаженні складних об'єктів, наприклад, при одночасній зміні фільтрів за видом тварини та ваговою категорією в каталозі препаратів.

Це особливо важливо для функціонала фільтрації ліків за складними параметрами. Бекенд-частина проєктується як модульний моноліт або набір мікросервісів, що дозволяє ізолювати критичні бізнес-процеси, такі як обробка платежів та валідація медичних рецептів, від загального каталогу товарів.

Взаємодія між компонентами відбувається через RESTful API, де кожен запит проходить через шар проміжного програмного забезпечення або так званого Middleware для перевірки прав доступу та валідації вхідних даних. Для забезпечення високої швидкодії системи на рівні архітектури передбачено впровадження шару кешування, наприклад, Redis, що дозволяє миттєво віддавати популярні запити каталогу препаратів, зменшуючи навантаження на основну реляційну базу даних.

Опис архітектурних рівнів включає рівень представлення, бізнес-логіки, рівень інфраструктури. Рівень представлення відповідає за візуалізацію даних та збір інформації від користувача. Він взаємодіє виключно з рівнем API, не маючи прямого доступу до бази даних. Це гарантує безпеку системи, оскільки бізнес-логіка прихована за інтерфейсом програмування застосунків.

Рівень бізнес-логіки містить ключові сутності системи: Препарат, Тварина, Рецепт та Замовлення. Тут реалізуються алгоритми перевірки сумісності ліків та логіка переходу замовлення між станами. Використання патерну Dependency Inversion дозволяє зробити цей шар незалежним від конкретних баз даних або зовнішніх сервісів оплати.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		30

Рівень інфраструктури забезпечує зв'язок із зовнішнім світом. Він включає адаптери для роботи з PostgreSQL, інтеграцію з хмарними сховищами для збереження фотографій рецептів та модулі для надсилання сповіщень. Такий підхід дозволяє легко змінити, наприклад, постачальника послуг доставки або платіжний шлюз, не зачіпаючи основний код програми.

Для реалізації обраної архітектури пропонується використовувати мову TypeScript як на фронтенді, так і на бекенді (Node.js/NestJS). Це забезпечує типізацію даних на всіх етапах розробки, що значно знижує кількість помилок при передачі об'єктів між сервісами. Як основне сховище даних обирається PostgreSQL завдяки її підтримці складних зв'язків та транзакційної цілісності, що є критичним для фармацевтичного обліку. Для контейнеризації всієї інфраструктури використовується Docker, що гарантує ідентичність середовища розробки та продуктового сервера, спрощуючи процес розгортання та масштабування.

Рисунок 2.1 демонструє архітектуру спроектованого рішення.

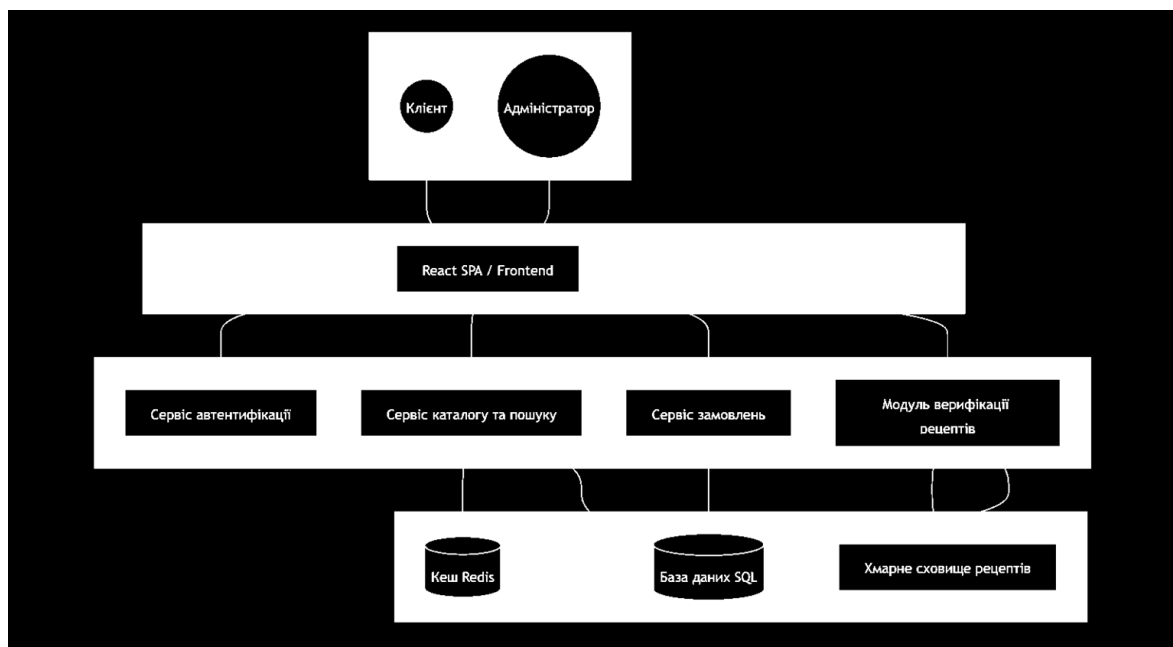


Рисунок 2.1 – Архітектура спроектованого рішення

2.2 Проектування логічної моделі бази даних

Проектування бази даних є критичним етапом, оскільки від структури збереження даних залежить продуктивність пошуку та цілісність бізнес-логіки. Для ветеринарного вебсервісу реляційна модель PostgreSQL є найбільш виправданою, оскільки вона дозволяє чітко визначити зв'язки між користувачами, їхніми улюбленцями та специфічними фармацевтичними вимогами.

Центральною сутністю системи є таблиця Користувачі, яка пов'язана із таблицею Тварини відношенням один до багатьох. Це дозволяє одному власнику мати кілька профілів улюбленців, що є ключовим для персоналізації підбору ліків. Кожна картка тварини містить атрибути ваги та виду, які використовуються системою для фільтрації допустимих препаратів.

Таблиця Товари (Препарати) має складну структуру атрибутів, включаючи прапорець `is_prescription_required`. При оформленні замовлення, якщо в кошику є рецептурний препарат, система створює запис у таблиці Рецепти, що зв'язує конкретне замовлення з файлом підтвердження. Самі замовлення декомпонуються на таблиці Замовлення та Позиції замовлення для забезпечення нормалізації та збереження ціни товару на момент покупки.

Для забезпечення цілісності даних у проєкті застосовано третю нормальну форму 3NF. Це мінімізує надмірність даних, наприклад, через винесення категорій в окрему таблицю та використання зовнішніх ключів (FK) для зв'язків між замовленнями та клієнтами. Особлива увага приділена таблиці ORDER_ITEMS: вона зберігає `price_at_purchase`, що дозволяє уникнути змін у фінансовій звітності при зміні поточної ціни товару в основному каталозі.

Зв'язок між PETS та ORDERS є опціональним, але архітектурно важливим. Він дозволяє системі вести аналітику: які ліки найчастіше купують для конкретних порід чи видів тварин, що в майбутньому дозволить впровадити систему інтелектуальних рекомендацій.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		32

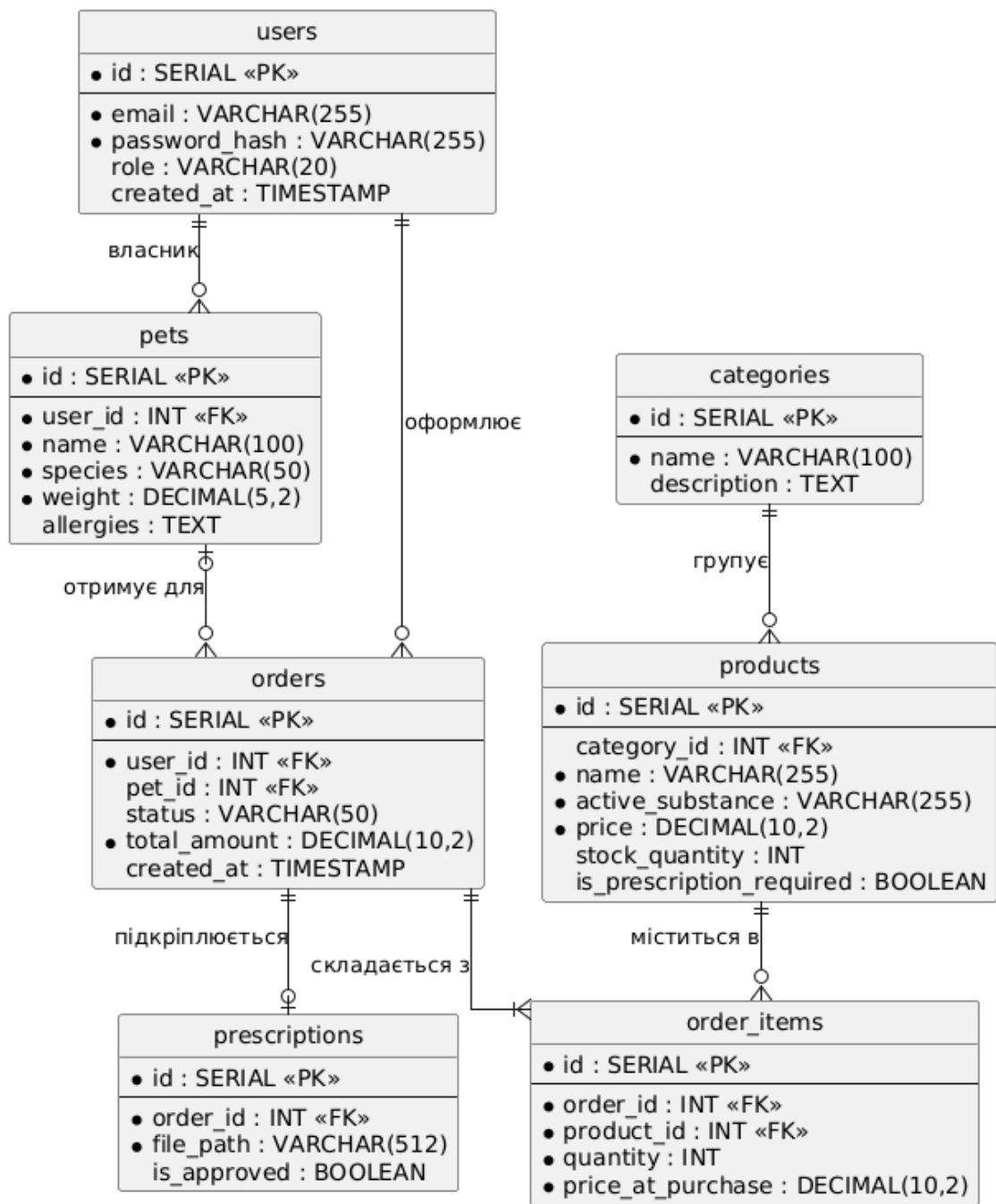


Рисунок 2.5 – ER-діаграма бази даних

Нижче подано детальний опис таблиць, з яких складається база даних розроблюваної системи.

Таблиці можна розподілити за блоками. До першого блоку Структура сутностей користувачів та тварин відносяться дані, що складають основу

системи авторизації та дозволяють реалізувати персоналізацію контенту на основі профілю тварини.

Таблиця 2.1 – users

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
users	id	SERIAL	PRIMARY KEY	Унікальний ідентифікатор
	email	VARCHAR(255)	UNIQUE, NOT NULL	Електронна пошта (логін)
	password_hash	VARCHAR(255)	NOT NULL	Зашифрований пароль
	role	VARCHAR(20)	DEFAULT 'USER'	Роль (ADMIN, USER)
	created_at	TIMESTAMP	DEFAULT NOW()	Дата реєстрації

Таблиця 2.2 – pets

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
pets	id	SERIAL	PRIMARY KEY	Ідентифікатор тварини
	user_id	INT	FOREIGN KEY (users.id)	Власник тварини
	name	VARCHAR(100)	NOT NULL	Кличка тварини
	species	VARCHAR(50)	NOT NULL	Вид (кіт, собака тощо)

	weight	DECIMAL(5,2)	NOT NULL	Вага для дозування (кг)
	allergies	TEXT	NULLABLE	Медичні застереження

Таблиці 2.3-2.4 показують структуру товарного каталогу та забезпечують зберігання інформації про препарати та їх категоризацію за медичними властивостями.

Таблиця 2.3 – categories

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
categories	id	SERIAL	PRIMARY KEY	Ідентифікатор категорії
	name	VARCHAR(100)	UNIQUE, NOT NULL	Назва групи (напр. Вакцини)
	description	TEXT	NULLABLE	Опис застосування

Таблиця 2.4 – products

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
products	id	SERIAL	PRIMARY KEY	Ідентифікатор товару
	category_id	INT	FOREIGN KEY (categories.id)	Посилання на категорію

	name	VARCHAR(255)	NOT NULL	Назва препарату
	substance	VARCHAR(255)	NOT NULL	Діюча речовина
	price	DECIMAL(10,2)	NOT NULL	Актуальна ціна
	is_prescription	BOOLEAN	DEFAULT FALSE	Вимога рецепта

За транзакційну частину системи та контроль за рецептурними препаратами відповідає блок структура процесу продажу та верифікації (таблиці 5-6).

Таблиця 2.5 – orders

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
orders	id	SERIAL	PRIMARY KEY	Номер замовлення
	user_id	INT	FOREIGN KEY (users.id)	Посилання на покупця
	status	VARCHAR(50)	DEFAULT 'NEW'	Стан (Paid, Shipped...)
	total_amount	DECIMAL(10,2)	NOT NULL	Загальна вартість
	created_at	TIMESTAMP	DEFAULT NOW()	Час оформлення

Таблиця 2.6 – order_items

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
order_items	id	SERIAL	PRIMARY KEY	Ідентифікатор позиції
	order_id	INT	FOREIGN KEY (orders.id)	Зв'язок із замовленням
	product_id	INT	FOREIGN KEY (products.id)	Зв'язок із товаром
	quantity	INT	NOT NULL	Кількість одиниць
	fixed_price	DECIMAL(10,2)	NOT NULL	Ціна на момент покупки

Таблиця 2.7 – prescriptions

Назва таблиці	Поле	Тип даних	Обмеження (Constraints)	Опис
prescriptions	id	SERIAL	PRIMARY KEY	Ідентифікатор рецепта
	order_id	INT	FOREIGN KEY (orders.id)	Посилання на замовлення
	file_path	VARCHAR(512)	NOT NULL	Шлях до зображення
	is_approved	BOOLEAN	DEFAULT FALSE	Статус перевірки

Ці таблиці є нормалізованими та готовими до впровадження. Такий поділ дозволяє чітко пояснити архітектуру даних.

2.3 Проектування інтерфейсу користувача

Проектування інтерфейсу користувача або UI/UX є сполучною ланкою між технічною архітектурою системи та кінцевим споживачем. Для ветеринарного вебсервісу цей етап зосереджений на створенні середовища, яке мінімізує когнітивне навантаження на власника тварини, що часто перебуває у стані стресу через хворобу улюбленця. Основний акцент робиться на інтуїтивності навігації та швидкості доступу до критично важливих функцій: пошуку ліків, завантаження рецептів та редагування медичного профілю тварини.

Загалом проектування інтерфейсу включає кілька етапів:

- аналіз інформаційної архітектури та сценаріїв взаємодії;
- розробку прототипів та вибір компонентної бази;
- візуальний дизайн та адаптивність.

Процес розробки розпочинається з побудови карти сайту та аналізу типових шляхів користувача. На відміну від звичайних інтернет-магазинів, тут додається специфічний крок - вибір активного профілю тварини. Це дозволяє системі динамічно підлаштовувати контент, відображаючи лише ті препарати, що відповідають виду та ваговій категорії обраного улюбленця. Інформаційна архітектура проєкту будується за ієрархічним принципом, де головна сторінка слугує хабом для переходу до категорій препаратів або особистого кабінету.

На етапі створення прототипів низької точності визначається композиція екранів та розташування ключових елементів керування. Особлива увага приділяється формі оформлення замовлення, яка повинна бути адаптивною,

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38

тобто, якщо у кошику є рецептурний препарат, інтерфейс має динамічно відобразити поле для завантаження документа. Для забезпечення візуальної цілісності та прискорення розробки використовується компонентний підхід, де кожен елемент інтерфейсу є перевикористовуваним модулем (таблиця 8).

Таблиця 2.8 - Характеристика ключових компонентів інтерфейсу

Назва компонента	Функціональне призначення	Особливості UX-реалізації
Селектор улюбленця	Швидке перемикання між профілями тварин користувача.	Візуалізація через іконки видів тварин для миттєвого розпізнавання.
Картка препарату	Відображення основної інформації про товар.	Наявність чіткого маркера «Потрібен рецепт» для уникнення непорозумінь.
Модальне вікно рецепта	Завантаження та перегляд медичного підтвердження.	Підтримка функції Drag-and-drop та попередній перегляд зображення.
Фільтр специфікацій	Обмеження вибору за медичними параметрами.	Використання слайдерів для ваги та чекбоксів для діючих речовин.

Фінальна стадія проєктування передбачає створення високоточних макетів. Вибір кольорової палітри базується на принципах так званого медичного спокою, тобто у використанні відтінків синього та зеленого, які асоціюються з надійністю та здоров'ям. Обов'язковою вимогою є реалізація адаптивного дизайну, оскільки значна частина користувачів взаємодіє із сервісом через мобільні пристрої, перебуваючи безпосередньо у ветеринарній

клініці. Це вимагає оптимізації розмірів зон натискання та пріоритезації контенту на малих екранах.

Завершується етап проектування створенням інтерактивного прототипу, який дозволяє імітувати реальну роботу системи ще до написання коду. Це дає можливість провести тестування на зручність використання та внести корективи в архітектуру екранів, що значно знижує витрати на подальшу програмну реалізацію.



Рисунок 2.4 – Макет головної сторінки

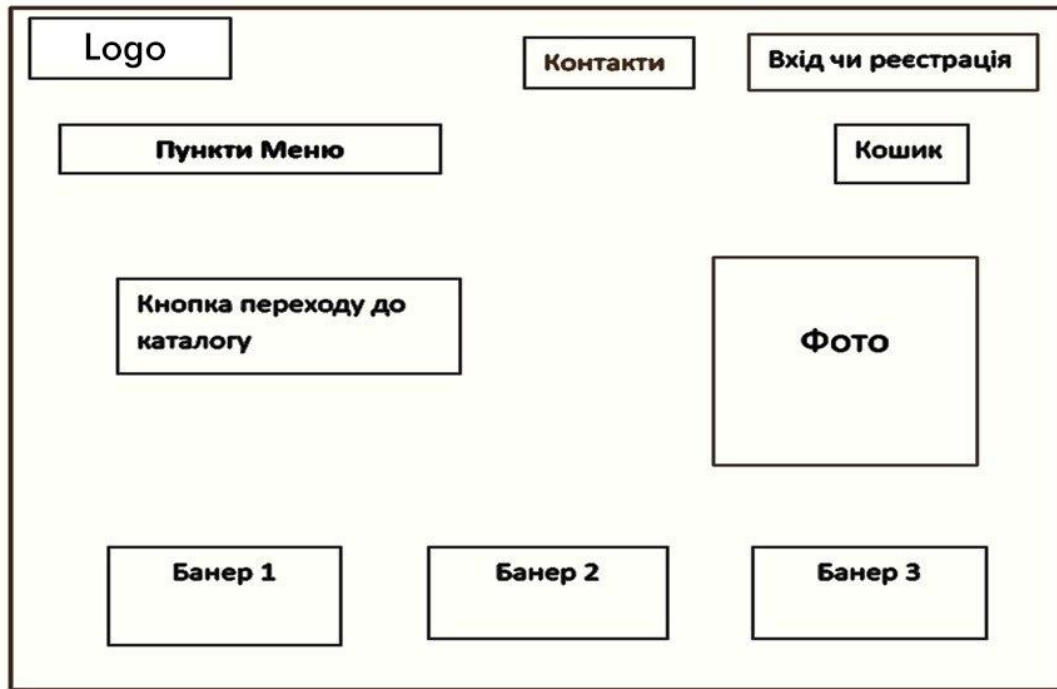


Рисунок 2.5 – Макет вікна сторінки каталогу товарів



Рисунок 2.6 – Макет діалогового вікна для здійснення реєстрації

Зм.	Арк	№ докум.	Підпис	Дата

Logo **Контакти** Вхід чи реєстрація

Кошик

Вхід до акаунту

e-mail

Введіть пароль

Запам'ятати мене

Забули пароль

Увійти

Рисунок 2.7 – Макет вікна авторизації

Кошик

Товар 1

Товар 2

Всього Сума

Переглянути кошик **Оформити замовлення**

Рисунок 2.7 – Макет вікна «Кошик»

2.4 Аналіз та вибір технологій і методів реалізації вебсервісу

Аналіз технологічного стеку для ветеринарного вебсервісу базується на необхідності поєднання високої швидкості розробки з максимальною надійністю обробки медичних та фінансових даних. Вибір інструментів зумовлений потребою в модульній архітектурі, яка дозволяє ізолювати логіку перевірки рецептів від загального функціоналу електронної комерції.

Серверна частина реалізується на базі NestJS, що є прогресивним фреймворком для Node.js. На відміну від мінімалістичних рішень, NestJS примусово впроваджує архітектурні стандарти, такі як ін'єкція залежностей та модульність. Це критично для системи, де бізнес-логіка, наприклад, фільтрація препаратів за вагою тварини має бути легкотестованою та відокремленою від інфраструктурного коду. Використання TypeScript на обох рівнях розробки, тобто бекленд та фронтенд забезпечує строгу типізацію, що мінімізує помилки на етапі компіляції, особливо при роботі зі складними об'єктами замовлень.

Для збереження даних обрано PostgreSQL - об'єктно-реляційну систему, що гарантує відповідність принципам ACID. У контексті ветеринарії це забезпечує цілісність зв'язків між рецептами та транзакціями. Взаємодія з базою здійснюється через Prisma ORM, яка автоматично генерує клієнтські типи на основі схеми БД, усуваючи невідповідність між моделями даних у кодї та реальними таблицями.

У таблиці 2.9 наведено детальний аналіз компонентів системи та їхнього впливу на кінцевий продукт.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		43

Таблиця 2.9 - Порівняльна характеристика технологічного стеку

Категорія	Технологія	Технічне обґрунтування та переваги
Runtime / Мова	Node.js (TypeScript)	Висока продуктивність завдяки неблокуючому вводу-виводу та уніфікація коду на всіх рівнях системи.
Backend Framework	NestJS	Використання архітектурних патернів (Module, Controller, Service) для масштабованості та простоти супроводу.
Frontend Library	React.js	Ефективне оновлення інтерфейсу через Virtual DOM, що важливо для динамічних фільтрів каталогу.
Database	PostgreSQL	Надійність збереження реляційних даних та підтримка складних запитів для медичної аналітики.
Data Access Layer	Prisma ORM	Автоматична міграція схем та безпека запитів (захист від SQL-ін'єкцій на рівні абстракції).
State Management	Zustand	Легковажне управління станом кошика та обраного профілю тварини без зайвого ускладнення коду.
API Architecture	REST / Swagger	Стандартизація взаємодії між клієнтом і сервером із автоматичною документацією ендпоінтів.

Для забезпечення ідентичності середовищ розробки та продуктивної експлуатації використовується Docker. Контейнеризація дозволяє упакувати застосунок, базу даних та кеш-сервер у єдину інфраструктурну одиницю, що виключає проблему роботи на окремому комп'ютері. Це особливо важливо для

дипломного проекту, оскільки гарантує стабільну демонстрацію роботи під час захисту.

Процес верифікації коду базується на використанні Jest. Цей інструмент дозволяє реалізувати автоматизоване модульне тестування, яке перевіряє коректність математичних розрахунків вартості та логіку розмежування прав доступу. Такий підхід забезпечує впевненість у тому, що нові зміни в коді, наприклад, додавання нової категорії ліків не порушать роботу вже існуючих модулів оплати чи реєстрації.

Висновки до 2-го розділу

Отже, 2-й розділ містить опис проектування архітектури вебсервісу, бази даних із визначенням таблиць та зв'язків між ними, аналіз та вибір засобів та технологій реалізації.

Обґрунтування вибору технологічного інструментарію для ветеринарного сервісу базується на пошуку балансу між оперативністю розробки та гарантією безпеки медичних і фінансових транзакцій. Ключовим критерієм вибору архітектури стала можливість декомпозиції системи, що дозволяє відокремити вузькоспеціалізовану логіку перевірки медичних приписів від стандартних процесів онлайн-торгівлі.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
						45
Зм.	Арк	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Реалізація бази даних

Розробка серверної частини та реалізація бізнес-логіки. Для програмної реалізації бекенд-частини вебсервісу обрано середовище виконання Node.js у поєднанні з прогресивним фреймворком NestJS. Використання NestJS дозволяє застосувати принципи об'єктно-орієнтованого програмування та строгу типізацію мови TypeScript. Завдяки вбудованій підтримці патерну впровадження залежностей, код системи стає модульним, що значно спрощує подальше модульне тестування. Для безпечної взаємодії з реляційною базою даних PostgreSQL інтегровано сучасний ORM-інструмент (Object-Relational Mapping), наприклад Prisma, який автоматично генерує типи на основі схеми бази даних та захищає систему від SQL-ін'єкцій.

Центральним компонентом бізнес-логіки системи є модуль обробки замовлень OrderService. Алгоритм оформлення покупки у ветеринарному сервісі суттєво відрізняється від класичного e-commerce. При надходженні запиту на створення замовлення система повинна не лише підрахувати загальну вартість, а й проаналізувати медичні атрибути кожного товару в кошику. Якщо хоча б один препарат має прапорець `is_prescription_required`, загальний статус замовлення має бути примусово встановлений в очікування верифікації документа, блокуючи можливість миттєвої оплати.

Нижче наведено фрагмент програмного коду, що демонструє реалізацію цієї логіки на рівні сервісу.

```
import { Injectable, BadRequestException } from '@nestjs/common';
import { PrismaService } from '../prisma/prisma.service';

@Injectable()
export class OrderService {
```

					КвРІПЗ.200169.01.19.ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    constructor(private prisma: PrismaService) {}

    async createOrder(userId: number, petId: number, items: { productId:
number; quantity: number }[]) {
        // Відкриваємо транзакцію для гарантії цілісності даних
        return this.prisma.$transaction(async (tx) => {
            let totalAmount = 0;
            let requiresPrescription = false;
            const orderItemsData = [];

            // Перевірка кожного товару з кошика
            for (const item of items) {
                const product = await tx.product.findUnique({ where: { id:
item.productId } });

                if (!product) {
                    throw new BadRequestException(`Товар з ID ${item.productId} не
знайдено`);
                }
                if (product.stock_quantity < item.quantity) {
                    throw new BadRequestException(`Недостатньо товару
"${product.name}" на складі`);
                }
                // Фіксуємо історичну ціну та перевіряємо статус рецептурності
                totalAmount += product.price * item.quantity;
                if (product.is_prescription_required) {
                    requiresPrescription = true;
                }

                orderItemsData.push({

```

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		47

```

    product_id: product.id,
    quantity: item.quantity,
    price_at_purchase: product.price,
  });

  // Резервуємо товар на складі (зменшуємо залишок)
  await tx.product.update({
    where: { id: product.id },
    data: { stock_quantity: product.stock_quantity - item.quantity },
  });
}

// Визначення початкового статусу замовлення
const initialStatus = requiresPrescription ? 'PENDING_PRESCRIPTION'
: 'NEW';

// Збереження замовлення в базу даних
const order = await tx.order.create({
  data: {
    user_id: userId,
    pet_id: petId,
    total_amount: totalAmount,
    status: initialStatus,
    order_items: {
      create: orderItemsData,
    },
  },
});

return order;

```

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

```
});  
}  
}
```

У наведеному лістингу критично важливою є обгортка *transaction*. Вона гарантує властивість атомарності *Atomicity* згідно з вимогами ACID. Якщо під час ітерації по товарах з'ясується, що одного з препаратів немає на складі, система згенерує виключення *BadRequestException*, і жоден запис (ні саме замовлення, ні списання залишків) не буде збережено в базу. Також код демонструє роботу з історичною ціною *price_at_purchase*, що була закладена під час проєктування структури бази даних.

Важливим також є SQL-скрипт (DDL) для створення структури бази даних у PostgreSQL на основі розробленої ER-діаграми та таблиць. Доцільно додати обмеження для забезпечення цілісності даних.

```
CREATE TABLE categories (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(100) UNIQUE NOT NULL,  
    description TEXT  
);
```

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    email VARCHAR(255) UNIQUE NOT NULL,  
    password_hash VARCHAR(255) NOT NULL,  
    role VARCHAR(20) DEFAULT 'USER' CHECK (role IN ('ADMIN',  
'USER')),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE pets (  
    id SERIAL PRIMARY KEY,  
    user_id INT REFERENCES users(id) ON DELETE CASCADE,
```

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		49

```

name VARCHAR(100) NOT NULL,
species VARCHAR(50) NOT NULL,
weight DECIMAL(5,2) NOT NULL CHECK (weight > 0),
allergies TEXT
);
CREATE TABLE products (
id SERIAL PRIMARY KEY,
category_id INT REFERENCES categories(id) ON DELETE SET NULL,
name VARCHAR(255) NOT NULL,
active_substance VARCHAR(255) NOT NULL,
price DECIMAL(10,2) NOT NULL CHECK (price >= 0),
stock_quantity INT DEFAULT 0 CHECK (stock_quantity >= 0), --
Кількість на складі
is_prescription_required BOOLEAN DEFAULT FALSE
);
CREATE TABLE orders (
id SERIAL PRIMARY KEY,
user_id INT REFERENCES users(id) ON DELETE RESTRICT,
pet_id INT REFERENCES pets(id) ON DELETE SET NULL,
status VARCHAR(50) DEFAULT 'NEW' CHECK (status IN ('NEW', 'PAID',
'SHIPPED', 'CANCELLED', 'PENDING_PRESCRIPTION')),
total_amount DECIMAL(10,2) NOT NULL DEFAULT 0 CHECK
(total_amount >= 0),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE order_items (
id SERIAL PRIMARY KEY,
order_id INT REFERENCES orders(id) ON DELETE CASCADE,
product_id INT REFERENCES products(id) ON DELETE RESTRICT,

```

					КвРПІІЗ.200169.01.19.ПІЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		50

```

quantity INT NOT NULL CHECK (quantity > 0),
price_at_purchase DECIMAL(10,2) NOT NULL
);
CREATE TABLE prescriptions (
id SERIAL PRIMARY KEY,
order_id INT UNIQUE REFERENCES orders(id) ON DELETE CASCADE,
-- Кожне замовлення має макс. 1 рецепт
file_path VARCHAR(512) NOT NULL,
is_approved BOOLEAN DEFAULT FALSE
);

```

У даному лістингу коду присутнє також обмеження у кодї, а саме: SERIAL, що автоматично генерує унікальний ID для кожного нового запису;

PRIMARY К унікально ідентифікує кожен рядок у таблиці;

REFERENCES здійснює встановлення зв'язку між таблицями, наприклад, pet_id в orders посилається на id в pets;

UNIQUE гарантує, що значення в полі не повторюються, наприклад, email користувача;

CHECK дозволяє задати умову для даних, що вносяться, наприклад, ціна не може бути від'ємною, або роль користувача має бути лише ADMIN або USER;

ON DELETE CASCADE/RESTRICT/SET NULL визначає, що робити з пов'язаними записами при видаленні батьківського запису. Це ключовий елемент для підтримки цілісності бази даних.

3.2 Реалізація модулів системи

Для розробки клієнтської частини вебсервісу було обрано бібліотеку React, що дозволяє будувати інтерфейси на основі компонентного підходу.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		51

Використання концепції Single Page Application забезпечує миттєву реакцію системи на дії користувача без необхідності повного перезавантаження сторінки. Для стилізації компонентів інтегровано фреймворк Tailwind CSS, який базується на utility-класах і дозволяє створювати адаптивний дизайн, що коректно відображається як на десктопних, так і на мобільних пристроях. Управління глобальним станом застосунку (наприклад, даними про авторизованого користувача та обраного улюбленця) реалізовано за допомогою сучасного стейт-менеджера Zustand, який відрізняється мінімальним розміром та відсутністю зайвого шаблонного коду.

Головним викликом при проектуванні користувацького інтерфейсу ветеринарної аптеки є реалізація динамічної фільтрації каталогу. Користувач не повинен самостійно вчитувати інструкції до кожного препарату, щоб дізнатися, чи підходить він для його тварини. Система має автоматично приховувати нерелевантні товари на основі обраного профілю, наприклад, приховувати ліки для собак, якщо користувач купує для kota, або фільтрувати препарати за допустимою вагою).

Нижче наведено лістинг головного компонента каталогу, який демонструє реалізацію алгоритму розумної фільтрації на боці клієнта з використанням хуків React.

```
import React, { useState, useMemo, useEffect } from 'react';
import { useStore } from '../store/useStore';
import ProductCard from './ProductCard';

const MedicationCatalog = () => {
  // Отримання активного профілю тварини з глобального стану
  const { activePet } = useStore();
  const [products, setProducts] = useState([]);
  const [isLoading, setIsLoading] = useState(true);
  // Імітація запиту до API для отримання повного каталогу
```

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

```

useEffect(() => {
  const fetchCatalog = async () => {
    try {
      const response = await fetch('/api/v1/products');
      const data = await response.json();
      setProducts(data);
    } catch (error) {
      console.error("Помилка завантаження каталогу", error);
    } finally {
      setIsLoading(false);
    }
  };
  fetchCatalog();
}, []);

```

// Оптимізація фільтрації: перерахунок лише при зміні товарів або профілю тварини

```

const filteredProducts = useMemo(() => {
  if (!activePet) return products; // Якщо тварину не обрано, показуємо все
  return products.filter(product => {
    // 1. Перевірка на відповідність виду тварини (species)
    const isSpeciesMatch = product.target_species.includes(activePet.species)
    || product.target_species === 'ALL';

    // 2. Перевірка на відповідність ваги (якщо препарат має такі обмеження)
    const isWeightMatch = (!product.min_weight || activePet.weight >=
product.min_weight) &&
      (!product.max_weight || activePet.weight <=
product.max_weight);

    return isSpeciesMatch && isWeightMatch;
  });

```

```

    });
  }, [products, activePet]);

  if (isLoading) return <div>Завантаження каталогу...</div>;

  return (
    <div className="catalog-container p-4">
      <h2 className="text-2xl font-bold mb-4">
        {activePet ? `Рекомендовані препарати для: ${activePet.name}` :
        'Загальний каталог'}
      </h2>
      <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
        {filteredProducts.length > 0 ? (
          filteredProducts.map(product => (
            <ProductCard key={product.id} data={product} />
          ))
        ) : (
          <p className="text-gray-500">Для вашого улюбленця не знайдено
          відповідних препаратів.</p>
        )}
      </div>
    </div>
  );
};

export default MedicationCatalog;

```

Важливим архітектурним рішенням у наведеному кодi є використання хука useMemo. Оскільки масив товарів може містити сотні позицій, застосування useMemo гарантує, що ресурсоемна операція фільтрації масиву не буде виконуватися при кожному перемальовуванні компонента, наприклад,

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		54

якщо користувач просто відкрив бокове меню, а запуститься лише тоді, коли зміниться базовий масив даних від сервера або користувач перемкнеться на профіль іншої тварини. Це прямо відповідає нефункціональним вимогам щодо швидкодії системи, описаним у першому розділі роботи.

3.3 Інструкція користувача

Для зручного користування розробленим застосунком доцільно описати його використання для потенційного користувача в залежності від виконуваної ролі.

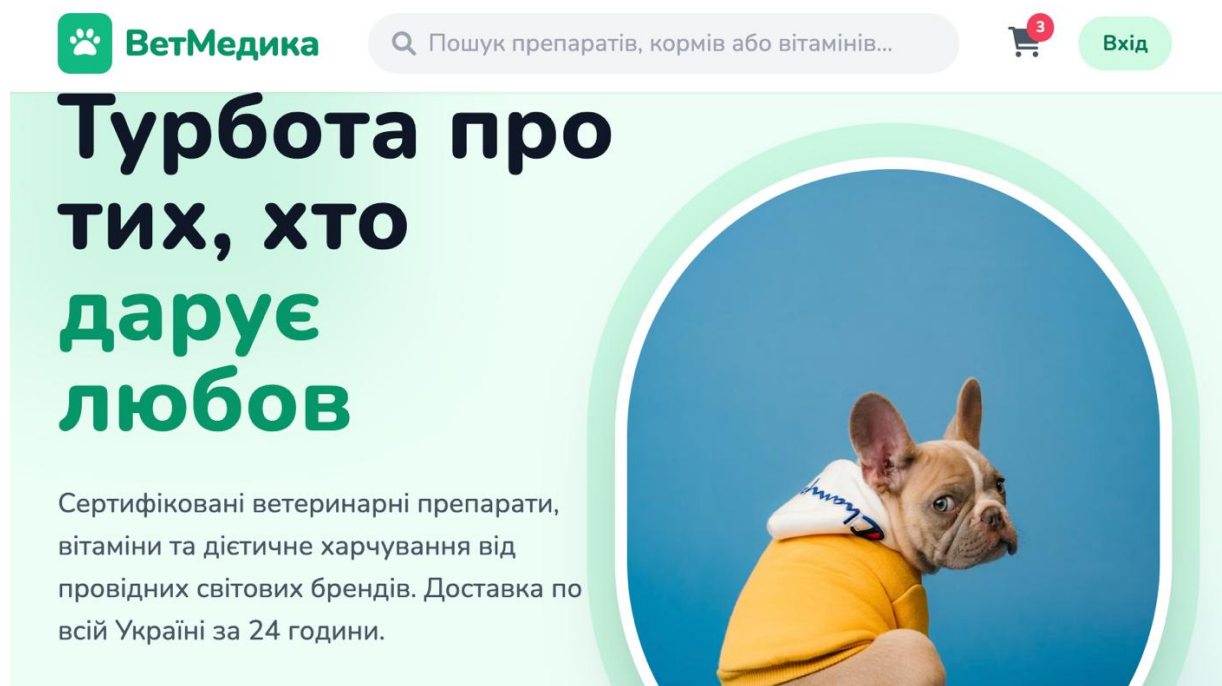


Рисунок 3.1 – Головне вікно

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		55

Для здійснення покупки використовується такий інструмент як вибір за видом тварини (рисунок 3.2). із переміщенням у кошику, що дає можливість обирати замовнику потрібний препарат, а також керувати його кількістю, асортиментом, можливістю оплати, способом доставки. Для того, щоб здійснювати якісь дії з товаром клієнт має можливість переглянути його ще раз у кошику, натиснувши на відповідну іконку кошика.

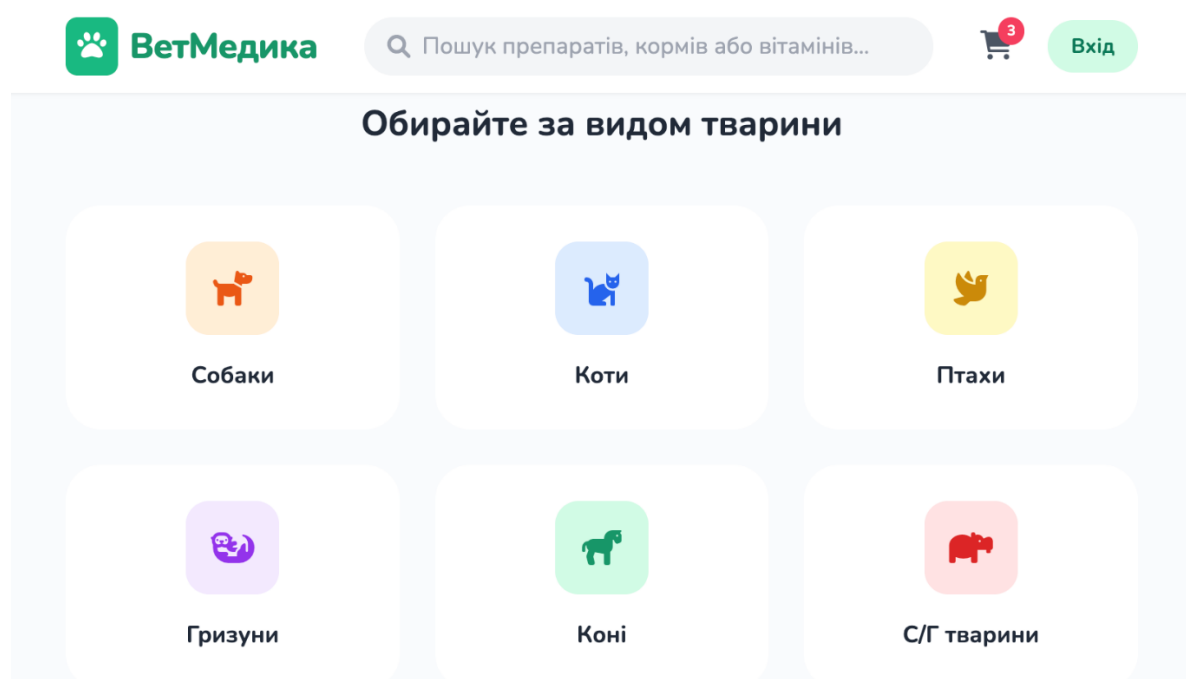


Рисунок 3.2 – Вибір за видом тварини

Щоб користувач міг робити замовлення він обов'язково повинен здійснити реєстрацію. Для цього йому потрібно натиснути на відповідну кнопку реєстрації, після чого з'явиться діалогове вікно, що зображено на рисунку 3.3. У цьому вікні треба ввести ім'я, прізвище, поштову адресу, пароль із повторним підтвердженням, далі надати погодження із правилами та політикою безпеки та обрати кнопку Зареєструватися.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		56

Реєстрація

Ім'я

Прізвище

Email

Пароль

Пароль повинен містити не менше 8 символів, включаючи цифри та спеціальні символи

Підтвердження пароля

Я погоджуюсь з умовами використання та політикою конфіденційності

Зареєструватися

[Вже є акаунт? Увійти](#)

Рисунок 3.3 – Вікно реєстрації

Після того, як користувач здійснив реєстрацію, він може здійснювати перегляд препаратів відповідно до своїх потреб, вибір та купівлю із подальшою оплатою та доставкою (рисунок 3.4).

Вхід до акаунту

Email

Пароль

Запам'ятати мене [Забули пароль?](#)

Увійти

[Немає акаунту? Зареєструватися](#)

Рисунок 3.4 – Авторизація

Також є можливість обрати й інші послуги, зокрема швидку доставку, цілодобову підтримку лікаря із допомогою у підборі препарату, обрати сертифікат для того чи іншого препарату (рисунок 3.5).

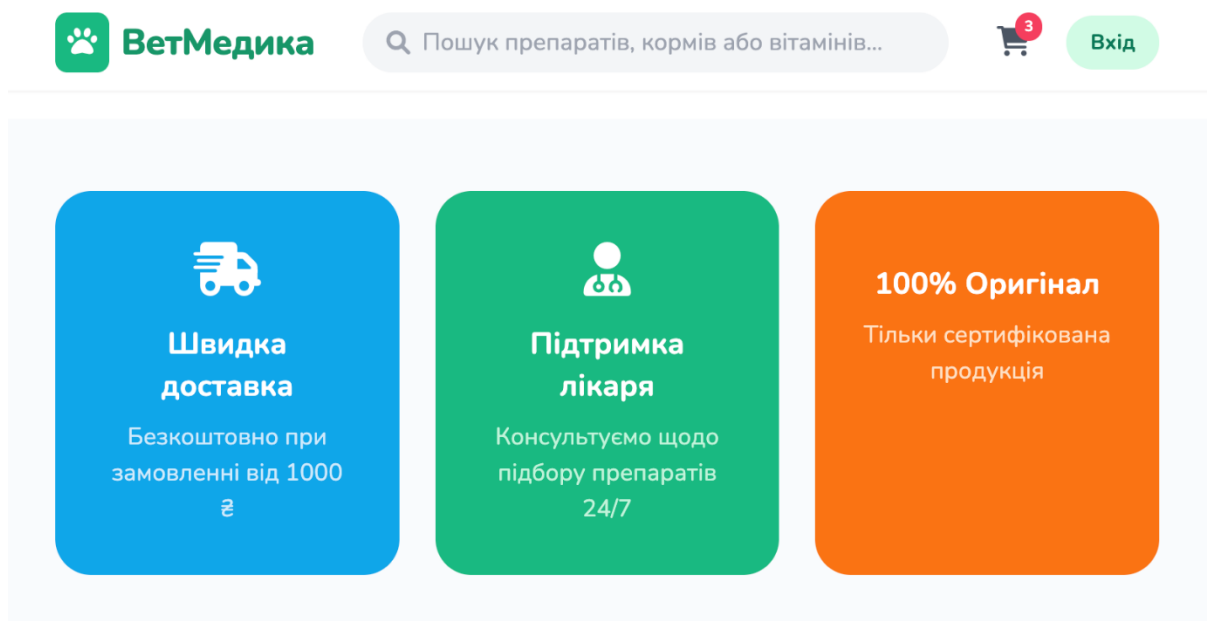


Рисунок 3.5 – Вікно вибору необхідної допомоги

Отже, як видно користування подібного роду застосунками є дуже простим та досить лаконічним та не потребує якогось додаткового навчання.

3.4 Тестування вебзастосунку

Метою тестування розробленого вебсервісу є перевірка відповідності функціоналу встановленим вимогам та виявлення дефектів на ранніх етапах. У межах даного проєкту було реалізовано стратегію піраміди тестування, яка включає модульне, інтеграційне та функціональне тестування. Основний акцент зроблено на модульному тестуванні критичних вузлів системи - алгоритмів розрахунку вартості та валідації рецептів.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		58

Для написання тестів обрано фреймворк Jest, який є стандартом для екосистеми JavaScript/TypeScript. Тестування проводилося за методологією ізоляції: залежності, наприклад, доступ до бази даних замінювалися на макети, що дозволило перевірити чисту логіку сервісів без навантаження на інфраструктуру.

Нижче наведено приклад модульного тесту для OrderService, який перевіряє сценарій, коли користувач намагається замовити товар у кількості, що перевищує наявний запас на складі.

```
import { OrderService } from './order.service';
import { BadRequestException } from '@nestjs/common';

describe('OrderService', () => {
  let orderService;
  let mockPrisma;

  beforeEach(() => {
    // Створення моку для бази даних
    mockPrisma = {
      product: {
        findUnique: jest.fn(),
      },
      $transaction: jest.fn((callback) => callback(mockPrisma)),
    };
    orderService = new OrderService(mockPrisma);
  });

  it('має видавати помилку, якщо товару недостатньо на складі', async ()
=> {
    // Дані для тесту: на складі 5 одиниць, користувач хоче 10
```

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		59

```

const mockProduct = { id: 1, name: 'Антибіотик', stock_quantity: 5, price:
100 };
mockPrisma.product.findUnique.mockResolvedValue(mockProduct);

const orderData = [{ productId: 1, quantity: 10 }];

// Перевірка, чи сервіс викидає правильне виключення
await expect(orderService.createOrder(1, 1, orderData))
  .rejects
  .toThrow(BadRequestException);

await expect(orderService.createOrder(1, 1, orderData))
  .rejects
  .toThrow ('Недостатньо товару Антибіотик на складі');
});
});

```

Для підтвердження якості виконання роботи наводиться таблиця відповідності реалізованих функцій вимогам, що були визначені в першому розділі (таблиця 3.1).

Таблиця 3.1 – Відповідність реалізованих функцій згідно визначених вимог

Об'єкт тестування	Очікуваний результат	Статус (Pass/Fail)
Реєстрація користувача	Створення запису в таблиці users, хешування пароля	Pass

Фільтрація ліків	Відображення лише тих препаратів, що підходять виду тварини	Pass
Створення замовлення	Зміна статусу на PENDING_PRESCRIPTION при виборі ліків з рецептом	Pass
Валідація залишків	Блокування оформлення при quantity > stock_quantity	Pass
Безпека API	Помилка 401 Unauthorized при запиті без JWT-токена	Pass

Отже, як видно розроблюваний програмний застосунок був перевірений на помилки під час реєстрації, відповідність фільтрації ліків, валідацію залишків, а також безпеку API. Помилки виявлено не було.

3.5 Висновки до третього розділу

Отже, третій розділ містить опис реалізації вебсервісу для торгівлі ветеринарними препаратами, що включає модуль для обробки замовлень. Показано як здійснюється безпосередня реалізація спроектованої раніше бази даних, а також інтерфейсу користувача. Разом з цим подано й інструкцію для комфортного користування розробленим програмним продуктом. В рамках життєвого циклу розроблено тестові сценарії та протестовано дане програмне забезпечення. Відповідно до результатів тестування вказано висновки щодо відповідності вимог, які вказувались на початку розробки.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

ВИСНОВКИ

Дослідження показало, що специфіка ветеринарного ритейлу вимагає особливого підходу до обробки даних, що включає контроль за рецептурними препаратами та ведення профілів тварин. Виявлені недоліки в існуючих бізнес-процесах, а саме ручна обробка, відсутність автоматичного контролю термінів придатності підтвердили актуальність створення спеціалізованого програмного рішення.

У ході виконання кваліфікаційної роботи було розроблено вебсервіс для торгівлі ветеринарними препаратами, який автоматизує процес підбору та купівлі медикаментів із врахуванням специфічних потреб домашніх тварин. За результатами проведеного дослідження та розробки можна зробити наступні висновки.

Проаналізовано предметну область та виявлено потребу у спеціалізованому рішенні, яке інтегрує медичні профілі тварин у процес електронної комерції, що дозволило сформулювати вимоги до системи, зокрема щодо необхідності верифікації рецептів та динамічної фільтрації товарів.

Спроектовано архітектуру системи на основі багат шарового підходу та сервіс-орієнтованої моделі. Використання діаграм варіантів використання, станів та моделі контейнерів дозволило створити стійку до навантажень та легку в масштабуванні структуру застосунку.

Розроблено реляційну модель бази даних у третій нормальній формі. Структура БД забезпечує цілісність даних через систему обмежень (constraints) та дозволяє зберігати історію замовлень із фіксацією цін на момент покупки, що є критичним для фінансової звітності.

Реалізовано програмний продукт із використанням сучасного стеку технологій: Node.js/NestJS для серверної частини та React для клієнтської. Впроваджено алгоритм автоматичної перевірки рецептурності препаратів, що підвищує безпеку використання ветеринарних засобів.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		62

Проведено тестування основних функціональних модулів за допомогою фреймворку Jest. Результати тестування підтвердили повну відповідність розробленого сервісу технічному завданню, надійність обробки транзакцій та стійкість системи до некоректних вхідних даних.

У процесі розробки було успішно застосовано стек технологій PERN (PostgreSQL, Express.js, React, Node.js). Це дозволило створити відмовостійку систему з високою швидкістю відгуку. Проєктування реляційної бази даних забезпечило цілісність інформаційного забезпечення, а використання REST API - гнучку інтеграцію з платіжними системами та логістичними сервісами.

Спроектований вебмодуль успішно реалізує:

- автоматизований алгоритм обробки замовлень, що враховує специфіку ветеринарних засобів;
- систему валідації цифрових рецептів, що мінімізує юридичні ризики підприємства;
- модуль логістики та оплати, який забезпечує автоматичне формування ТТН та фіскалізацію чеків.

Мета кваліфікаційної роботи досягнута в повному обсязі. Отримані результати, архітектурні рішення та розроблений програмний код використані як основна частина кваліфікаційної роботи бакалавра.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про електронну комерцію : Закон України від 03.09.2015 р. № 675-VIII. URL: <https://zakon.rada.gov.ua/laws/show/675-19> (дата звернення: 10.03.2026).
2. Про ветеринарну медицину : Закон України від 16.11.1992 р. № 2498-XII. URL: <https://zakon.rada.gov.ua/laws/show/2498-12> (дата звернення: 10.03.2026).
3. Про захист персональних даних : Закон України від 01.06.2010 р. № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 10.03.2026).
4. ДСТУ ISO/IEC/IEEE 12207:2023 (ISO/IEC/IEEE 12207:2017, IDT). Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення. Київ : ДП «УкрНДНЦ», 2023. 112 с.
5. ДСТУ ISO/IEC 25010:2016. Системна та програмна інженерія. Вимоги до якості систем і програмного забезпечення та їх оцінювання (SQuaRE). Київ : ДП «УкрНДНЦ», 2016. 45 с.
6. Брукс Ф. Міфічний людино-місяць. Нариси з системного програмування. Київ : Вид. група BHV, 2021. 304 с.
7. Мартін Р. Чиста архітектура. Мистецтво розроблення програмного забезпечення. Харків : Фабула, 2019. 368 с.
8. Мартін Р. Чистий код. Харків : Фабула, 2019. 448 с.
9. Фаулер М. Рефакторинг. Поліпшення структури існуючого коду. Київ : Діалектика, 2019. 448 с.
10. Гамма Е., Гелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проектування. Патерни проектування. Санкт-Петербург : Пітер, 2021. 368 с.
11. Банг Е. Гнучка розробка програмного забезпечення: Скрам, Канбан, Lean. Київ : Наш Формат, 2018. 232 с.
12. Соммервілл І. Інженерія програмного забезпечення. 10-те вид. Київ : Діалектика, 2018. 848 с.
13. Клеппман М. Проектування високонавантажених застосунків. Київ : Діалектика, 2020. 688 с.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		64

14. Конон Д., Баклі Дж. Node.js. Путівник для початківців. Київ : Діалектика, 2021. 416 с.
15. Чиннатхамбі К. React: бібліотека JavaScript для створення користувацьких інтерфейсів. Київ : Діалектика, 2019. 352 с.
16. Троелсен Е., Дженікс Ф. Мова програмування C# 8 і платформи .NET Core. Київ : Діалектика, 2020. 1100 с. (використовується для порівняння платформ).
17. Н'юмен С. Створення мікросервісів. Санкт-Петербург : Пітер, 2021. 624 с.
18. Хемблтон Д. Гнучке тестування: практичний посібник для тестувальників та гнучких команд. Київ : Діалектика, 2020. 520 с.
19. Форд Н. Архітектура програмного забезпечення: фундаментальні принципи. Київ : Діалектика, 2021. 432 с.
20. Шваб К. Четверта промислова революція. Київ : Форс Україна, 2019. 232 с.
21. Блос К. PostgreSQL 12. Основи. Санкт-Петербург : Пітер, 2020. 448 с.
22. Фрімен Е., Робсон Е. Head First JavaScript Programming. Sebastopol : O'Reilly Media, 2014. 704 p.
23. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. Sebastopol : O'Reilly Media, 2020. 360 p.
24. Choi K. NestJS Progressively. Packt Publishing, 2022. 412 p.
25. Katz S. S. Microservices Architecture: Aligning Design Principles with Business Goals. New York : Apress, 2021. 380 p.
26. NestJS Documentation. URL: <https://docs.nestjs.com/> (дата звернення: 10.03.2026).
27. React Documentation. URL: <https://react.dev/> (дата звернення: 10.03.2026).
28. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 10.03.2026).
29. Prisma ORM Documentation. URL: <https://www.prisma.io/docs/> (дата звернення: 10.03.2026).
30. Docker Documentation. URL: <https://docs.docker.com/> (дата звернення: 10.03.2026).

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		65

- 31.Redux Toolkit Guide. URL: <https://redux-toolkit.js.org/> (дата звернення: 10.03.2026).
- 32.Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 10.03.2026).
- 33.Jest Testing Framework. URL: <https://jestjs.io/> (дата звернення: 10.03.2026).
- 34.TypeScript Handbook. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 10.03.2026).
- 35.OWASP Top Ten Project. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 10.03.2026).
- 36.Гільберт Г. Патерни проектування для TypeScript. Київ : Діалектика, 2021. 288 с.
- 37.Бредлі К. Розробка REST API на Node.js. Санкт-Петербург : Пітер, 2022. 320 с.
- 38.Фаулер М. Шаблони корпоративних програмних додатків. Київ : Вільямс, 2018. 544 с.
- 39.Дюваль П. Безперервна інтеграція: Поліпшення якості програмного забезпечення та зниження ризику. Київ : Діалектика, 2019. 304 с.
- 40.Кан К. Архітектура чистого коду на React. Нью-Йорк : Apress, 2022. 290 с.

					КвРІПЗ.200169.01.19.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		66

Додаток А (обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

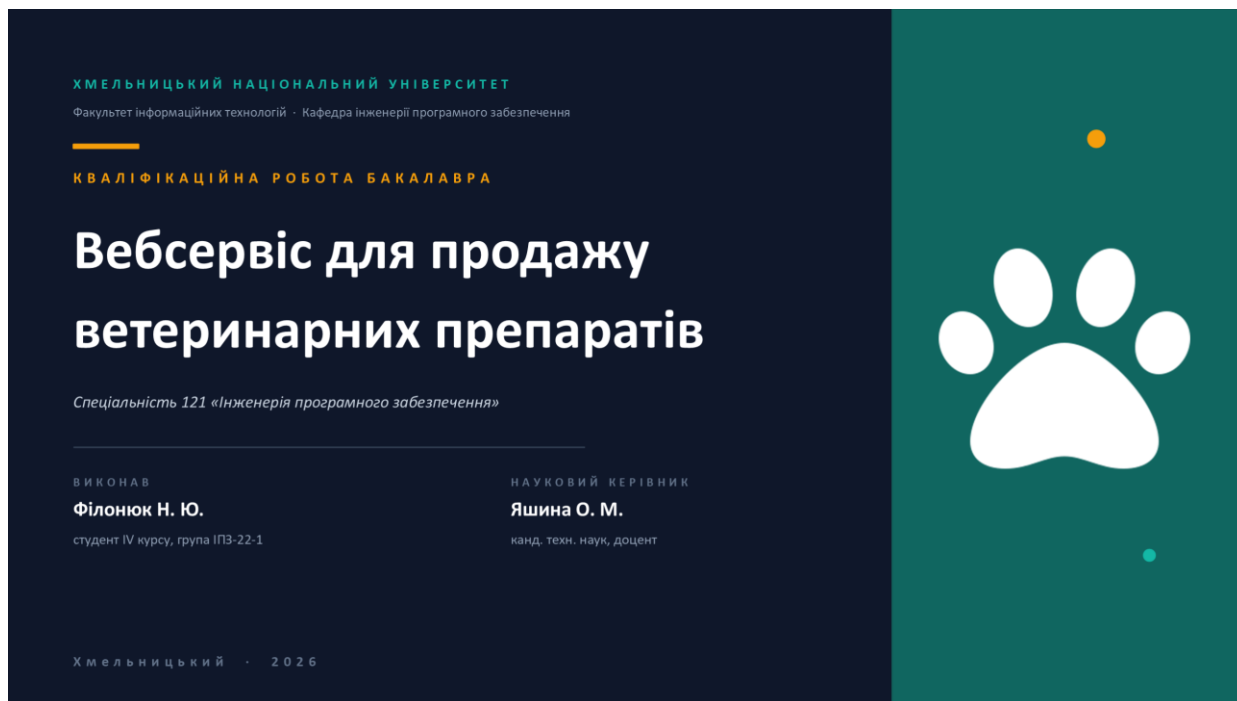


Рисунок А.1 – Титульний слайд

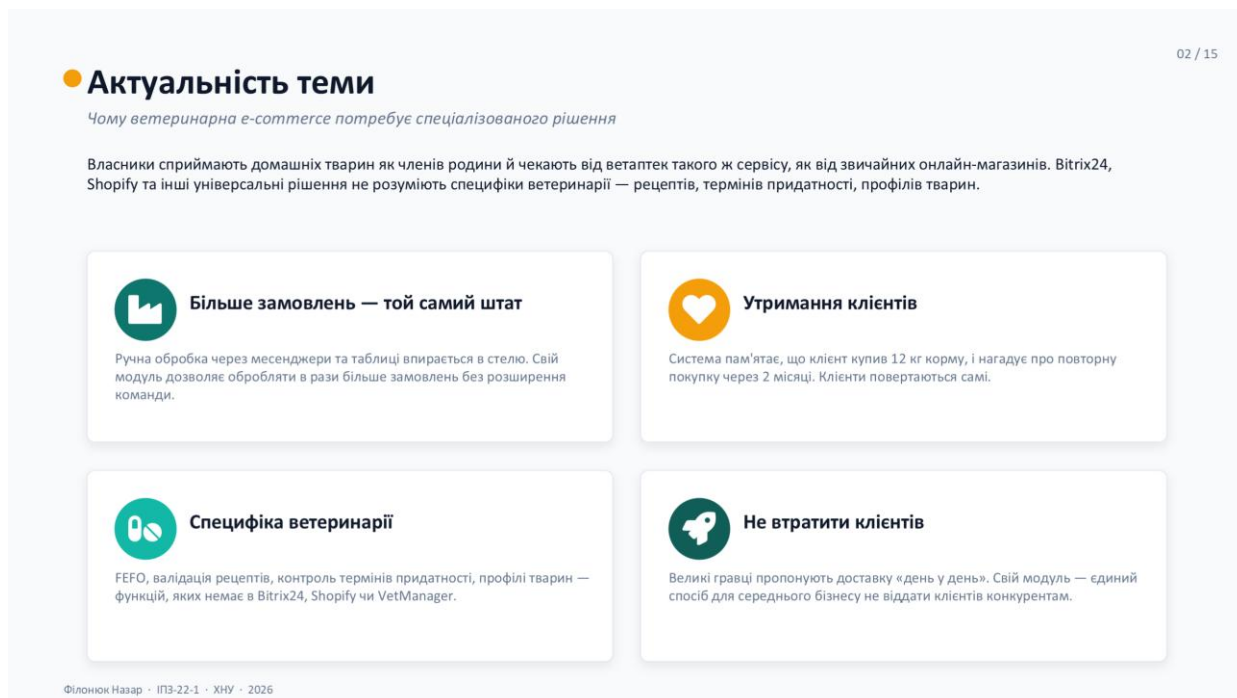


Рисунок А.2 – Актуальність теми

03 / 15

● Мета та завдання роботи

МЕТА РОБОТИ
Проведення проектування і розроблення вебмодуля для обробки замовлень у ветеринарному онлайн-магазині.

ЗАВДАННЯ ДЛЯ ДОСЯГНЕННЯ МЕТИ

<p>1 Проаналізувати область продажу ветеринарних товарів, визначити особливості та межі використання функціоналу</p>	<p>2 Проаналізувати існуючі вебмодулі для реалізації та обробки замовлень на ресурсах із продажу ветеринарних товарів</p>
<p>3 Встановити плюси та мінуси існуючих розробок</p>	<p>4 Здійснити постановку задачі із подальшою розробкою технічного завдання</p>
<p>5 Вибрати тип архітектури для розробки програмного забезпечення</p>	<p>6 Проаналізувати та вибрати інструментарій, засоби та технології розроблення ПЗ</p>
<p>7 Здійснити реліз та тестування розробленого програмного забезпечення</p>	

Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.3 – Мета та завдання роботи

04 / 15

● Дослідження предметної області

Як влаштований ветеринарний e-commerce і що в ньому особливого

<div style="display: flex; align-items: center; margin-bottom: 10px;"> <p style="margin: 0;">СПЕЦИФІКА ДОМЕНУ Ветеринарний e-commerce</p> </div> <p>Покупець ≠ споживач: ліки приймає тварина, а замовляє і платить власник. Система працює з даними обох.</p> <p>Це не звичайний магазин: тут продаються ліки. Потрібен контроль термінів придатності, перевірка рецептів, відповідальність за здоров'я тварини.</p> <p>Три ключові сутності: Клієнт, його Тварина і Препарат. Кожен препарат пов'язаний з активною речовиною та видом тварини, якій підходить.</p> <p>Покупки повторюються: корм, обробка від паразитів, хронічні ліки — клієнт купує по циклу. Система нагадує, коли час замовляти знову.</p> <p>Зв'язок замовлення з твариною: кожне замовлення прив'язане не лише до клієнта, а й до конкретної тварини в його профілі.</p>	<div style="display: flex; align-items: center; margin-bottom: 10px;"> <p style="margin: 0;">ІНФОРМАЦІЙНІ ПОТОКИ Архітектура даних</p> </div> <p>В ХІДН А Дані клієнта, параметри тварини (вид, вага, вік), фото рецептів, поставки від постачальників із термінами</p> <p>В НУТРІШН Я Каталог препаратів, довідники дозувань, статуси замовлень, залишки на складі</p> <p>В ИХІДН А Накладні Нової Пошти, рахунки, листи клієнту, звіти про продажі, акти списання</p>
---	--

Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.4 – Дослідження предметної області

● Аналіз існуючого програмно-технічного забезпечення

05 / 15

Порівняння категорій рішень для ветеринарного e-commerce

Категорія платформи	Приклади	Переваги	Недоліки для ветаптеки
Універсальні CRM (SaaS)	Bitrix24, Zoho	Потужні інструменти продажів, налаштовувані воронки, автоматизація	Не знає про тварин і рецепти. Не розуміє медичну логіку
Спеціалізовані медичні ІС	VetManager, ezyVet	Глибокий медичний облік, картки пацієнтів, історія лікування	Це інструмент для клініки, а не магазин. Немає кошика, доставки, оплати
E-commerce платформи	Shopify, Magento	Найкращий UX для покупок, SEO-оптимізація, маркетинг-інструменти	Дорога кастомізація під ветеринарну логіку (FEFO, рецепти)
Локальні ветаптеки	33korovy.in.ua	Добра категоризація за медичним призначенням, професійні описи	Застарілий стек, повільні, не адаптуються під мобільні
Міжнародні гіганти	Chewy.com, 1800petmeds	Мікросервіси, профілі тварин, автоматична перевірка рецептів	Не працюють в Україні, не відповідають місцевому законодавству

ВИСНОВОК: існуючі рішення або «загальні» (не розуміють специфіки ліків), або «клінічні» (не зручні для торгівлі). На українському ринку немає магазину, що поєднує сучасний стек із розумінням ветеринарії. Цю прогалину закриває моя робота.

Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.5 – Аналіз існуючого програмно-технічного забезпечення

● Функціональні та нефункціональні вимоги

06 / 15

Що система робить та як вона це робить

ФУНКЦІОНАЛЬНІ ВИМОГИ Що робить система	НЕФУНКЦІОНАЛЬНІ ВИМОГИ Як система працює
<p>Профілі тварин Картка з видом, вагою, віком, алергіями — щоб перевіряти, чи підходить ліки тварині</p>	<p>Швидкість Сервер відповідає менше ніж за 200 мс на 95% запитів 200ms</p>
<p>Перевірка рецептів Не дає оплатити рецептурні ліки, поки клієнт не завантажить фото рецепта</p>	<p>Масштабованість Витримує 500 одночасних користувачів і 50 менеджерів 500</p>
<p>Інтеграція з ПРРО (Checkbox) Автоматично формує фіскальний чек при оплаті замовлення</p>	<p>Безпека HTTPS, JWT-токени, шифрування персональних даних JWT</p>
<p>Накладні Нової Пошти Генерує ТТН через API «Нової Пошти» або «Укрпошти» прямо з адмінки</p>	<p>Доступність Працює 99.9% часу — не більше 9 годин простою на рік 99.9%</p>
<p>Контроль FEFO Першим відвантажує товар з найближчим терміном придатності — щоб не списувати протрочене</p>	<p>Бекапи Резервне копіювання, відновлення за 2 години ≤ 2h</p>
<p>Сповіщення клієнтам Надсилає статус замовлення через SMS, Email, Viber або Telegram</p>	<p>Адаптивність Працює на десктопі, планшеті, смартфоні Mobile</p>

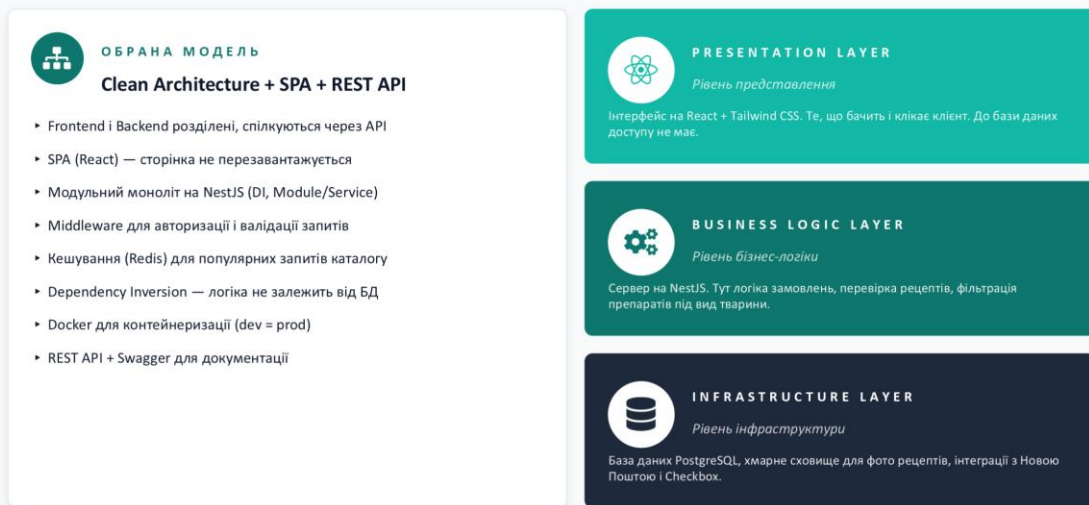
Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.6 – Функціональні та нефункціональні вимоги

Архітектура та шаблони проектування

07 / 15

Багатошарова сервіс-орієнтована архітектура за принципами Clean Architecture



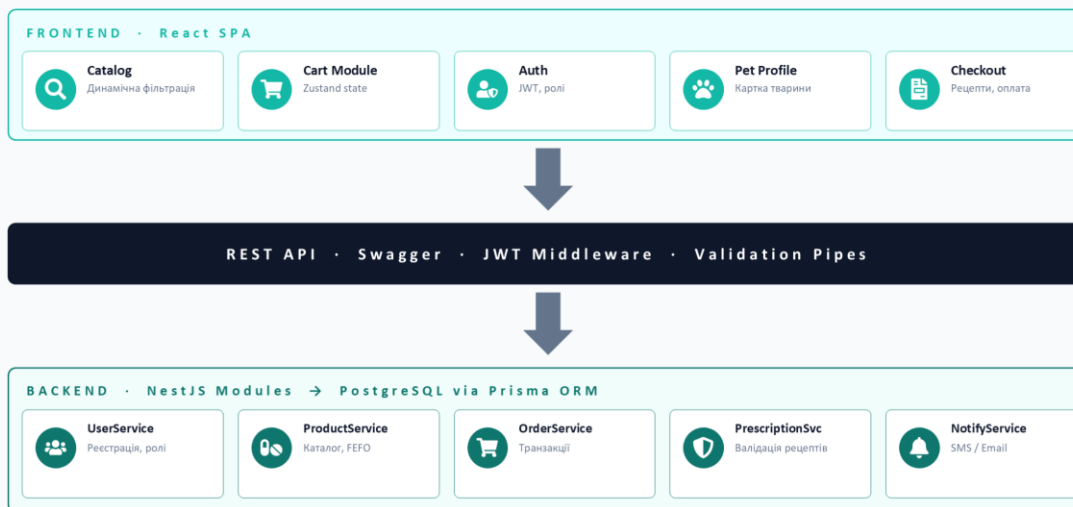
Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.7 – Архітектура та шаблони проектування

Декомпозиція, залежності та інтерфейси

08 / 15

Структурні модулі системи та їхня взаємодія



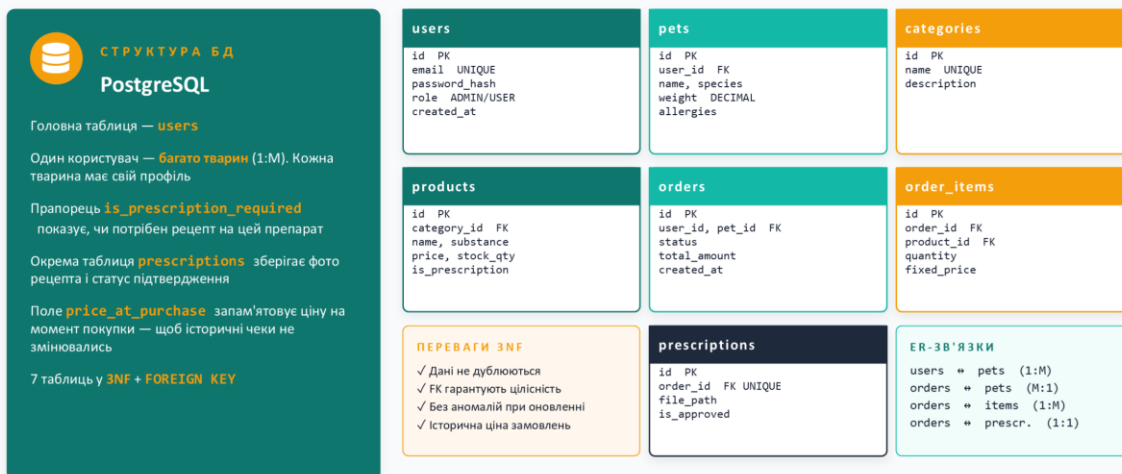
Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.8 – Декомпозиція, залежності та інтерфейси

● Проектування модулів і даних

Реляційна модель PostgreSQL у третій нормальній формі (3NF)

09 / 15



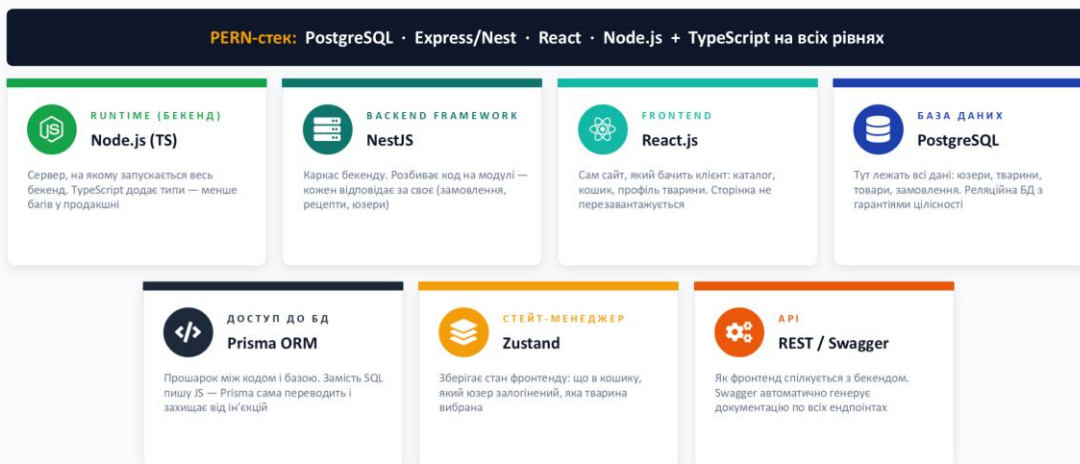
Філонок Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.9 – Проектування модулів і даних

● Аналіз та вибір технологій

Технологічний стек із обґрунтуванням вибору

10 / 15



Філонок Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.10 – Аналіз та вибір технологій

● Програмна реалізація модулів і бази даних

11 / 15

Що написано на бекенді й фронтенді — ключові модулі

РЕАЛІЗОВАНІ МОДУЛІ

Backend + Frontend

OrderService
Створює замовлення в транзакції — перевіряє залишки, запам'ятовує ціну на момент покупки, ставить статус PENDING_PRESCRIPTION для рецептурних ліків

MedicationCatalog
React-компонент каталогу. Через useMemo фільтрує препарати під обрану тварину: вид, вагу, обмеження

AuthModule
JWT-токени, паролі через bcrypt, middleware розділяє доступ для ADMIN і USER

PostgreSQL DDL
7 таблиць у 3NF, CHECK-обмеження, FOREIGN KEY з ON DELETE CASCADE / RESTRICT / SET NULL

```

order.service.ts · NestJS

@Injectable()
export class OrderService {
  async createOrder(userId, petId, items) {
    return this.prisma.$transaction(async (tx) => {
      let totalAmount = 0;
      let requiresPrescription = false;

      for (const item of items) {
        const product = await tx.product.findUnique(...);
        if (product.stock_quantity < item.quantity) {
          throw new BadRequestException(...);
        }
        totalAmount += product.price * item.quantity;
        if (product.is_prescription_required)
          requiresPrescription = true;
      }

      const initialStatus = requiresPrescription
        ? 'PENDING_PRESCRIPTION' : 'NEW';
      return await tx.order.create({ ... });
    });
  }
}

```

Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.11 – Програмна реалізація модулів і бази даних

● Вимоги до технічного та програмного забезпечення

12 / 15

Матриця відповідності нефункціональних вимог технічним рішенням (Табл. 3.1)

Вимога	Технічне рішення
Висока швидкість	Використання Node.js (Event Loop) та індексація БД PostgreSQL
Безпека даних	Впровадження JWT-токенів та Middleware валідації запитів
Надійність	Автоматичне резервне копіювання на хмарне сховище щоночі
Безпека з'єднання	Обов'язкове використання протоколу HTTPS для всіх з'єднань
Захист від ін'єкцій	Prisma ORM — захист від SQL-ін'єкцій на рівні абстракції
Ідентичність середовищ	Docker для пакування застосунку, БД та кеш-сервера в єдину одиницю
Адаптивність	Tailwind CSS — utility-класи, коректне відображення на десктопах і мобільних



ВИСНОВОК: кожна нефункціональна вимога має конкретне технічне рішення. Не просто слова — все втілено в код.

Філонюк Назар · ІПЗ-22-1 · ХНУ · 2026

Рисунок А.12 – Вимоги до технічного та програмного забезпечення

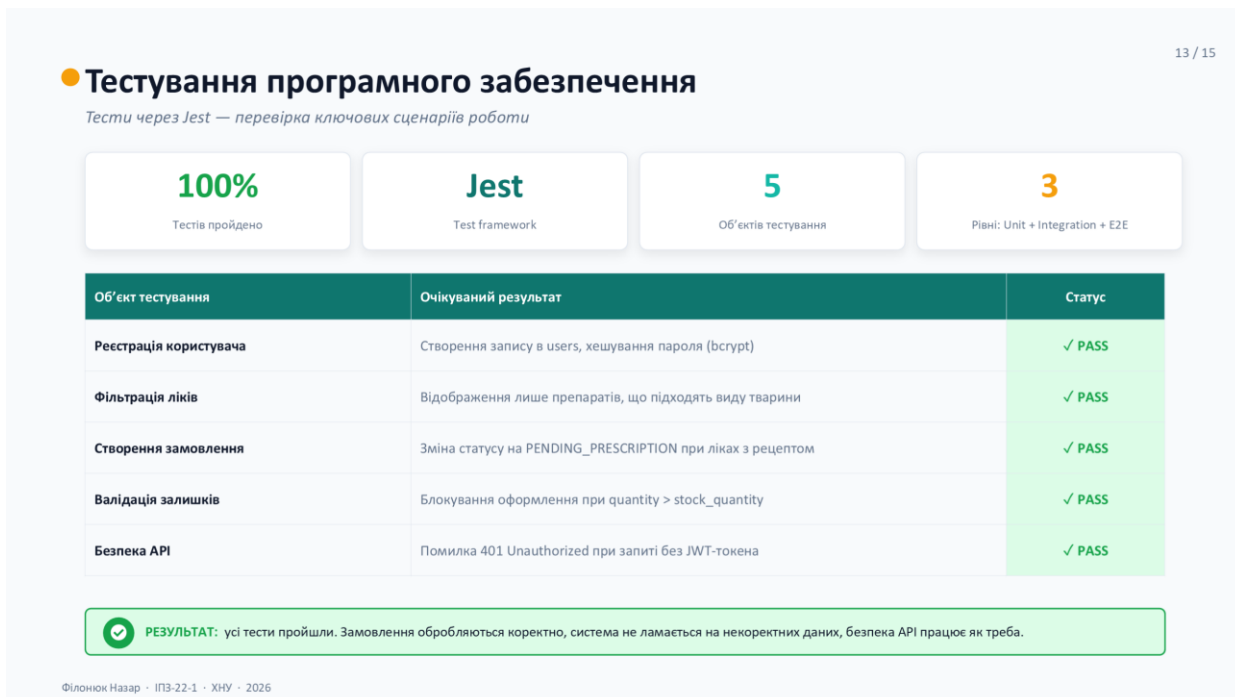


Рисунок А.13 – Тестування програмного забезпечення

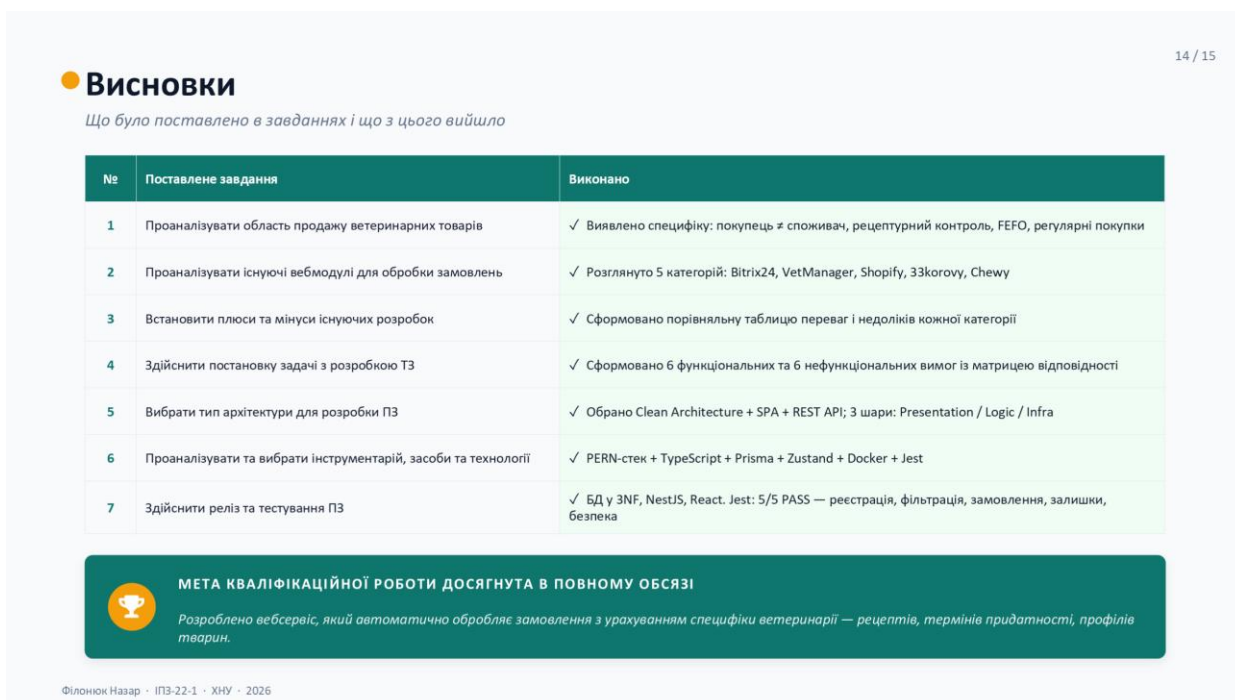


Рисунок А.14 – Висновки

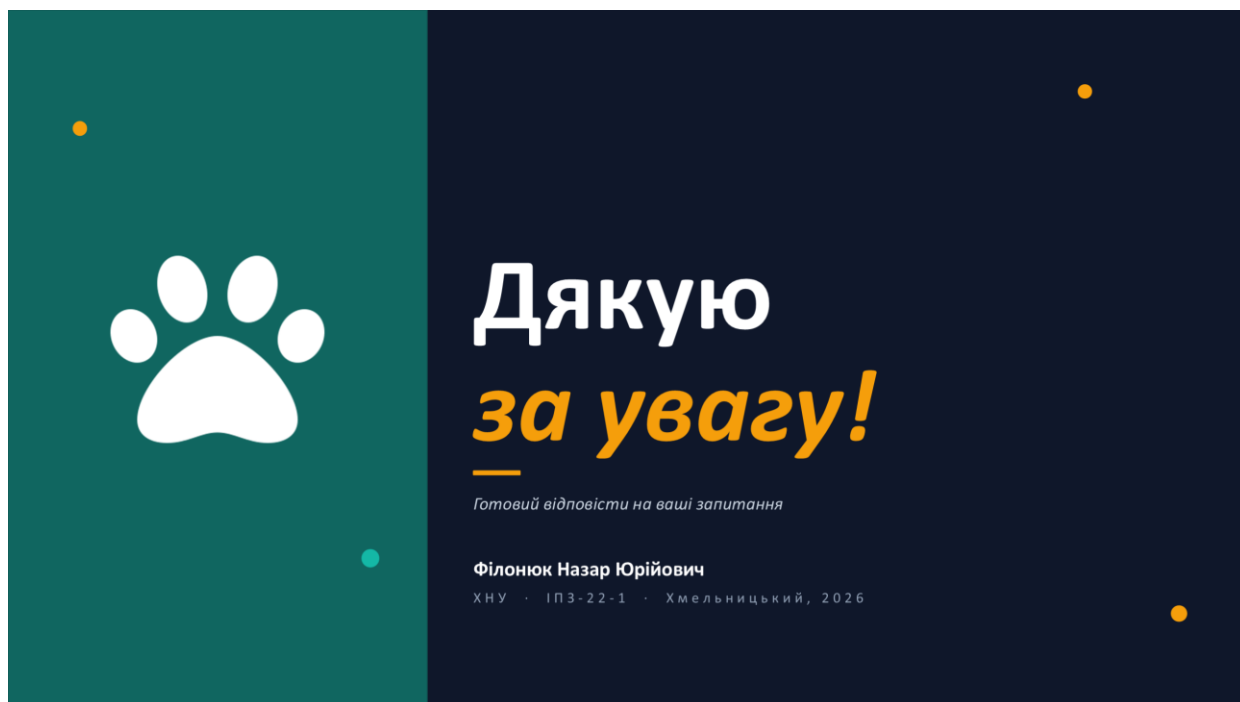


Рисунок А.15 – Дякую за увагу!

Додаток Б
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

1. Вступ

Проект спрямований на задоволення потреби користувача та створення зручного Вебзастосунок для продажу ветеринарних препаратів та виконується в рамках кваліфікаційної роботи бакалавра.

2. Обсяг робіт:

Розробка вебзастосунку:

- здійснити вибір та опис архітектури розроблюваного програмного продукту;
- провести аналіз та вибір технологій для розробки;
- реалізувати та протестувати розроблюваний програмний продукт.

3. Вимоги до дизайну вебзастосунку

Візуальний дизайн (UI Elements & Style) має враховувати кольорову палітру, фон, типографіку, просторові відступи, фотографії.

Кольорова палітра має містити такі основні кольори: теплі, природні відтінки, а саме, бежевий, коричневий, зелений, креманий для асоціації з натуральністю горіхів. Акцентні кольори можуть бути яскраві, але не кричущі (наприклад, червоний або оранжевий для кнопок заклику до дії, але з приглушеними відтінками) для солодоців. Фон обирається світлий, чистий, щоб продукти виділялися.

Заголовки передбачено чіткі, читабельні, сучасні шрифт, наприклад, Lato, Open Sans. Для основного тексту обираються легкі для читання шрифти з достатнім міжрядковим інтервалом, наприклад, Roboto, Merriweather. Прості, інтуїтивно зрозумілі іконки, а саме кошик, профіль, пошук, фільтри. Фотографії виключно високоякісні, професійні тобто такі, що мають чистий фон, добре

освітлення, акцент на текстурі, оскільки це критично для продажу ліків. Просторові відступи передбачають достатній «білий простір» навколо елементів для кращої читабельності та візуальної ієрархії.

4.Очікувані результати:

Детальний план проектування із вказаними етапами та термінами.

Прототипи дизайну вебзастосунку для затвердження.

5.Часові рамки:

01.06.2026 р.

6. Бюджет:

Доступний бюджет становить 2500 євро.

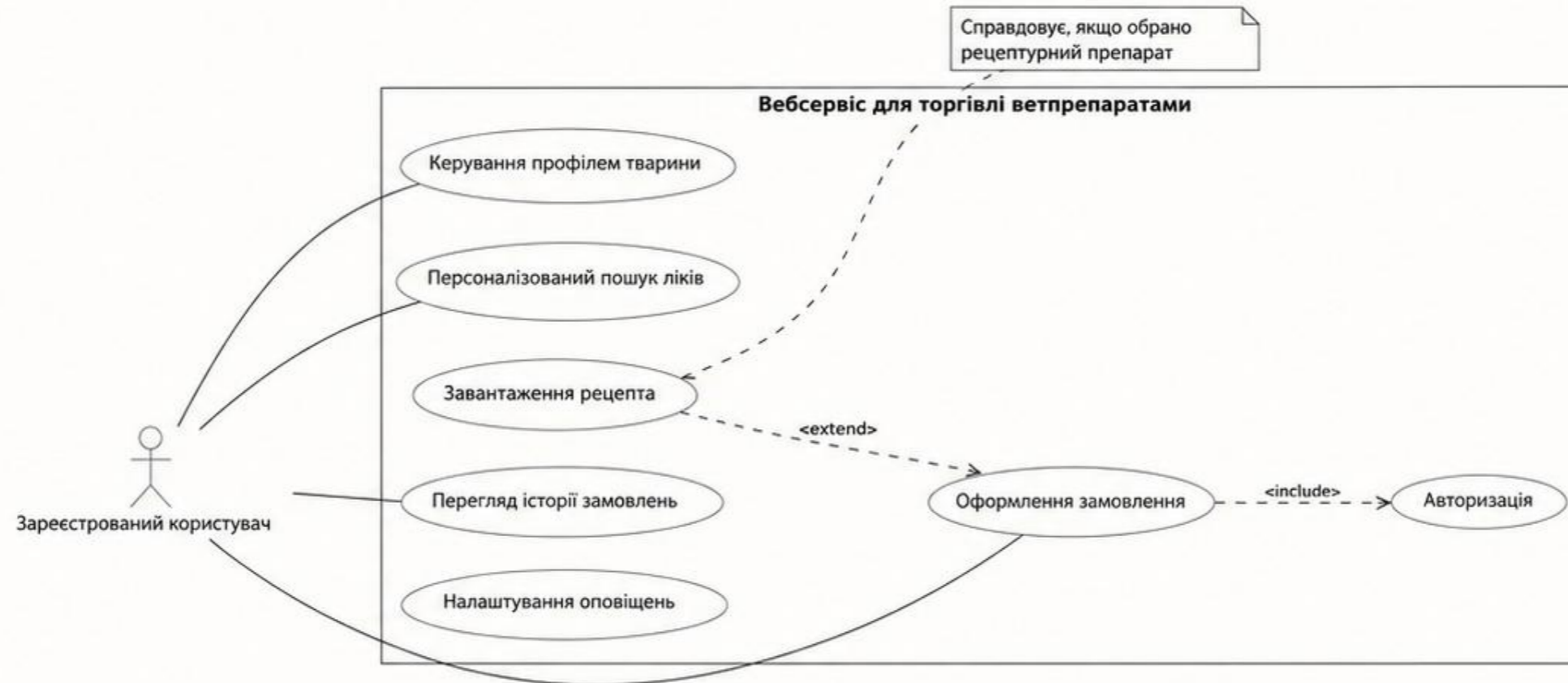
7. Подання пропозицій:

Пропозиції потрібно надсилати на електронну адресу:

afgn_copy@gmail.com.

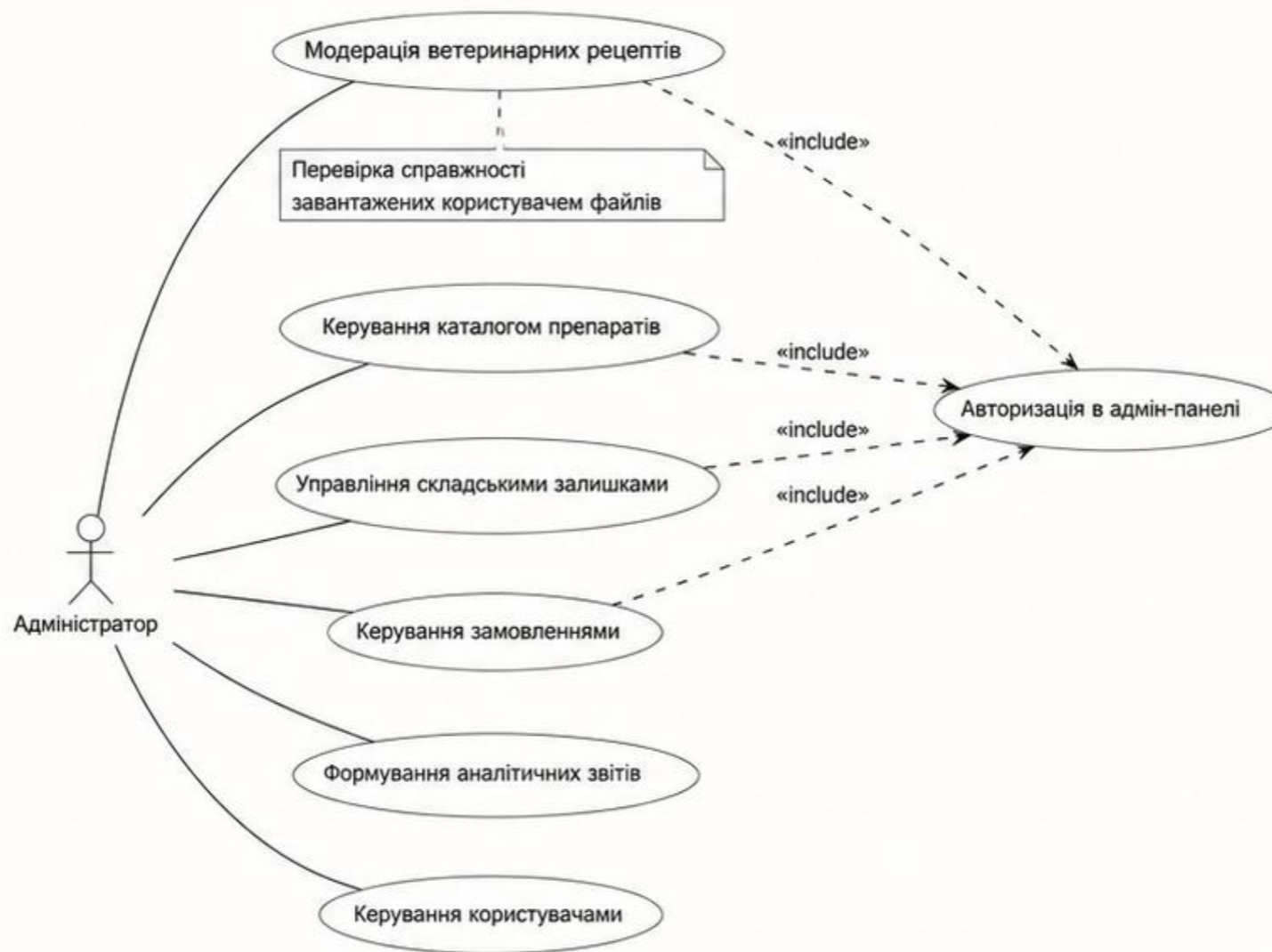
ГРАФІЧНА ЧАСТИНА

Діаграма варіантів використання для зареєстрованого користувача



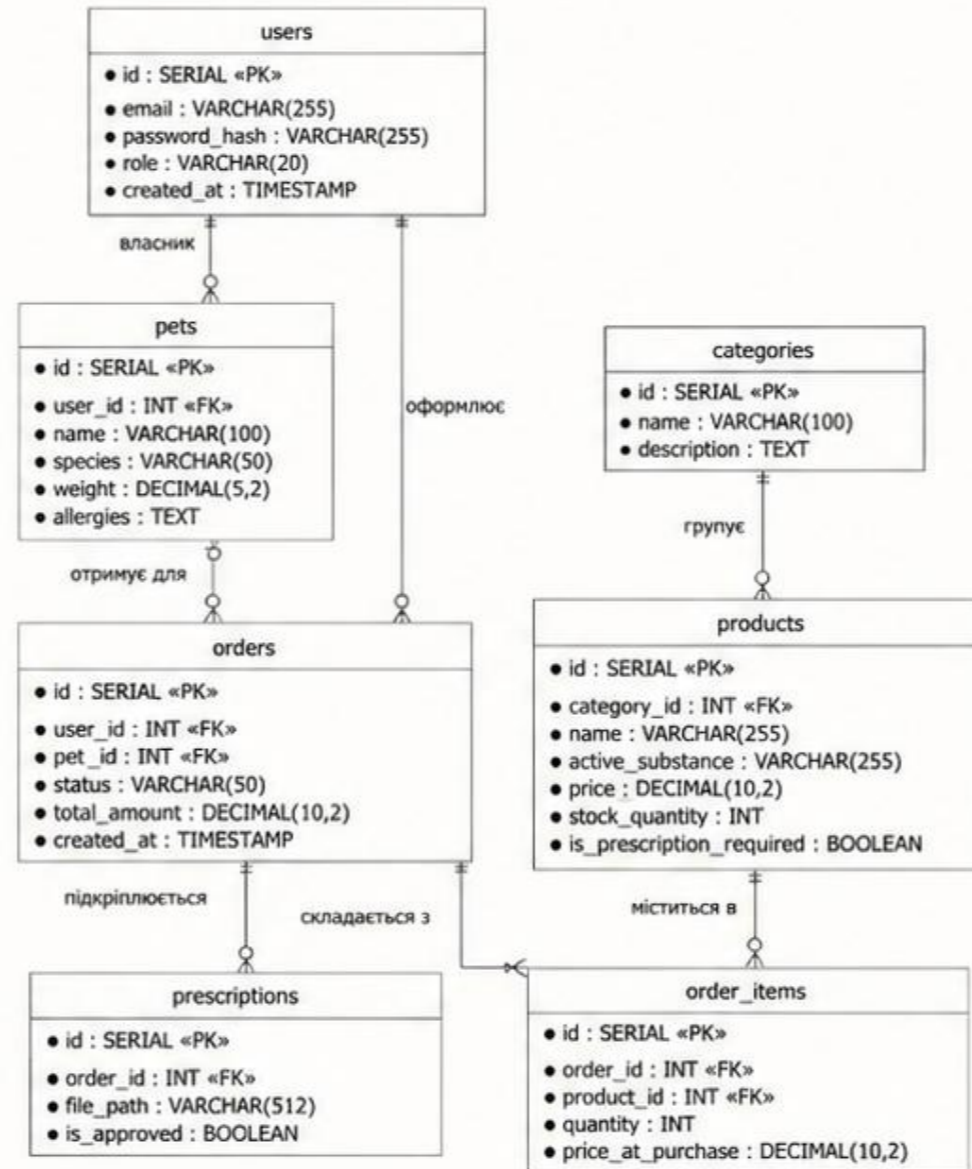
					КвРІПЗ.22011113.01.18.E8					
Зм.	Арк.	№ докум.	Підпис	Дата	Вебсервіс для продажу ветеринарних препаратів			Літера	Маса	Масштаб
Розробив		Філонюк Н.Ю.	<i>[Signature]</i>	25.05				Діаграма варіантів використання		
Керівник		Яшина О.М.	<i>[Signature]</i>	25.05						
Консульт.										
Н. Контр.		Форкун Ю.В.	<i>[Signature]</i>	25.05				Аркуш 1	Аркушів 3	
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	25.05				ХНУ, ІПЗ-22-1		

Діаграма варіантів використання для адміністратора



					КвРІПЗ.2201113.01.18.E8			
Зм.	Арк.	№ докум.	Підпис	Дата	Вебсервіс для продажу ветеринарних препаратів Діаграма варіантів використання	Літера	Маса	Масштаб
Розробив		Філонюк Н.Ю.		25.05				
Керівник		Яшина О.М.		25.05				
Консульт.								
Н. Контр.		Форкун Ю.В.		25.05				
Зав. каф.		Бедратюк Л.П.		25.05				
						Аркуш 1	Аркушів 3	
						ХНУ, ІПЗ-22-1		

ER-діаграма бази даних



Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Філонюк Н.Ю.	<i>[Signature]</i>	25.05
Керівник		Яшина О.М.	<i>[Signature]</i>	25.05
Консульт.				
Н. Контр.		Форкун Ю.В.	<i>[Signature]</i>	25.05
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	25.05

КВРІПЗ.2201113.01.18.E8			
Вебсервіс для продажу ветеринарних препаратів Діаграма варіантів використання	Літера	Маса	Масштаб
	Аркуш 1	Аркушів 3	
ХНУ, ІПЗ-22-1			

Супровідні документи

Завідувачу кафедри інженерії програмного
забезпечення проф. Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Філонюка Назара Юрійовича
факультет ІТ, ІV курс, група ІПЗ-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмно-технічного засобу AntiPlagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

2.01.2026

дата



підпис

Mon May 25 00:28:19 EEST 2026, Форкун Юрій Вікторович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (<http://ap.km.ua>) v-16.718

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: UA, US, RU. **Помилко в документах: 14%**

ID: 272083 Назва: Вебсервіс для продажу ветеринарних препаратів Додано в БД: 2026-05-25 Автора: Назар ФІЛОНЮК Керівники: канд. техн. наук, доцент Оксана ЯШИНА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	62625	480	3405 (5%)	48 (10%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Назар ФІЛОНЮК

Співавтор:

Назва: Вебсервіс для продажу ветеринарних препаратів

Науковий керівник: канд. техн. наук, доцент Оксана ЯШИНА

Підрозділ: Кафедра інженерії програмного забезпечення

Коефіцієнт подібності 1: 5.54%

Коефіцієнт подібності 2: 0.51%

Мікропробіми: 12

Заміна букв: 0

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2026-05-25 01:41:25.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32, ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

Дата

28.05.26

експерт

 (Петруха Н.В.)

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «Бакалавр»

Дипломник Філонюк Назар ЮрійовичТема Вебсервіс для продажу ветеринарних препаратівСпеціальність 121 – Інженерія програмного забезпечення**Обсяг кваліфікаційної роботи:**Кількість листів креслень 3; кількість сторінок записки 80

1. Короткий зміст пояснювальної записки та прийнятих рішень у кваліфікаційній роботі досліджено та проаналізовано предметну область, визначено функціональні та нефункціональні вимоги до програмного забезпечення. Проведено аналіз існуючих програмно-технічних рішень для онлайн-продажу товарів, розглянуто їх переваги й недоліки та обгрунтовано актуальність розробки. Спроектовано архітектуру вебсервісу, його модульну структуру та базу даних, обрано технології реалізації. Виконано програмну реалізацію модулів і бази даних, наведено інструкцію користувача та проведено тестування, за результатами якого підтверджено коректну роботу застосунку.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням усіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі обгрунтовано актуальність теми, визначено мету та завдання роботи. У першому розділі проведено дослідження предметної області, аналіз існуючого програмно-технічного забезпечення та визначено функціональні й нефункціональні вимоги. У другому розділі спроектовано архітектуру застосунку із застосуванням шаблонів проектування, розроблено логічну структуру бази даних та інтерфейс користувача, проаналізовано й обрано технології розробки. У третьому розділі виконано програмну реалізацію бази

даних і модулів системи, наведено інструкцію користувача та проведено тестування програмного забезпечення, що підтвердило його коректну роботу.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною, оскільки онлайн-продаж ветеринарних препаратів — затребуваний і недостатньо охоплений сегмент електронної комерції. Застосовано сучасні вебтехнології та мову програмування JavaScript, дотримано принципів модульної побудови застосунку.

5. Негативні сторони роботи Доцільно було б розширити функціональність пошуку товарів за додатковими фільтрами та передбачити інтеграцію онлайн-оплати. Окремі рішення потребують подальшого масштабування. Зазначені зауваження не впливають на загальну позитивну оцінку роботи.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми роботи й подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно з вимогами чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний і чіткий, що дозволяє зрозуміти викладений матеріал у межах тематики проектування. Графічний матеріал наочно відображає деталі проектування системи.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленому завданню та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ

д.т.н., професор, кафедра КИС ХНУ
Мисенко Сергій Миколайович

« 2 » червня 2026 р.


(підпис)

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів перевірки роботи, продукованими програмно-технічним засобом (ами), на наявність текстових збігів.

Назва кваліфікаційної роботи: «Вебсервіс для продажу ветеринарних препаратів»

Автор: Філонюк Назар Юрійович

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Спеціальність: 121 – Інженерія програмного забезпечення

Науковий керівник: Яшина Оксана Миколаївна, кандидат технічних наук, доцент

Після аналізу звіту/звітів зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається до захисту.	відповідає
2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована.	
3	Виявлені запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Виявлені запозичення частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнуті. Робота може бути допущена до захисту після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Коефіцієнт подібності 1 – 5,54 %, коефіцієнт подібності 2 – 0,51 %. Запозичення, виявлені в роботі, є законними і не є плагіатом; рівень подібності не перевищує допустимої межі. Робота допускається до захисту.

Дата 3.06.2026

Завідувач кафедри

Леонід БЕДРАТЮК

Гарант освітньої програми

Леонід БЕДРАТЮК

Керівник кваліфікаційної роботи

Оксана ЯШИНА