

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

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**Розробка ІС веб-застосунку для магазину інноваційних продуктів у світі
домашніх тварин**

здобувача Коломійця Дмитра Володимировича

Науковий керівник:

к.т.н., доцент

_____Маєвський О.В

**Проект допущений до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент.

_____Тішков Б.О.

Київ 2024

Міністерство освіти і науки України

**Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО_ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)

освітньо-професійної програми

_____ Іванченко Г.Ф.

“ ____ ” _____ 2024 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ ____ ” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти ***Коломійця Дмитра Володимировича***

денної форми навчання

на підготовку кваліфікаційного бакалаврського проекту
на тему: **« Розробка ІС веб-застосунку для магазину інноваційних продуктів у світі домашніх тварин »**

Тему затверджено наказом ректора Університету від 11.03 2024 р. № 529-ст.

Кваліфікаційний бакалаврський проект виконується на матеріалах
наукових публікацій, інтернет-ресурсів, технічної документації

План кваліфікаційного бакалаврського проекту

Розділ 1	Характеристика та аналіз предметної галузі (назва розділу)
Розділ 2	Розробка вимог і моделювання інформаційної системи (назва розділу)
Розділ 3	Проектування та реалізація компонентів системи (назва розділу)

Об'єкт дослідження: процеси та явища, обрані для дослідження, зокрема методи і технології управління продажами інноваційних продуктів для домашніх тварин.

Предмет дослідження: Інформаційні системи та комп'ютерні технології, що використовуються для додатку інноваційних продуктів для домашніх тварин.

Мета випускного бакалаврського проекту: Провести дослідження і розробити проектні рішення для створення інформаційної системи, що враховує особливості продажів інноваційних продуктів для домашніх тварин.

Завдання, які необхідно виконати для досягнення мети:

У розділі 1: Описати галузь інноваційних продуктів для домашніх тварин, вивчити структуру організації продажів, приділити увагу факторам, що впливають на створення інформаційної системи. Навести схему організаційної структури. Провести аналіз літературних джерел для виявлення сучасних методик автоматизації управління продажами.

У розділі 2: Проаналізувати і визначити вимоги до інформаційної системи магазину з продажу інноваційних продуктів для домашніх тварин. Дотримуючись стандартів, розробити постановку задач та алгоритми їх вирішення для автоматизації управлінських функцій у магазині. Створити інформаційну модель відповідно до методичних рекомендацій кафедри.

У розділі 3: Спроекувати компоненти системи: інформаційне, технічне та програмне забезпечення у відповідності з методичними рекомендаціями кафедри, обґрунтувати можливості та доцільність впровадження розробок на практиці. У додатках надати вихідні документи, запити чи програми.

Завдання підготував науковий керівник

О. В.Маєвський.

(ініціали, прізвище)

(підпис)

« »

2024 р.

Завдання одержав студент

Д. В. Коломієць

(ініціали, прізвище)

(підпис)

« »

2024 р.

ВІДГУК НАУКОВОГО КЕРІВНИКА

про кваліфікаційний бакалаврський проект
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

студента Коломійця Дмитро Володимировича

на тему: **Розробка ІС веб-застосунку для магазину інноваційних продуктів у світі домашніх тварин**

Дипломний проект здобувача присвячений розробці інформаційної системи веб застосунку магазину інноваційних продуктів для домашніх тварин. Актуальність питань розвитку ринку інноваційних продуктів є високою, з огляду на зростаючий попит на нові та сучасні рішення у сфері догляду за домашніми улюбленцями.

До ключових положень дипломного проекту належать проведення змістовних досліджень і формування стратегії прийняття рішень стосовно інтеграції сучасних технологій, таких як MS SQL, React та Node.js, при створенні ефективної інформаційної системи. У проекті розроблено ключові компоненти системи, включаючи базу даних, інтерфейс користувача та серверну частину. Дипломний проект здобувача заслуговує достатньо високої оцінки та може бути допущеним до захисту. Однак, слід звернути увагу на деякі аспекти документування процесу розробки та детальніше описати методи тестування системи. В цілому, проект є комплексним та демонструє високу якість виконання.

Науковий керівник

доцент, к.т.н. Маєвський О.В.

(підпис)

АНОТАЦІЯ

випускного бакалаврського проекту студента 4 курсу
Навчально-наукового інституту «Інститут інформаційних технологій в
економіці»

Коломійця Дмитра Володимировича, виконаного на тему:
« Розробка ІС веб-застосунку для магазину інноваційних продуктів у світі
домашніх тварин »

Київ: кафедра інформаційних систем в економіці, 2024 р.

У даній роботі досліджується та реалізується проект веб-застосунку для магазину інноваційних продуктів для домашніх тварин. Об'єктом дослідження є комплексна інформаційна система, що включає в себе веб-застосунок, спрямований на полегшення вибору та покупку продуктів для домашніх улюбленців, з урахуванням сучасних технологічних трендів і вимог ринку. Під час виконання роботи проведено глибокий аналіз потреб цільової аудиторії, визначено як функціональні, так і нефункціональні вимоги до застосунку.

Дипломний проект включає всебічну характеристику предметної галузі, зокрема опис актуальних інноваційних продуктів для домашніх тварин та аналіз поточного стану ринку. Також здійснено аналіз існуючих альтернативних рішень, що дозволяє оцінити їхні переваги та недоліки в контексті створення власного веб-застосунку. Розроблено детальні вимоги до інформаційної системи, включаючи моделювання основних процесів та структур, що забезпечують її функціонування.

У роботі також проведено проектування та реалізацію компонентів веб-застосунку, включаючи інформаційне, технічне, програмне та організаційне забезпечення. Всі ці компоненти детально описані та обґрунтовані з точки зору їхньої ефективності, надійності та доцільності використання в рамках даного проекту.

Висновки роботи містять узагальнення проведеного дослідження та аналіз отриманих результатів, що дозволяє зробити висновки щодо ефективності, доцільності та перспективності запропонованого веб-застосунку для магазину інноваційних продуктів у світі домашніх тварин. Результати підтверджують потенційну користь від впровадження даного веб-застосунку для цільової аудиторії, а також окреслюють напрямки вдосконалення та розвитку системи.

РЕФЕРАТ

Випускний бакалаврський проект містить 72 сторінок, 6 таблиць, 31 рисунок, список літератури з 12 найменувань.

РОЗРОБКА ІС ВЕБ-ЗАСТОСУНКУ ДЛЯ МАГАЗИНУ ІННОВАЦІЙНИХ ПРОДУКТІВ У СВІТІ ДОМАШНІХ ТВАРИН

Перелік ключових слів: веб-застосунок, магазин інноваційних продуктів, домашні тварини, інформаційна система, розробка.

Предметом дослідження є веб-застосунок для магазину інноваційних продуктів у світі домашніх тварин.

Об'єктом дослідження виступає інформаційна система, що включає в себе веб-застосунок для покупок продуктів для домашніх улюбленців.

Мета випускного бакалаврського проекту полягає в розробці і реалізації інформаційної системи для полегшення процесу вибору та покупки інноваційних продуктів для домашніх тварин.

Завданнями випускного бакалаврського проекту є аналіз потреб цільової аудиторії, визначення функціональних вимог до веб-застосунку, проектування та реалізація компонентів системи.

Апаратні та програмні засоби, що використовувались на при проектуванні: Операційна система Windows 10, мови програмування для розробки веб-сайту (Front-end, Back-end), використання Visual Studio Code.

Результати досягнуті в процесі роботи включають розроблений веб-застосунок, який дозволяє зручно та ефективно здійснювати покупки продуктів для домашніх тварин. Одержані результати можуть бути використані для покращення процесу обслуговування клієнтів магазинів інноваційних продуктів для домашніх тварин.

Рік виконання випускного бакалаврського проекту: 2024.

Рік захисту випускного бакалаврського проекту: 2024.

ЗМІСТ

Розділ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	10
1.1. Характеристика предметної області та об'єкта дослідження	10
1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі ..	12
Розділ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ/ПІДСИСТЕМИ	15
2.1. Аналіз і специфікація вимог до інформаційної системи/підсистеми	15
2.2. Постановка та алгоритм розв'язання задачі	20
2.2.1. Постановка задачі;	20
2.2.2. Алгоритм розв'язання задачі;	25
2.3. Моделювання інформаційної підсистеми	28
Моделювання поведінки системи;	28
Розподіл вимог за компонентами системи; Було створено діаграму трасування, на якій присутні вимоги, прецеденти, варіанти тестування та елементи користувацького інтерфейсу та відповідні зв'язки між ними.	33
Розділ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ/ПІДСИСТЕМИ.....	35
3.1 Інформаційне забезпечення;	35
Загальна характеристика інформаційного забезпечення;	35
Організація збору і передавання первинної інформації;	36
Побудова системи класифікації та кодування;	37
Проектування форм первинних документів, машинограм та відеокадрів;	38
Структура інформаційних масивів;	40
Вибір СКБД;	41
Даталогічна модель бази (сховища) даних.	42
3.2. Технічне забезпечення	46
Загальні положення та схема автоматизації.....	46
Структура комплексу технічних засобів;	47
Опис автоматизованого робочого місця (АРМ);	48
Схема мережі передачі даних;	49
3.3. Програмне забезпечення	50
Структура програмного забезпечення;	50
Системне програмне забезпечення;	50
Прикладне програмне забезпечення;	51
Програмна документація;	52
3.4. Результати реалізації інформаційної підсистеми	54
ВИСНОВОК:	61
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДОДАТКИ	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І
ТЕРМІНІВ

ІС	Інформаційна система
CASE	Computer-Aided Software Engineering
БД	Бази даних
ПП	Програмний продукт
ПЗ	Програмне забезпечення
UI	Користувацький інтерфейс

ВСТУП

У сучасному світі, де технологічний прогрес невпинно трансформує різні галузі життя, важливо виявляти інноваційні підходи до забезпечення комфорту та догляду за нашими домашніми тваринами. Із зростанням інтересу до інноваційних продуктів для домашніх улюбленців, стає яскравою актуальністю розробка веб-застосунку для магазину інноваційних товарів у світі домашніх тварин. Цей проект націлено на висвітлення та розвиток сучасних підходів до електронної комерції у сфері товарів для тварин. Використовуючи передові технології та стандарти розробки веб-застосунків, я прагну створити платформу, що забезпечить зручність та доступність інноваційних продуктів для домашніх тварин для широкого кола споживачів.

В цьому проекті не лише розглядається розробка функціонального веб-застосунку, але й покладено акцент на просування та популяризацію інноваційних підходів до догляду за тваринами серед їх власників. Це не лише створить зручний інструмент для покупки продуктів для домашніх улюбленців, але й сприятиме поширенню інформації про нові можливості та технології у цій галузі. Прагнення підвищити свідомість та зацікавленість власників тварин у використанні інноваційних рішень свідчить про глибоке розуміння потреб цільової аудиторії та бажання сприяти їхньому благополуччю та задоволенню.

Також проект спрямований на впровадження новаторських підходів у сфері догляду за домашніми тваринами. Я прагну створити екосистему, де власники тварин можуть з легкістю знаходити та придбавати передові рішення для забезпечення комфорту, здоров'я та щасливого життя своїх улюбленців.

При виконанні дипломного проекту було використано такі інструментальні засоби:

- операційна система Windows 10;
- середовище розробки VS Code;
- середовище Microsoft SQL Server 2019
- засіб прототипування Figma;
- CASE-засіб Enterprise Architect;
- текстовий редактор MS Word.

Розділ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної області та об'єкта дослідження

З зростанням інтересу до домашніх тварин і бажання власників забезпечувати їм найкращий догляд і комфорт, виникає необхідність у вдосконаленні та модернізації магазинів інноваційних продуктів для тварин. Одним із способів забезпечення доступу до передових рішень у цій області є розробка інформаційної системи у вигляді веб-застосунку для ефективного ведення та оптимізації операцій магазину.

Предметна область: Предметною областю є ринок продажу інноваційних товарів для домашніх тварин. Цей ринок постійно зростає, оскільки власники тварин шукають нові та покращені способи задоволення потреб своїх улюбленців. Інноваційні продукти охоплюють широкий спектр товарів, від спеціальних кормів та іграшок до технологічних рішень для здоров'я та комфорту тварин.

Об'єкт дослідження: Об'єктом дослідження є інформаційна система у вигляді веб-застосунку, спрямована на оптимізацію процесів управління магазином інноваційних товарів для домашніх тварин. Ця система має на меті полегшити вибір та придбання товарів, сприяти взаємодії з клієнтами та оптимізувати внутрішні процеси магазину для підвищення його ефективності.

Характеристика предметної області: Сучасний ринок товарів для домашніх тварин переживає значний ріст інтересу до інноваційних продуктів. Власники тварин активно шукають нові рішення для покращення життя своїх улюбленців. Тенденціями та інноваціями є розширення асортименту інноваційних товарів, таких як екологічні корми, інтелектуальні іграшки, засоби для догляду з використанням передових технологій.

Здоров'ям та Еко-підходами є зростання попиту на продукти, спрямовані на підтримку здоров'я тварин та врахування екологічних аспектів у виробництві товарів.

Характеристика Об'єкта дослідження: Описом об'єкта є інформаційна система яка представляє собою веб-застосунок, призначений для оптимізації управління та обслуговування магазину інноваційних товарів для домашніх тварин.

Функціональність: Забезпечення можливості зручного перегляду та вибору інноваційних товарів, ефективного оформлення замовлень, ведення клієнтської бази та моніторингу стану товарного асортименту.

Ключовими Характеристиками є взаємодія з базою даних товарів та клієнтів, ведення обліку замовлень, інтеграція з платіжними системами, а також створення особистих облікових записів для користувачів.

Метою є забезпечити зручний та ефективний механізм взаємодії магазину з клієнтами, полегшити процес вибору та покупки інноваційних продуктів для домашніх тварин.

Орієнтованою Кінцевою Аудиторією є Власники домашніх тварин, які цікавляться інноваційними та високоякісними товарами для своїх улюбленців.

Ці характеристики визначають контекст та обсяг дослідження у роботі та вказують на цільові результати розробки інформаційної системи. Така система буде включати інтерфейс для зручного перегляду та вибору товарів, механізми для ефективного оформлення замовлень, функціонал для ведення клієнтської бази та моніторингу стану товарного асортименту. Крім того, вона має забезпечувати безперебійну взаємодію з базою даних товарів та клієнтів, інтеграцію з платіжними системами та створення особистих облікових записів для користувачів. Отже, обсяг дослідження включає аналіз потреб ринку, вивчення проблем та можливостей управління магазином інноваційних товарів для домашніх тварин, а також розробку та впровадження функціональної інформаційної системи. Результатом має стати збільшення задоволення клієнтів, підвищення ефективності магазину та розширення його можливостей на ринку інноваційних товарів для домашніх тварин.

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

Інформаційні системи (ІС) та сучасні технології є необхідною складовою управління та розвитку бізнесу у галузі роздрібної торгівлі, особливо в контексті інноваційних продуктів для домашніх тварин. Цей розділ присвячений аналізу наукових джерел та практичного досвіду використання ІС у даній сфері, метою якого є виявлення основних тенденцій, переваг і викликів, з якими зіштовхуються підприємства цього сегменту ринку.

Одним з яскравих прикладів успішного використання ІС є компанія PetSmart, яка інтегрувала свої онлайн та офлайн сервіси, забезпечивши безшовний досвід для клієнтів. Використання аналітики даних та CRM систем дозволило компанії підвищити рівень обслуговування клієнтів та оптимізувати маркетингові кампанії (Tiwari і Kumar, 2021). Підприємства, що спеціалізуються на продажі товарів для домашніх тварин, все частіше використовують інформаційні системи для автоматизації бізнес-процесів. Наприклад, системи управління складом дозволяють ефективно контролювати запаси, впровадження POS-систем спрощує процес продажу, а аналітичні системи допомагають в прийнятті стратегічних рішень.

З розвитком електронної комерції, онлайн магазини для домашніх тварин стають все більш популярними. Платформи, такі як Amazon, пропонують широкий асортимент продуктів для домашніх тварин з можливістю швидкої доставки та зручного обслуговування. Це створює значні переваги для споживачів, хоча і вимагає від компаній адаптації своїх бізнес-процесів для ефективної роботи в онлайн-середовищі (Forbes, 2021). ІС дозволяють підприємствам збільшувати ефективність, підвищувати рівень обслуговування клієнтів та оптимізувати бізнес-процеси. Проте існують виклики, такі як високі витрати на впровадження систем, складнощі у інтеграції з існуючими інфраструктурами та необхідність навчання персоналу. Однак, для магазинів

інноваційних товарів для домашніх тварин, особливо тих, що спеціалізуються на унікальних продуктах або нішевих ринках, розробка власної інформаційної системи може стати конкурентною перевагою. Це дозволяє їм не лише підтримувати контроль над процесами продажу та обслуговування, але і створювати унікальний користувацький досвід, що відповідає конкретним потребам власників домашніх тварин.

Таблиця 1.2

Порівняння існуючих інформаційних систем

Веб-додаток	Функціональність	Дизайн	Інтеграція інноваційних технологій	Інтерактивність та залучення користувачів
PetSmart	Великий вибір продуктів	Привабливий	Використання віртуальної реальності для відображення товарів	Можливість коментування та оцінювання товарів, участь у групах спільноти
Chewy	Автоматичні підписки	Сучасний	Використання штучного інтелекту для рекомендацій товарів	Онлайн-форуми для обміну досвідом та порадами
Amazon Pets	Швидка доставка	Простий та зрозумілий	Використання голосових асистентів для покупок	Зворотний зв'язок через огляди, рецензії та питання користувачів
Zooplus	Програма лояльності	Чистий та організований	Використання блокчейн-технологій для забезпечення безпеки та відстеження постачання	Блоги та інформаційні статті для освіти користувачів

Розвиток інноваційних продуктів, таких як розумні іграшки та пристрої для моніторингу здоров'я домашніх тварин, сприяє зростанню ринку. Такі

продукти не тільки задовольняють потреби власників тварин, але й створюють нові можливості для бізнесу. Інновації в цій галузі часто базуються на використанні ІС для збору та аналізу даних, що дозволяє створювати більш ефективні та зручні продукти (TechCrunch, 2022).

Аналіз літературних джерел та практичного досвіду показує, що ІС та сучасні технології відіграють критичну роль у розвитку бізнесу в сфері продуктів для домашніх тварин. Використання ІС дозволяє підприємствам підвищити ефективність управління, покращити обслуговування клієнтів та створювати інноваційні продукти. Зважаючи на швидкі темпи розвитку технологій, подальші дослідження та впровадження новітніх ІС є необхідними для збереження конкурентоспроможності на ринку. Літературні джерела зосереджуються на різних аспектах використання ІС у продажу товарів для домашніх тварин, включаючи управління запасами, CRM, аналітику даних, онлайн-продажі та багато іншого. Підходи до впровадження цих систем розглядаються як у контексті великих ланцюгів магазинів, так і в маленьких магазинах та стартапах.

Я вважаю що успішна інформаційна система повинна відповідати потребам користувачів. Розуміння їхніх потреб та вимог допоможе розробити оптимальні рішення щодо функціональності та інтерфейсу системи. Розуміння потреб і вимог користувачів є ключовим для розробки інформаційної системи, що успішно відповідає вимогам сучасного ринку продуктів для домашніх тварин та сприяє збереженню конкурентоспроможності підприємства.

Розділ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ/ПІДСИСТЕМИ

Розробка інформаційної системи (ІС) для магазину інноваційних продуктів у світі домашніх тварин включає визначення вимог та моделювання підсистем, які забезпечать ефективне управління бізнесом та обслуговування клієнтів. Цей розділ присвячений аналізу вимог до ІС, а також методам та інструментам моделювання системи, що дозволить забезпечити її відповідність потребам бізнесу та очікуванням користувачів.

2.1. Аналіз і специфікація вимог до інформаційної системи/підсистеми

Стейкхолдер	Бізнес-ціль	Бізнес-вимоги
Власник магазину	Збільшення оборотів та прибутку шляхом ефективного продажу інноваційних товарів для домашніх тварин.	<ul style="list-style-type: none"> • Привабливий інтерфейс веб-застосунку для залучення клієнтів. • Можливість швидко додавати нові продукти. • Зручний функціонал для обробки замовлень. • Забезпечення безпечних транзакцій.
Клієнти	Отримання інформації про інноваційні продукти для своїх домашніх улюбленців.	<ul style="list-style-type: none"> • Зручний пошук товарів за видами та характеристиками домашніх тварин. • Детальні описи продуктів та візуальне представлення. • Простий процес замовлення та оплати.
Постачальники	Забезпечення постачання якісних та інноваційних товарів для магазину.	<ul style="list-style-type: none"> • Можливість зареєструвати свої продукти та взаємодіяти зі замовленнями. • Можливість отримання інформації про запити та замовлення для планування виробництва та поставок. • Можливість інтеграції з системами зберігання товарів.
Відділ доставки	Забезпечення швидкої та надійної доставки продуктів клієнтам.	<ul style="list-style-type: none"> • Можливість отримання та обробки інформації про замовлення для ефективного планування маршрутів доставки. • Можливість виконання доставок у встановлені терміни та забезпечення безпеки товарів.

Рисунок 2.1.1 Бізнес-вимоги до системи

Веб-застосунок для магазину інноваційних продуктів у світі домашніх тварин має задовольняти різноманітні потреби та очікування основних стейкхолдерів: власників магазину, клієнтів, постачальників і відділу доставки. Кожна група стейкхолдерів має свої унікальні бізнес-цілі, які повинні бути враховані при розробці системи.

Для власників магазину головною метою є збільшення обороту та прибутку через ефективний продаж інноваційних товарів. Для досягнення цієї мети веб-

застосунок повинен мати привабливий інтерфейс, що сприяє залученню клієнтів, а також зручні функції для обробки замовлень та безпечних транзакцій. Важливо, щоб власники могли швидко та легко додавати нові продукти, щоб підтримувати актуальність асортименту.

Клієнти, в свою чергу, шукають зручний спосіб знаходити та купувати інноваційні продукти для своїх домашніх улюбленців. Система повинна забезпечувати легкий пошук товарів за видами та характеристиками тварин, надавати детальні описи та візуальні представлення продукції, а також пропонувати простий процес замовлення і оплати.

Постачальники зосереджені на забезпеченні магазину якісними та інноваційними товарами. Їм необхідно мати можливість реєструвати свої продукти та взаємодіяти із замовленнями. Система повинна дозволяти постачальникам отримувати інформацію про запити та замовлення, планувати поставки та інтегруватися з системами зберігання товарів.

Для відділу доставки важливо забезпечити швидку та надійну доставку продуктів клієнтам. Вони потребують інструментів для отримання та обробки інформації про замовлення, оптимізації маршрутів доставки та відстеження статусу замовлень. Крім того система має підтримувати можливість безпечних і своєчасних транзакцій.

Враховання всіх цих бізнес-вимог у процесі розробки веб-застосунку є ключовим для забезпечення успішної реалізації проекту. Послідовне врахування потреб кожної групи стейкхолдерів допоможе створити продукт, який буде не лише функціональним, але й задовольнить очікування користувачів та сприятиме покращенню їхнього досвіду взаємодії з магазином. Це, в свою чергу, позитивно вплине на досягнення загальних бізнес-цілей магазину, таких як збільшення обсягів продажів, підвищення рівня задоволеності клієнтів та підтримка конкурентоспроможності на ринку.

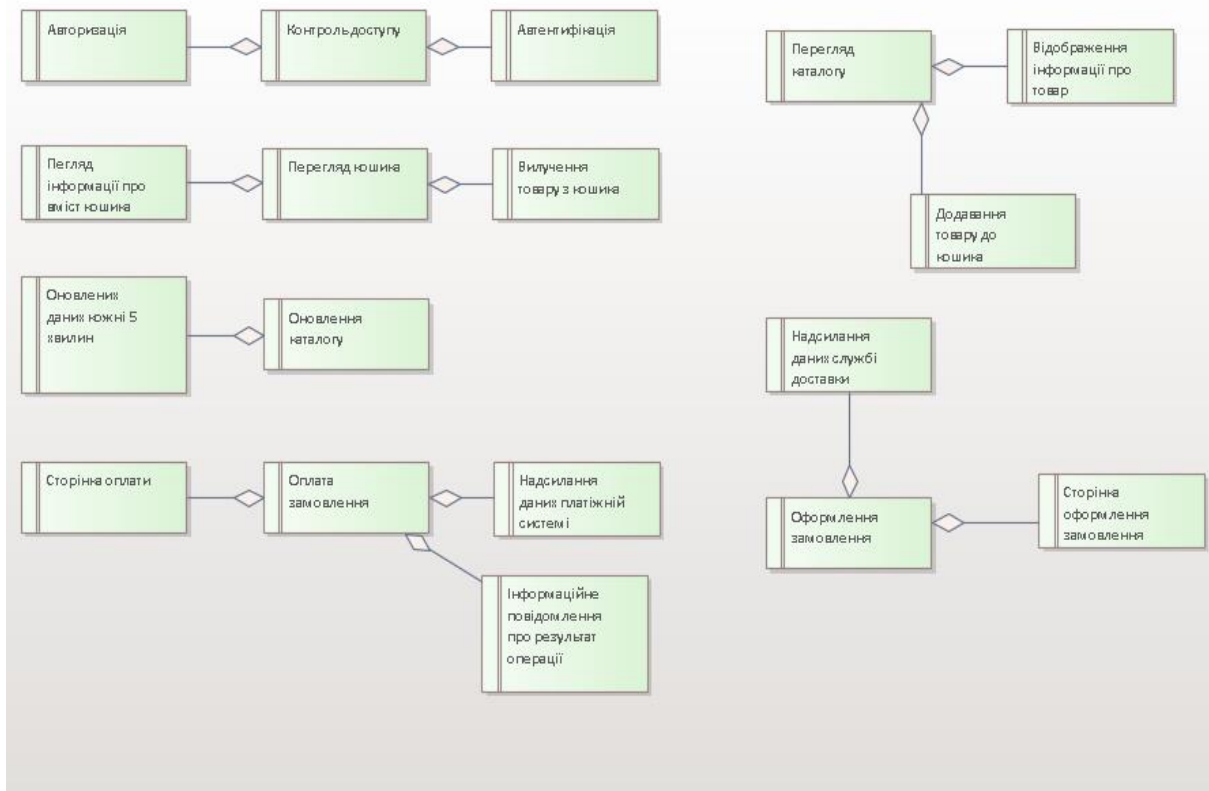


Рисунок 2.1.2 Функціональні вимоги до системи

Функціональні вимоги до системи - це опис того, що система повинна робити, які функції вона повинна виконувати для задоволення потреб користувачів та досягнення бізнес-цілей. Вони описують конкретні дії, які користувачі можуть виконувати в системі та як система повинна відреагувати на ці дії. Важливо, щоб функціональні вимоги відображали потреби користувачів та бізнес-цілі підприємства, а також були реалізовані з урахуванням найсучасніших технологій та кращих практик розробки програмного забезпечення. Крім того, функціональні вимоги повинні бути чітко сформульовані і документовані, щоб забезпечити однозначне розуміння та виконання їх розробниками програмного забезпечення.

<input checked="" type="checkbox"/> Автентифікація	Functional	Approved	Low	Medium
<input checked="" type="checkbox"/> Авторизація	Functional	Approved	Low	Medium
<input checked="" type="checkbox"/> Відображення інформації про товари	Functional	Approved	Medium	High
<input checked="" type="checkbox"/> Додавання товарів до кошика	Functional	Approved	Medium	High
<input checked="" type="checkbox"/> Інформаційне повідомлення про результат операції	Functional	Approved	Low	High
<input checked="" type="checkbox"/> Контроль доступу	Functional	Approved	Medium	Medium
<input checked="" type="checkbox"/> Надсилання даних платіжній системі	Functional	Approved	Medium	High
<input checked="" type="checkbox"/> Оновлення даних кожні 5 хвилин	Functional	Approved	Medium	High
<input checked="" type="checkbox"/> Оновлення каталогу	Functional	Approved	Medium	High
<input checked="" type="checkbox"/> Оплата замовлення	Functional	Approved	High	High
<input checked="" type="checkbox"/> Оформлення замовлення	Functional	Approved	High	High
<input checked="" type="checkbox"/> Перегляд інформації про вміст кошика	Functional	Approved	Low	Medium
<input checked="" type="checkbox"/> Перегляд каталогу	Functional	Approved	Low	Medium
<input checked="" type="checkbox"/> Перегляд кошика	Functional	Approved	Low	Medium
<input checked="" type="checkbox"/> Сторінка оплати	Functional	Approved	Low	High

Рисунок 2.1.3 Специфікація вимог

Специфікація вимог за допомогою менеджера специфікацій виконана за допомогою Enterprise Architect. Розробка специфікації вимог за допомогою менеджера специфікацій у програмі Enterprise Architect надає можливість створювати, організовувати та керувати вимогами до системи у зручному інтерфейсі. Також специфікація вимог дозволяє забезпечити чіткість, систематичність та відстеження вимог протягом усього життєвого циклу проекту.

Нефункціональні вимоги до системи: Було складено нефункціональні вимоги. Вони допомагають забезпечити високу якість продукту та його відповідність вимогам користувачів та ринку. Крім того, нефункціональні вимоги можуть мати великий вплив на архітектуру системи та визначати вибір технологій розробки.

Продуктивність. Система повинна забезпечувати швидкий відгук на дії користувачів. Веб-сайт повинен завантажуватися швидко. Швидке завантаження веб-сайту сприяє зручності користування та позитивному враженню від взаємодії з системою.

Безпека, тобто забезпечення конфіденційності та цілісності даних користувачів, а також засобів оплати. Забезпечення безпеки є критично важливим аспектом для будь-якої системи, особливо в разі зберігання та обробки конфіденційних даних користувачів та платіжної інформації. Реалізація

механізмів автентифікації та авторизації, а також застосування сучасних методів шифрування даних, допомагає запобігти несанкціонованому доступу та зберегти цілісність інформації.

Сумісність. Можливість інтеграції з різними веб-браузерами (Chrome, Firefox, Safari, Edge). Сумісність з різними веб-браузерами є ключовою для того, щоб забезпечити доступність системи для максимальної кількості користувачів. Це дозволяє забезпечити зручне користування веб-сайтом незалежно від вибраного браузера.

Доступність тобто використання семантичного HTML та адекватного опису елементів інтерфейсу для полегшення використання веб-сайту для всіх користувачів.

Масштабованість тобто можливість розширення функціональності та обсягу веб-застосунку для врахування зростаючого обсягу бізнесу. Забезпечення ефективної масштабованості важливо для того, щоб система могла витримувати зростаюче навантаження та функціонувати без перебоїв.

Документація: Надання документації з використання, налаштування та підтримки системи є ключовим для забезпечення користувачам зрозумілого та зручного користування системою, а також для підтримки її ефективного функціонування та розвитку в майбутньому.

2.2. Постановка та алгоритм розв'язання задачі

2.2.1. Постановка задачі;

Характеристика задачі, призначення та техніко-економічна сутність задачі: Задача розробки інформаційної системи (ІС) веб-застосунку для магазину інноваційних продуктів у світі домашніх тварин полягає у створенні ефективної платформи для продажу товарів. Метою є оптимізація процесів продажу, покращення обслуговування клієнтів, спрощення взаємодії з постачальниками та забезпечення надійної доставки продукції. Використання ЕОМ (електронно-обчислювальних машин) дозволяє автоматизувати ці процеси, що призводить до економії часу, зниження витрат та підвищення точності операцій. Перелік об'єктів управління включає: товари - інноваційні продукти для домашніх тварин, клієнти - власники домашніх тварин, які купують товари, замовлення - процеси обробки та виконання замовлень клієнтів, доставка - процеси, пов'язані з транспортуванням товарів до клієнтів.

Призначення й використання вихідної інформації: Вихідна інформація включає: Підтвердження замовлень для клієнтів, звіти про продажі та залишки товарів для власників магазину, маршрути та графіки доставки для відділу доставки. Використання цієї інформації дозволяє покращити управління запасами, забезпечити своєчасну доставку товарів та підвищити задоволеність клієнтів.

Періодичність розв'язання і термін видачі вихідної інформації: Підтвердження замовлень в режимі реального часу. Звіти про продажі щоденно, щотижнево, щомісячно. Запити на поповнення запасів за необхідності, на основі аналізу продажів і залишків. Маршрути та графіки доставки щоденно.

Умови припинення автоматизованого розв'язання задачі:

Автоматизоване розв'язання задачі може бути припинено у випадку технічних збоїв, необхідності оновлення програмного забезпечення або зміни бізнес-процесів, що потребують ручного втручання.

Зв'язки з іншими задачами: інтеграція з бухгалтерськими системами для автоматичного оновлення фінансової інформації. Зв'язок з системами управління запасами для своєчасного поповнення товарів. Взаємодія з CRM-системою для управління взаємовідносинами з клієнтами.

Розподіл дій між персоналом і технічними засобами: За нормальних умов діє автоматична обробка замовлень та оновлення запасів здійснюється системою. Персонал займається в цілому управлінням винятковими ситуаціями та взаємодією з клієнтами. У разі технічних збоїв персонал вручну обробляє замовлення та оновлює дані в системі. Також технічні фахівці займаються усуненням проблем та відновленням роботи системи.

Далі наведено інформаційну модель задачі, ця модель допомагає визначити, які дані та операції необхідні для виконання задачі, а також встановити зв'язки між ними. Відповідна інформаційна модель може полегшити комунікацію між розробниками та зацікавленими сторонами, сприяти

уникненню недорозумінь та допомогти виокремити ключові вимоги до системи. Крім того, інформаційна модель може стати важливим документом для документування та аналізу системи як на етапі розробки, так і під час подальшого удосконалення та супроводження. Ця модель задачі дозволяє виявити потенційні проблеми та визначити можливості для оптимізації системи ще на ранніх стадіях проекту, що допомагає уникнути зайвих витрат часу та ресурсів у майбутньому.

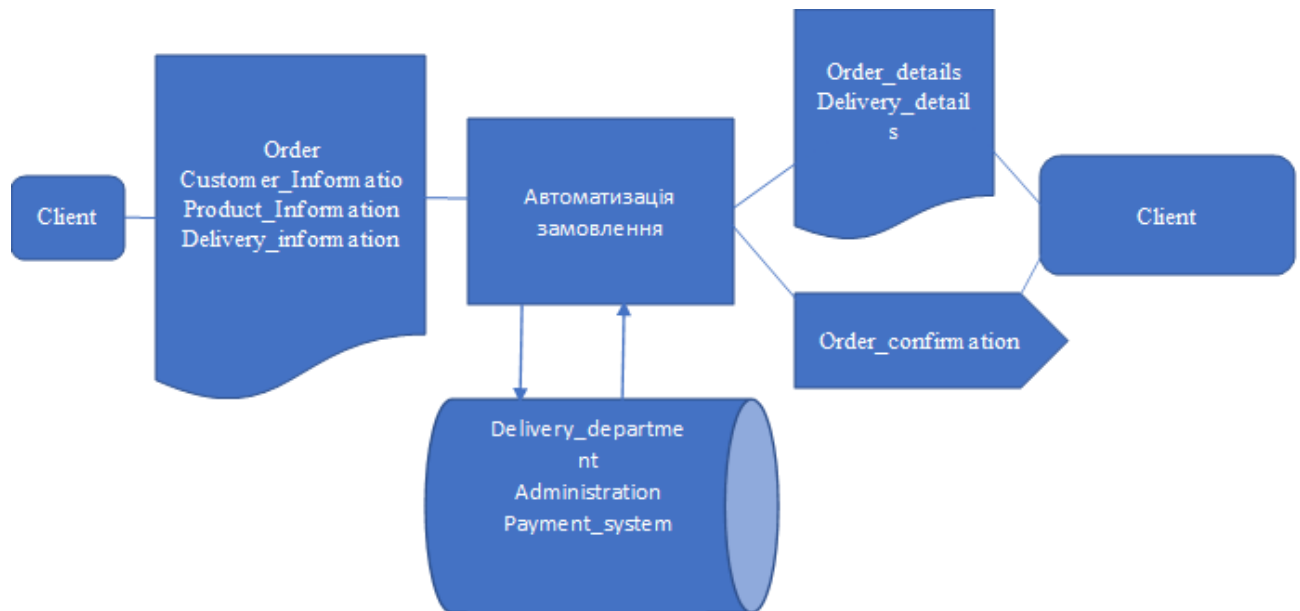


Рисунок 2.2.1 Інформаційна модель задачі.

Далі наведений перелік і опис вихідних повідомлень в табл. 2.2.1. Ці повідомлення визначають, яку інформацію система надсилає своїм користувачам або іншим системам у результаті виконання певних операцій або обробки даних. Розуміння вихідних повідомлень допомагає не лише забезпечити ефективну комунікацію між системою та користувачами чи іншими системами, але й гарантує, що інформація, яку надсилає система, відповідає очікуванням отримувачів та відповідає їхнім потребам. Крім того, визначення вихідних повідомлень допомагає підтримувати консистентність та стабільність взаємодії між системою та її користувачами чи іншими системами протягом усього життєвого циклу продукту. Знання про вихідні повідомлення також допомагає визначити ефективні методи комунікації та формати, що найкраще відповідають потребам користувачів чи інших систем, що сприяє покращенню загального досвіду використання системи.

ПЕРЕЛІК І ОПИС ВИХІДНИХ ПОВІДОМЛЕНЬ

№з /п	Назва вихідного повідомлення	Ідентифікатор	Форма подання та вимоги до неї	Періодичність видання	Термін видання й допустимий час	Користувачі інформації
1	Підтвердження замовлень	Confirmation_Order	Електронне повідомлення, текстовий формат	Реальний час	Негайно після оформлення замовлення	Клієнти
2	Звіти про продажі	Order_details	Електронний звіт, таблиця Excel	Щоденно, щотижнево, щомісячно	До 12:00 кожного дня (для щоденного), докінця робочого дня (щотижневого, щомісячного)	Власники магазину
3	Звіти доставки	Delivery_details	Електронний звіт, таблиця Excel	Щоденно, щотижнево, щомісячно	До 00:00 кожного дня (для щоденного), докінця робочого дня (щотижневого, щомісячного)	Власники магазину

Перелік і опис структурних одиниць вихідних повідомлень: **Номер замовлення:** Повна назва Order_Number. Ідентифікатор вихідного повідомлення - Confirmation_Order. Вимоги до точності та надійності обчислення: Важливо, щоб кожне замовлення мало унікальний номер без повторень. **Сума замовлення:** Повна назва: Order_Total. Ідентифікатор вихідного повідомлення: Confirmation_Order. Вимоги до точності та надійності обчислення - Потрібно обчислити точну суму замовлення на основі цін товарів і кількості. **Кількість товарів у замовленні:** Повна назва: Quantity_Ordered. Ідентифікатор вихідного повідомлення: Confirmation_Order. Вимоги до точності та надійності обчислення: Необхідно вказати точну кількість кожного товару, який замовлено. **Дата та час оформлення замовлення:** Повна назва: Order_DateTime. Ідентифікатор вихідного повідомлення: Confirmation_Order. Вимоги до точності та надійності обчислення: Дата та час мають бути точними та коректними. **Назва товару:** Повна назва: Product_Name. Ідентифікатор вихідного повідомлення: Sales_Reports. Вимоги до точності та надійності обчислення: Необхідно коректно відображати назву кожного товару. **Кількість проданого товару:** Повна назва: Quantity_Sold. Ідентифікатор вихідного повідомлення: Sales_Reports. Вимоги до точності та надійності обчислення: Потрібно обчислити точну кількість проданого кожного товару.

Перелік і опис вхідних повідомлень наведений в табл. 2.2.2. Розуміння вхідних повідомлень є важливим етапом у проектуванні будь-якої інформаційної системи. Вони визначають те, яка інформація потрібна системі для виконання своїх функцій та як вона повинна реагувати на цю інформацію. Детальний аналіз вхідних повідомлень допомагає зрозуміти потреби користувачів та визначити, як система може найкраще їх задовольняти. При проектуванні системи важливо враховувати різноманітність можливих вхідних повідомлень, адже вони можуть бути представлені у різних форматах, від текстових запитів користувачів до структурованих даних, що надходять через API. Ретельний аналіз цих повідомлень дозволяє системі ефективно обробляти інформацію, реагувати на зміни у вхідних даних та забезпечувати коректну роботу в усіх можливих сценаріях взаємодії з користувачами та зовнішніми системами.

Таблиця 2.2.2

ПЕРЕЛІК І ОПИС ВХІДНИХ ПОВІДОМЛЕНЬ

№ з/п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1.	Замовлення	Order	Електронний	За необхідністю	Клієнти
2.	Інформація про клієнтів	Customer_Information	Електронний	При реєстрації, при покупках	Клієнти
3.	Інформація про товари	Product_Information	Електронний	При внесенні нових товарів	Постачальники, власники магазину
4.	Інформація про доставку	Delivery_Information	Електронний	При оформленні замовлення, під час доставки	Клієнти, служби доставки

Перелік структурних одиниць вхідних повідомлень: **Номер замовлення:** Повна назва - Order_Number. Вимоги до точності числового значення - Кожне замовлення має унікальний номер, який не повинен повторюватися. Джерелом інформації є база даних замовлень. Ідентифікатор джерела інформації: Order.

Ідентифікатор клієнта: Повна назва - Customer_ID. Вимоги до точності числового значення - Унікальний ідентифікатор кожного клієнта. Джерело інформації - База даних клієнтів. Ідентифікатор джерела інформації - Customer_Information.

Кількість одиниць товару у замовленні: Повна назва: Quantity_Ordered. Вимоги до точності числового значення: Кількість товару у замовленні має бути цілим додатнім числом. Джерело інформації: База даних замовлень. Ідентифікатор джерела інформації: Order.

Ціна товару за одиницю: Повна назва: Unit_Price. Вимоги до точності числового значення: Ціна товару за одиницю має бути додатнім числом, може містити дробову частину. Джерело інформації: База даних товарів. Ідентифікатор джерела інформації: Product_Information.

Дата та час оформлення замовлення: Повна назва: Order_DateTime. Вимоги до точності числового значення: Дата та час оформлення замовлення повинні бути точними. Джерело інформації: Веб-сайт клієнта. Ідентифікатор джерела інформації: Order

2.2.2. Алгоритм розв'язання задачі;

Використовувана інформація, Призначення використовуваної інформації:

- Замовлення: для обробки та виконання замовлень клієнтів.
- Інформація про клієнтів: для ідентифікації клієнтів та персоналізації обслуговування.
- Інформація про товари: для управління товарними запасами та цінами.
- Інформація про доставку: для планування та виконання доставок товарів.

Перелік масивів, що використовується під час розв'язання задачі, подано в табл 2.2.3. Кожен з цих масивів може мати свою власну структуру та використовуватися для різних обчислень або обробки даних у відповідності до поставленої задачі.

Таблиця 2.2.3

ПЕРЕЛІК МАСИВІВ ВИКОРИСТОВУВАНОЇ ІНФОРМАЦІЇ

Масив	Ідентифікатор	Максимальна кількість записів
1	2	3
Замовлення	Orders	100,000
Інформація про клієнтів	Customers	50,000
Інформація про товари	Products	10,000
Інформація про доставку	Deliveries	100,000

Результати розв’язання, призначення результатів розв’язання: Для підтвердження замовлень, тобто надсилання клієнтам смс підтверджень про отримання та обробку їхніх замовлень. Звіти про продажі, тобто аналіз з моніторингом продажів для власників магазину. Запити на поповнення запасів, тобто забезпечення своєчасного оновлення/поповнення товарів у магазині. Маршрути та графіки доставки, тобто планування та організація безпосередньої доставки товарів до клієнтів.

Таблиця 2.2.4

ПЕРЕЛІК МАСИВІВ РЕЗУЛЬТАТНОЇ ІНФОРМАЦІЇ

Масив	Ідентифікатор	Максимальна кількість записів
1	2	3
Підтвердження замовлень	Order_Confirmations	100,000
Звіти про продажі	Sales_Reports	10,000
Запити на поповнення запасів	Stock_Replenishment_Requests	5,000
Маршрути та графіки доставки	Delivery_Schedules	50,000

Ці масиви формуються для видачі вихідних повідомлень і зберігаються для подальшого використання, включаючи аналіз та покращення процесів продажу, поповнення запасів та доставки товарів. Результати автоматизованих процесів використовуються для прийняття рішень та оптимізації роботи магазину.

Математичне забезпечення системи: 1. Розрахунок суми замовлення: Сума замовлення S розраховується як добуток кількості одиниць товару Q_i на ціну за одиницю P_i для всіх товарів у замовленні: $S=Q_1 \cdot P_1 + Q_2 \cdot P_2 + \dots + Q_n \cdot P_n$ (2.1), де: S - сума замовлення, Q_i - кількість одиниць товару i , P_i - ціна одиниці товару i , n - загальна кількість різних товарів у замовленні.

2. Розрахунок загальної кількості проданих товарів: Загальна кількість проданих товарів Q_{total} розраховується як сума кількостей проданих одиниць кожного товару: $Q_{total} = Q_1 + Q_2 + \dots + Q_m$ (2.2), де: Q_{total} - загальна кількість проданих товарів, Q_j - кількість проданих одиниць товару j , m - загальна кількість різних товарів.

3. Розрахунок загальної виручки від продажів: Загальна виручка R розраховується як сума вартості проданих одиниць кожного товару: $R=Q_1 \cdot P_1 + Q_2 \cdot P_2 + \dots + Q_m \cdot P_m$ (2.3), де: R - загальна виручка від продажів, Q_j - кількість проданих одиниць товару j , P_j - ціна одиниці товару j , m - загальна кількість різних товарів.

Перелік прийнятих допущень і оцінок, відповідні допущення: Всі ціни товарів залишаються незмінними за період аналізу. Всі клієнти отримують свої замовлення у відповідно зазначений термін. Всі дані про безпосередню кількість проданих товарів точні і та своєчасно оновлюються в системі.

Оцінки відповідності моделі реальному процесу: Модель розрахунку суми замовлення та виручки відповідає реальному процесу, оскільки враховує всі необхідні параметри.

Модель управління запасами враховує динаміку продажів і потреби у поповненні, що дозволяє підтримувати оптимальний рівень запасів. Модель оптимізації маршруту доставки враховує всі пункти доставки і відстані між ними, що забезпечує мінімізацію часу і витрат на доставку.

2.3. Моделювання інформаційної підсистеми

Моделювання поведінки системи;

Розроблена діаграма прецедентів відображає різноманітні сценарії використання системи, фіксуючи типові взаємодії з її акторами. Цей інструмент узагальнено документує функціональні вимоги, демонструючи різні можливості взаємодії користувачів з системою. Діаграму прецедентів зображено на рис 2.3.1.

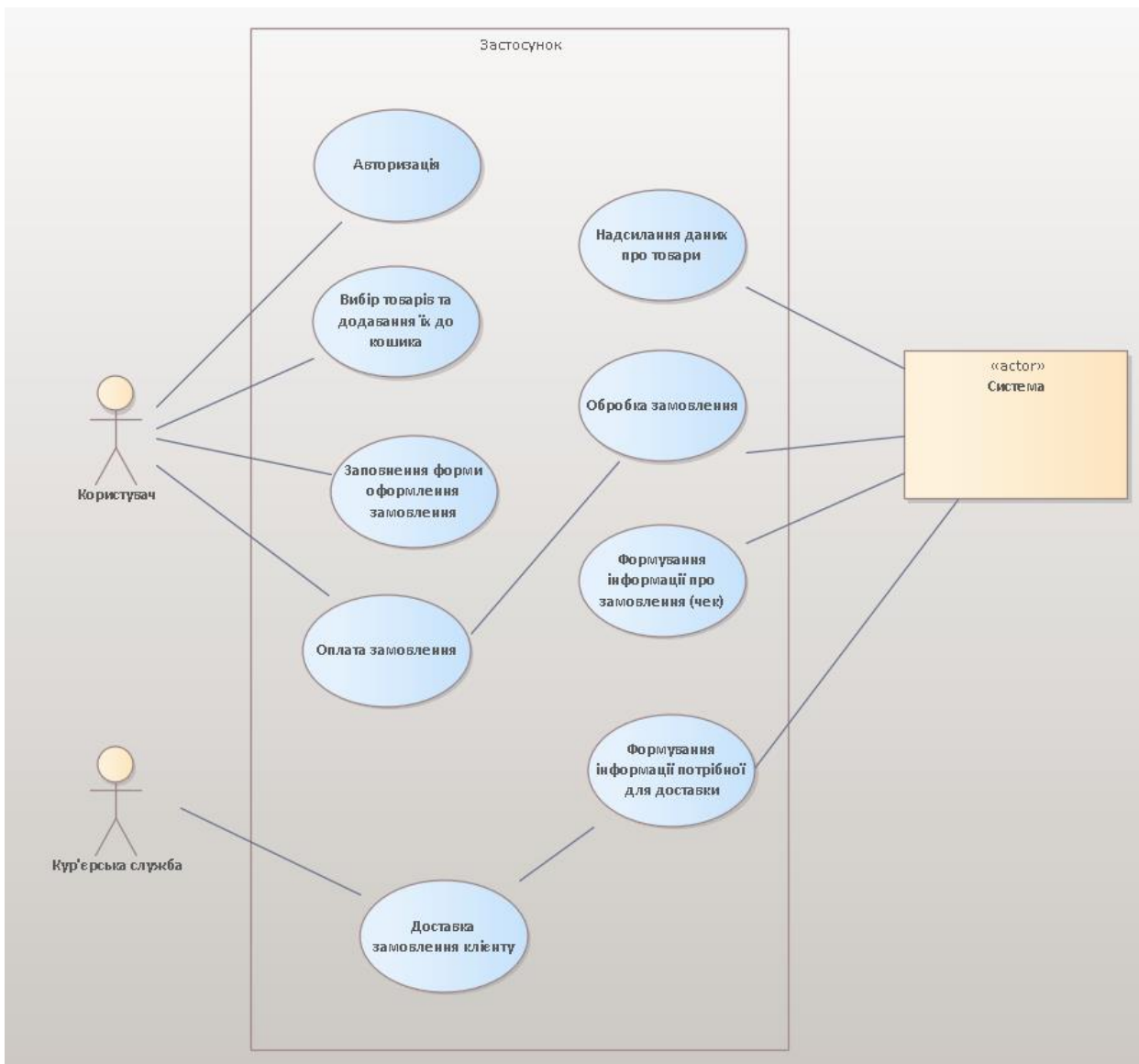


Рисунок 2.3.1 Діаграма Use Case

Далі зображено діаграму послідовності зображено на рисунку 2.3.2. Діаграма послідовності в контексті цієї роботи проектує взаємодію різних елементів системи веб-застосунку для магазину інноваційних продуктів для домашніх тварин під час виконання певних сценаріїв. Діаграма послідовності дозволяє уявити послідовність дій та повідомлень між різними частинами системи під час виконання конкретного сценарію, такого як процес вибору та придбання товару. Це допомагає розуміти, як система працює з точки зору функціональності та як користувачі взаємодіють з нею.

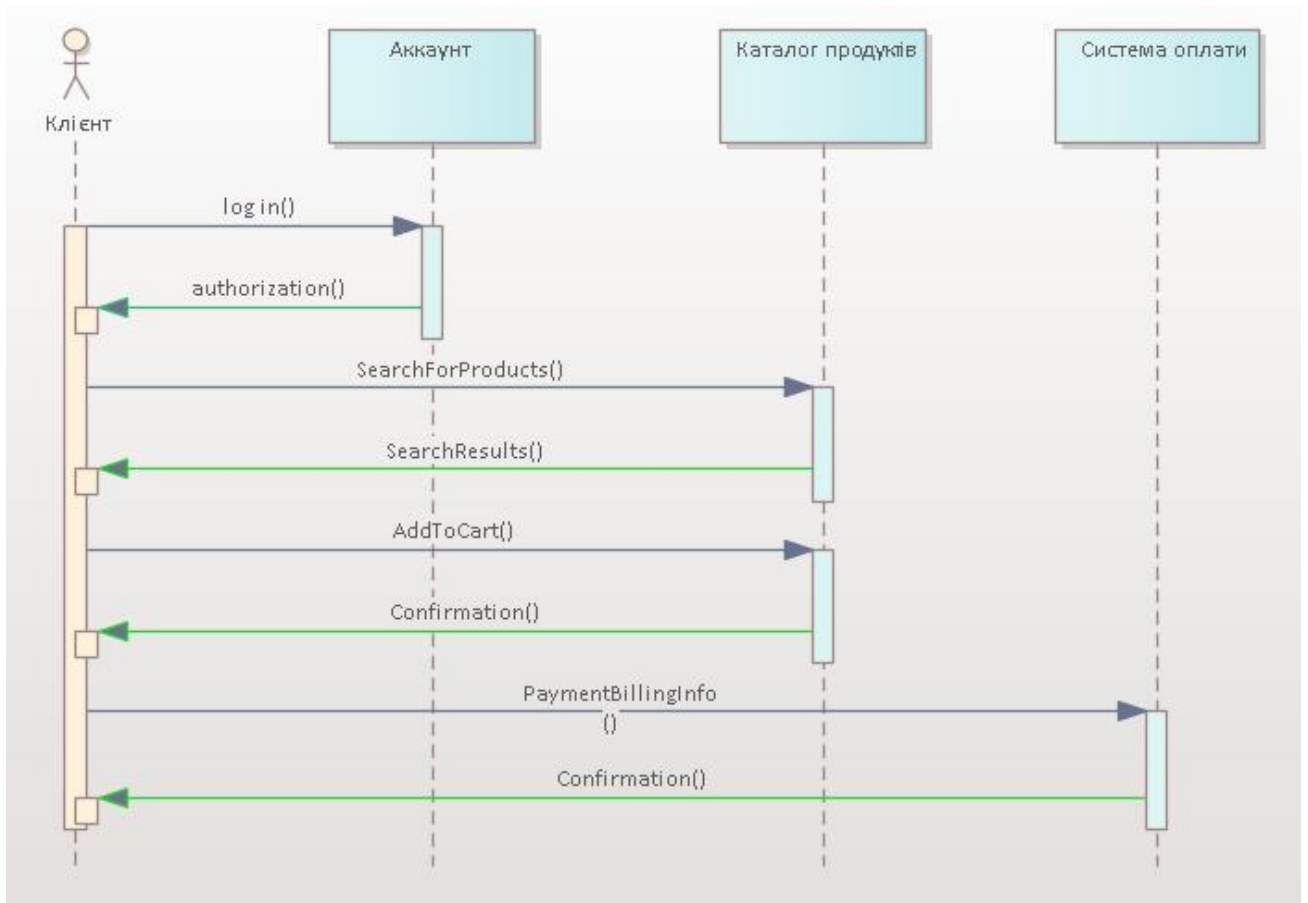


Рисунок 2.3.2 Sequence Diagram

Діаграма послідовності є важливим інструментом в аналізі та проектуванні системи, оскільки вона дозволяє візуалізувати та уточнити взаємодію між різними компонентами системи під час виконання конкретних сценаріїв. Це допомагає виявити потенційні проблеми та покращити ефективність системи ще на ранніх етапах розробки. Діаграма послідовності також може служити важливим документом для спілкування між розробниками та зацікавленими сторонами, що сприяє зрозумінню та узгодженню вимог до системи. Діаграма послідовності допомагає виявити можливі перешкоди

взаємодії між компонентами системи та уточнити послідовність операцій, необхідних для досягнення певного функціонального результату. Це сприяє підвищенню якості та надійності програмного забезпечення, а також полегшує процес розробки та тестування системи.

Модель структури системи; Було створено відповідну діаграму класів UML яка є видом діаграми, що використовується безпосередньо для моделювання статичної структури класів системи та зв'язків між ними. Ця діаграма дозволяє візуалізувати класи та їхні атрибути та методи, а також взаємозв'язки між класами. Вона є гарним інструментом для аналізу та відповідного проектування системи, що дозволяє краще розуміти її архітектуру та відповідну взаємодію між класами. Крім того, діаграма допомагає під час рефакторингу коду та розширенні функціональності системи, бо дає змогу візуалізувати поточну структуру та зв'язки класів для зручного аналізу та модифікацій. Використання діаграми класів UML допомагає зберігати документацію проекту у зручному та структурованому вигляді. Це стає важливим, особливо при роботі над великими або складними системами, де необхідно забезпечити чітку документацію для зрозуміння та підтримки програмного продукту в подальшому. Зазначу також, що використання діаграми класів UML сприяє підвищенню рівня абстракції при проектуванні системи. Це означає, що можна уникнути занурення у деталі реалізації на ранніх етапах, фокусуючись на концептуальній моделі системи. Такий підхід дозволяє зосередитися на важливих аспектах архітектури та функціональності, що сприяє збереженню простоти та ефективності проекту. Крім того, діаграма класів UML може служити основою для автоматичної генерації початкового коду. Деякі інтегровані середовища розробки надають можливість автоматично створювати структуру програми на основі цієї діаграми, що прискорює процес розробки та дозволяє уникнути помилок при ручному введенні коду. Діаграму класів зображено на рисунку 2.3.3.

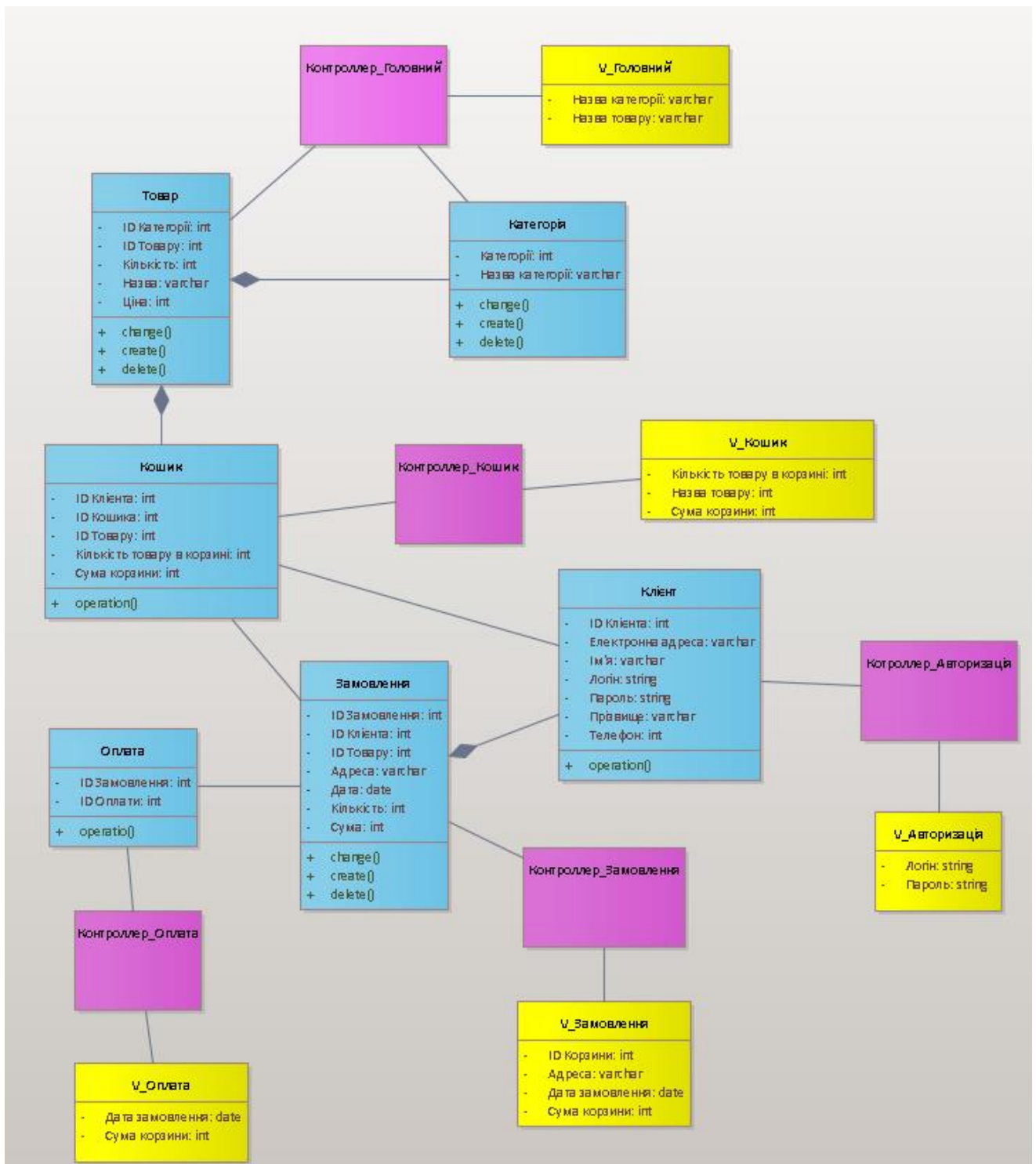


Рисунок 2.3.3 Діаграма класів

Також було створено діаграму визначення блоків яка є одним з видів діаграми у системному інженерінгу, який безпосередньо використовується для моделювання внутрішньої структури системи у вигляді ієрархії блоків різних типів. Ця діаграма дозволяє розкрити компоненти системи та їхні взаємозв'язки. Діаграма визначення блоків допомогла розкрити саме внутрішню структуру системи та її відповідні складові, що є гарно для розуміння її

архітектури та майбутнього проектування. Вона допомагає ідентифікувати основні компоненти системи та їх відповідні взаємозв'язки, що впливає на краще управління та на складність розвитку проекту.

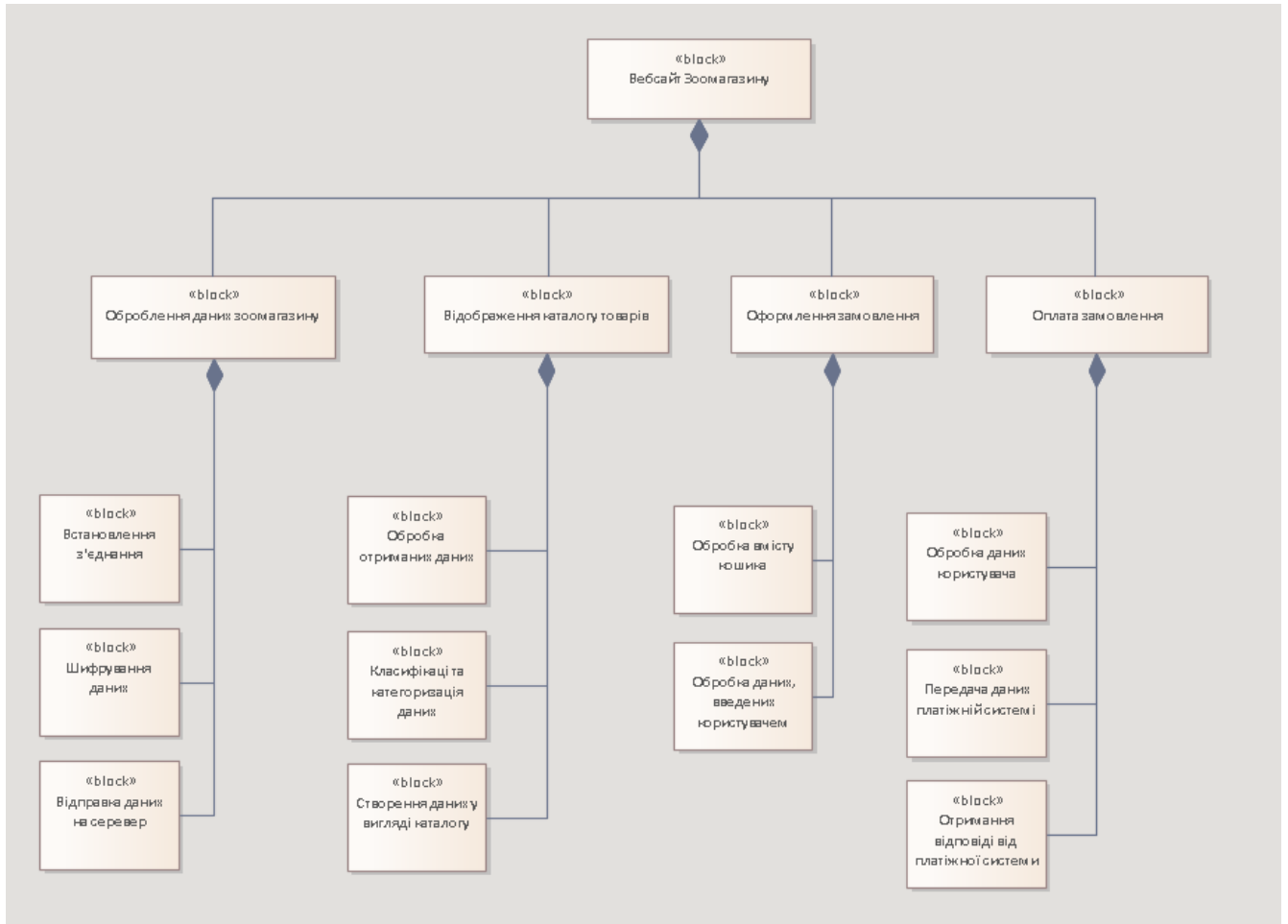


Рисунок 2.3.4 Діаграма визначення блоків

Також було розроблено діаграму внутрішніх блоків для подання їх як частин системи, їх з'єднання між собою безпосередньо через порти, а також потоків, що існують між частинами і портами. Діаграму внутрішніх блоків зображено на рисунку 2.3.5

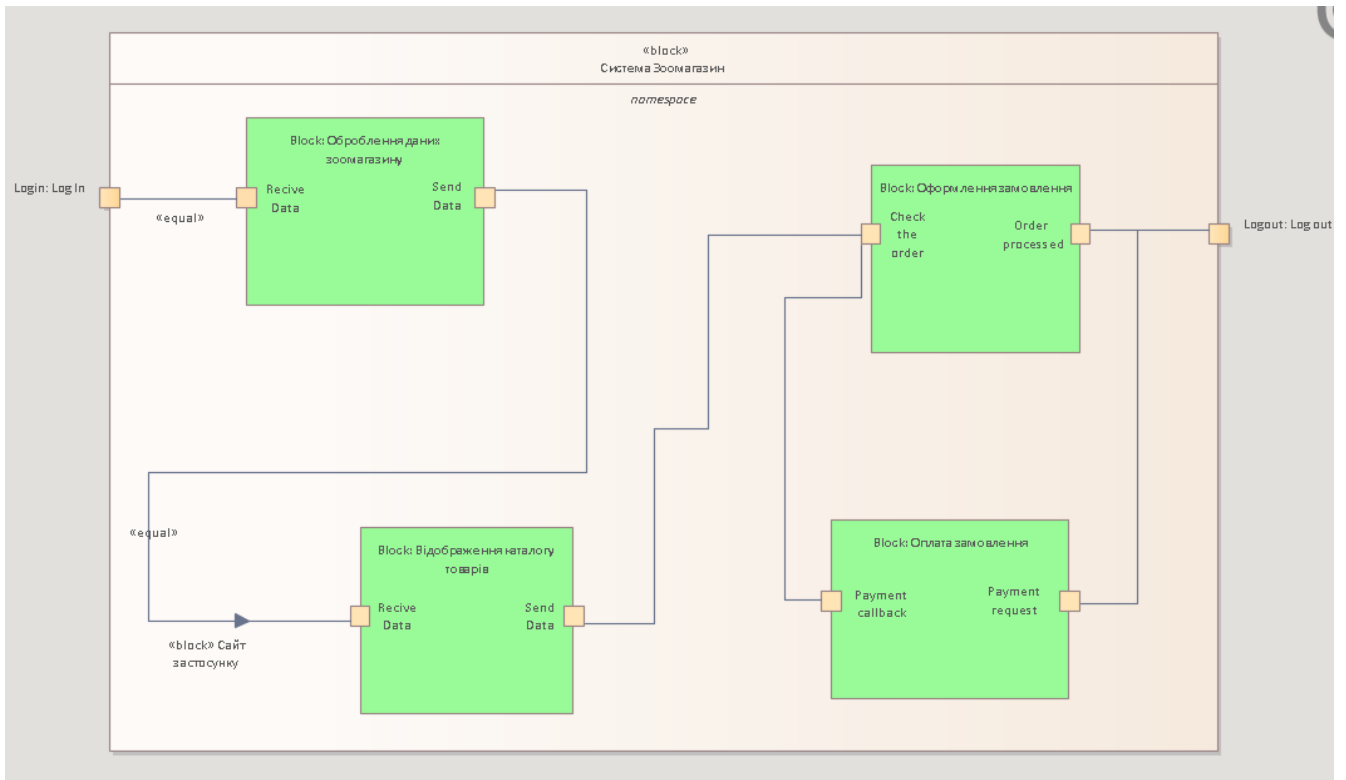


Рисунок 2.3.5 Діаграма внутрішніх блоків

Розподіл вимог за компонентами системи; Було створено діаграму трасування, на якій присутні вимоги, прецеденти, варіанти тестування та елементи користувацького інтерфейсу та відповідні зв'язки між ними.

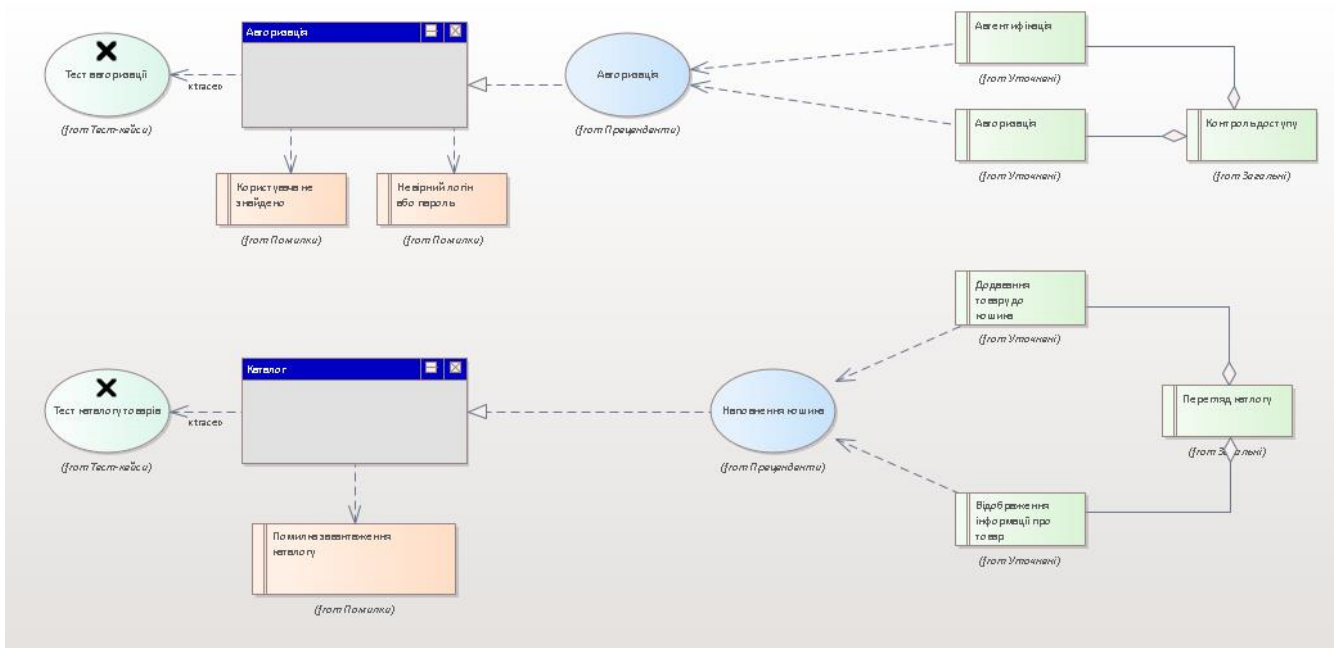


Рисунок 2.3.6 Діаграма трасування

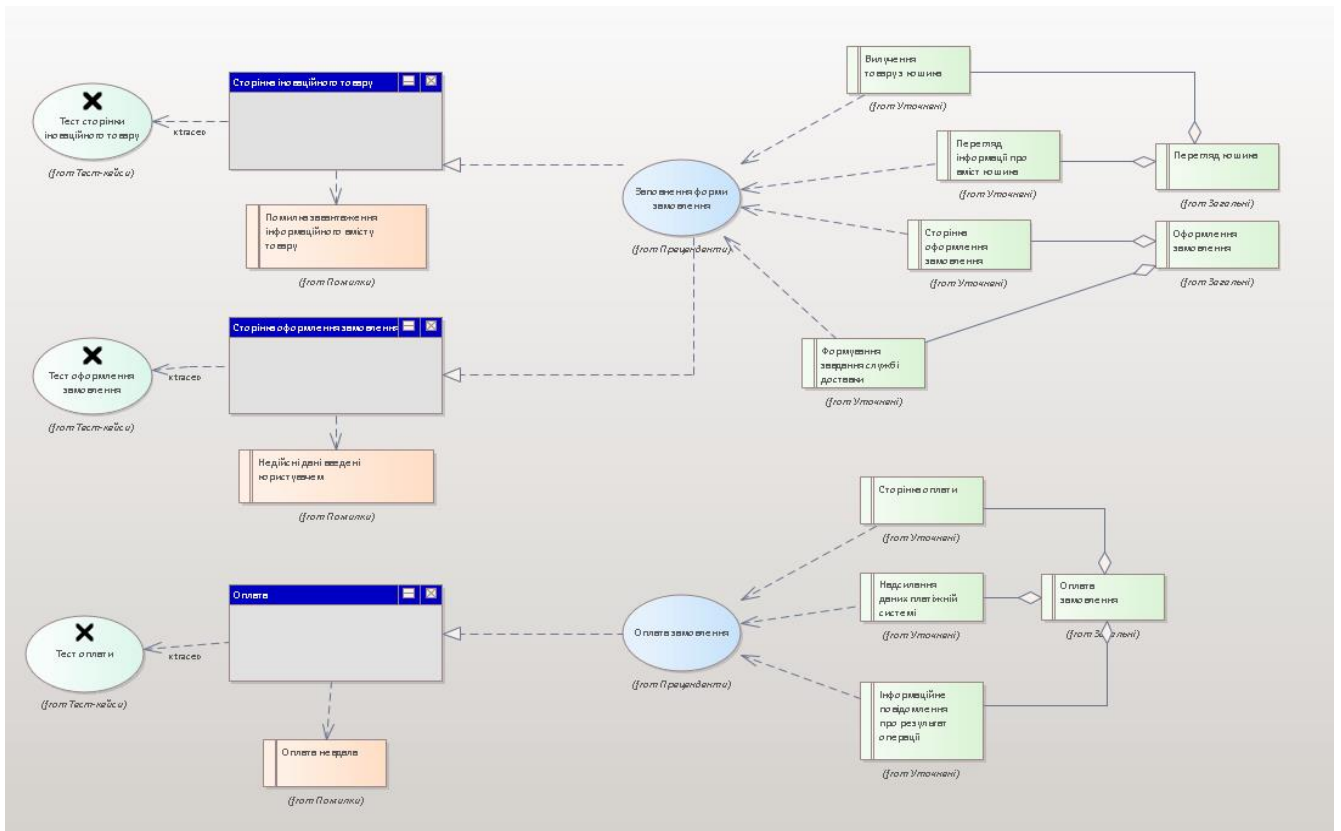


Рисунок 2.3.7 Діаграма трасування

Кожен компонент може мати свою функціональність, яка відображається через взаємодію з вимогами, прецедентами, варіантами тестування та елементами користувацького інтерфейсу. Це дозволяє краще розуміти, як різні частини системи взаємодіють між собою та виконують свої функції у контексті всього проекту. Діаграма трасування є важливим інструментом для візуалізації цих зв'язків і забезпечення їх належного відстеження протягом всього життєвого циклу розробки програмного забезпечення.

Розділ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ/ПІДСИСТЕМИ

3.1 Інформаційне забезпечення;

Загальна характеристика інформаційного забезпечення;

Інформаційне забезпечення (ІЗ) веб-застосунку для магазину інноваційних продуктів у світі домашніх тварин є на мою думку критичним компонентом, що дозволяє ефективно керувати бізнес-процесами та відповідно забезпечує надійність і головне точність роботи всієї системи. Воно охоплює організацію збору, обробки, зберігання і передачі інформації, безпосередньо необхідної для здійснення операцій і прийняття рішень. Основна мета полягає у створенні ефективної платформи для продажу інноваційних товарів, оптимізації процесів цього продажу, покращенні обслуговування клієнтів, та забезпеченні надійної доставки інноваційних товарів.

Перелік об'єктів, під час управління якими розв'язується задача:

- Товари – інноваційні продукти для домашніх тварин.
- Клієнти – власники домашніх тварин, які купують товари.
- Замовлення – процеси обробки та виконання замовлень клієнтів.
- Доставка – процеси, пов'язані з транспортуванням товарів до клієнтів.

Довідники і оперативна інформація. Для розв'язання цього комплексу задач використовуються довідники:

- Товарів – інформація про доступні інноваційні продукти, їх характеристики, ціни та наявність на складі.
- Клієнтів – ідентифікаційні дані клієнтів, історія їхніх замовлень та переваги.
- Замовлень – відомості про всі замовлення, їх статус, вміст та історія змін.
- Доставки – маршрути та графіки доставки, стан виконання доставок, трекінг посилок.

Періодичність розв'язання задачі: Підтвердження замовлень в режимі реального часу, звіти про продажі: щоденно, щотижнево, щомісячно. Маршрути та графіки доставки: щоденно.

Організація збору і передавання первинної інформації;

У даному параграфі розглянуто перелік джерел і носіїв інформації, а також загальні вимоги до організації збору і передання інформації для веб-застосунку магазину інноваційних продуктів у світі домашніх тварин.

Відділ продажів: Джерелом інформації є внутрішній підрозділ компанії, що відповідає за безпосередню обробку замовлень та взаємодію з клієнтами. Вхідними повідомленнями є інформація про замовлення клієнтів, деталі замовлень. Періодичність - постійне оновлення у реальному часі при отриманні нових замовлень. Спосібом надання інформації є електронні форми та CRM-система. Носієм інформації є електронні бази даних та внутрішня CRM-система.

Відділ закупівель: Джерелом інформації є внутрішній підрозділ, відповідальний за закупівлю товарів у постачальників. Вхідними повідомленнями є інформація про наявність товарів, нові поставки, умови поставок. Періодичність - щоденне оновлення або за необхідністю при надходженні нових партій товарів. Спосібом надання інформації є електронні таблиці, ERP-система. Носіями інформації є електронні бази даних, та ERP-система.

Відділ доставки: Джерелом інформації є внутрішній підрозділ, що координує процеси доставки. Вхідні повідомлення це інформація про відправлення, статус доставки, транспортні компанії. Періодичність: постійне оновлення у реальному часі. Спосібом надання інформації є електронні форми, системи управління доставкою (TMS). Носіями інформації є електронні бази даних та TMS.

Інтернет та зовнішні бази даних: Джерелом інформації є зовнішні постачальники даних, інтернет-ресурси. Вхідними повідомленнями є інформація про нові продукти, ціни, відгуки та рейтинги товарів. Періодичність тут щоденна, щотижнева. Спосібом надання інформації є API-запити, веб-сервіси. Носіями інформації є зовнішні бази даних, веб-сайти.

Загальні вимоги до організації збору і передання інформації; Усі вхідні повідомлення повинні надходити до системи своєчасно, без затримок, щоб забезпечити актуальність даних. Точність і надійність: Інформація повинна бути точною і достовірною. Всі джерела інформації мають відповідати за коректність поданих даних. Автоматизація збору даних: Використання автоматичних систем для збору і передання інформації з мінімальним залученням людських ресурсів. Це зменшує ймовірність помилок та підвищує ефективність процесу. Безпека і захист даних: Всі дані повинні передаватися та зберігатися з урахуванням вимог безпеки, щоб запобігти несанкціонованому доступу або втраті інформації. Регулярне оновлення: Періодичність оновлення даних повинна відповідати характеру і вимогам бізнес-процесів, що забезпечує своєчасне відображення змін у системі. Інтеграція з існуючими системами: Збір і передання інформації повинні бути інтегровані з існуючими системами компанії (CRM, ERP, TMS), що забезпечує єдність даних і полегшує управління інформацією. Організація збору і передання первинної інформації є ключовим

етапом у забезпеченні ефективної роботи інформаційної системи веб-застосунку для магазину інноваційних продуктів у світі домашніх тварин.

Побудова системи класифікації та кодування;

У даному параграфі розглянемо системи класифікації об'єктів, найменування використовуваних класифікаторів і кодів для веб-застосунку магазину інноваційних продуктів у світі домашніх тварин. Опишемо методи кодування об'єктів класифікації, структуру і довжину коду.

Системи класифікації об'єктів: Товари класифікуються за категоріями, типами, брендами та іншими характеристиками.

- Класифікатори: категорії товарів: товари для собак, товари для котів, товари для птахів, товари для гризунів, акваріумістика тощо.
- Типи товарів: їжа, іграшки, аксесуари, засоби догляду, ліки.
- Бренди: різні виробники та торгові марки.

Клієнти класифікуються за типами (роздрібні, оптові), регіонами, історією покупок.

- Класифікатори: типи клієнтів (індивідуальні, корпоративні).
- Регіони: класифікація за географічним розташуванням.
- Історія покупок: активні, нові, постійні.

Операції класифікуються за типами (продажі, повернення, обмін), статусами (завершено, в процесі).

- Класифікатори: типи операцій тобто продаж, повернення, обмін.
- Статуси операцій такі як завершено, в процесі, скасовано.

Методи кодування об'єктів класифікації;

Товари:

- Код товару складається з префікса, що позначає категорію товару, та унікального ідентифікатора.
- Структура коду: СТ-XXX-YYYY. СТ - код категорії товару (наприклад, DG для товарів для собак, СТ для товарів для котів).
- XXX - код типу товару (наприклад, FOD для їжі, TOY для іграшок).
- YYYY - унікальний ідентифікатор товару.
- Приклад коду: DG-FOD-0001 - їжа для собак, унікальний ідентифікатор 0001.

Клієнти:

- Код клієнта складається з префікса, що позначає тип клієнта, та унікального ідентифікатора.
- Структура коду: ТС-XXX-YYYY. ТС - код типу клієнта (наприклад, IN для індивідуальних клієнтів, CR для корпоративних клієнтів).
- XXX - код регіону (наприклад, 001 для Києва, 002 для Львова).

- YYYY - унікальний ідентифікатор клієнта.
- Приклад коду: IN-001-0001 - індивідуальний клієнт з Києва, унікальний ідентифікатор 0001.

Операції:

- Код операції складається з префікса, що позначає тип операції, та унікального ідентифікатора.
- Структура коду: OP-XXX-YYYY. OP - код типу операції (наприклад, SL для продажу, RT для повернення).
- XXX - код статусу операції (наприклад, CMP для завершено, PRC для в процесі).
- YYYY - унікальний ідентифікатор операції. Приклад коду: SL-CMP-0001 - продаж, завершено, унікальний ідентифікатор 0001.

Структура і довжина коду; Товари: код має довжину 10 символів. Клієнти: код має довжину 11 символів. Операції: код має довжину 11 символів.

Приклади класифікаторів. Категорії товарів: DG: Товари для собак, CT: Товари для котів, BR: Товари для птахів. Типи товарів: FOD: Їжа, TOY: Іграшки, ACC: Аксесуари. Типи клієнтів: IN: Індивідуальні, CR: Корпоративні. Регіони: 001: Київ, 002: Львів, 003: Одеса.

Проектування форм первинних документів, машинограм та відеокадрів;

У даному параграфі розглядається обґрунтування складу, змісту та форми представлення документів, машинограм і відеокадрів для веб-застосунку магазину інноваційних продуктів у світі домашніх тварин. Описуються вимоги до форм документів, їх найменування, ідентифікатори, користувачі, періодичність і терміни видачі, а також зазначаються документи, які будуть відправлені за допомогою електронної пошти.

Документ: Замовлення клієнта

- Склад: містить інформацію про клієнта, перелік замовлених товарів, кількість, ціни, загальну суму замовлення.
- Зміст: Номер замовлення, Дата замовлення, Інформація про клієнта (ім'я клієнта, адреса, контактні дані), Перелік товарів (назва, кількість, ціна), Загальна сума,
- Форма представлення: електронний документ у форматі PDF.
- Ідентифікатор: ORD-YYYYMMDD-NNNN (YYYYMMDD - дата замовлення, NNNN - унікальний номер замовлення).
- Користувачі: клієнти, менеджери з продажу.
- Періодичність: за запитом клієнта.
- Термін видачі: негайно після оформлення замовлення.

- Електронна пошта: надсилається підтвердження замовлення на електронну пошту клієнта.

Машинограма: Звіт про продажі

- Склад: містить безпосередньо агреговану інформацію про всі здійснені продажі за певний період.
- Зміст: Дата звіту. Період звіту. Перелік товарів (назва, кількість проданих одиниць, загальна сума). Загальна сума продажів за період.
- Фрма представлення: електронний документ у форматі CSV або Excel.
- Ідентифікатор: SALES-YYYYMMDD (де YYYYMMDD - дата звіту).
- Користувачі: менеджери з продажу, та адміністрація.
- Періодичність: щомісячно.
- Термін видачі: перше число кожного місяця.
- Електронна пошта: звіт надсилається на електронну пошту адміністрації.

Відеокадр, відеоогляд товару:

- Склад: містить відеоматеріал про товар, його функції та особливості.
- Зміст: Назва товару. Опис товару. Демонстрація використання товару.
- Форма представлення: відеофайл у форматі MP4.
- Ідентифікатор: VID-XXX-YYYYMMDD (де XXX - код товару, YYYYMMDD - дата створення відео).
- Користувачі: клієнти, менеджери з продажу.
- Періодичність: за потребою.
- Термін видачі: за запитом клієнта або менеджера.
- Електронна пошта: посилання на відео може бути надіслано клієнту на електронну пошту.

Вимоги до форм документів:

- Уніфікованість - усі документи мають відповідну стандартизовану форму для зручності обробки та аналізу.
- Доступність - документи повинні бути доступні в відповідному електронному вигляді для швидкого доступу та перегляду.
- Зрозумілість - структура і зміст документів повинні бути чітко зрозумілими для всіх користувачів.
- Безпека: усі документи повинні зберігатися в захищеному вигляді з мало обмеженим доступом для збереження конфіденційності даних.

Структура інформаційних масивів;

Таблиця 3.1.5 – Опис структури масиву

Найменування	Ідентифікатор у програмі	Умовне позначення у формулах	Формат	Бізнес-правила				Логічні семантичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Ідентифікатор продукту	product_id	pid	INT	Так	-	Так	ІНД	Продукти
Назва продукту	product_name	pd	VARCHAR	-	-	Так	-	Продукти
Ціна продукту	product_price	pp	INT	-	Не менше 0	Так	-	Продукти
Ідентифікатор замовлення	order_id	oid	INT	Так	-	Так	ІНД	Замовлення
Дата замовлення	order_date	od	DATETIME	-	-	Так	-	Замовлення
Статус замовлення	Order_status	ost	VARCHAR	-	'Pending', 'Shipped', 'Delivered', 'Cancelled'	Так	-	Замовлення
Ідентифікатор клієнта	customer_id	custid	INT	Так	-	Так	ІНД	Клієнти
Ім'я клієнта	Client_name	cln	VARCHAR	-	-	Так	-	Клієнти

Бізнес-правила: Первинний ключ (PK) / вторинний ключ (FK): product_id, category_id, order_id, customer_id є первинними ключами (PK). category_id в таблиці "Продукти" та customer_id в таблиці "Замовлення" є вторинними ключами (FK). Умова на значення: product_price в таблиці "Продукти" повинно бути не менше 0. Order_status в таблиці "Замовлення" обмежений списком можливих значень: 'Pending', 'Shipped', 'Delivered', 'Cancelled'. Обов'язкове поле: Поля product_id, name, price, category_id в таблиці "Продукти". Поля category_id, name в таблиці "Категорії". Поля order_id, customer_id, order_date, total_amount в таблиці "Замовлення". Поля customer_id, name, email в таблиці "Клієнти". Індексне поле: product_id, category_id, order_id, customer_id є індексними полями з умовою не допускати дублювання значень (ІНД). Логічні та семантичні зв'язки: У реляційній базі даних зв'язки між таблицями встановлюються через первинні та вторинні ключі. Для таблиць "Продукти", "Категорії", "Замовлення" та "Клієнти" зв'язки встановлюються наступним чином: Таблиця "Продукти" пов'язана з таблицею "Категорії" через поле category_id. Таблиця "Замовлення" пов'язана з таблицею "Клієнти" через поле customer_id.

Вибір СКБД;

Для реалізації інформаційної системи веб-застосунку магазину інноваційних продуктів у світі домашніх тварин було обрано Microsoft SQL Server (MSSQL). Я обрав цю СКБД завдяки її продуктивності, надійності, та безпеці. При проектуванні системи з використанням Microsoft SQL Server (MSSQL) слід враховувати наступні вимоги та обмеження:

Вимоги до апаратного забезпечення: Процесор: MSSQL потребує продуктивного процесора для обробки запитів і виконання транзакцій. Для невеликого магазину можна використовувати процесор середнього рівня, але для більшої кількості даних і користувачів може знадобитися більш потужний процесор. Оперативна пам'ять: Рекомендовано мати не менше 8 ГБ оперативної пам'яті для MSSQL Server. Збільшення обсягу оперативної пам'яті може покращити продуктивність бази даних. Дисковий простір: Важливо мати достатньо дискового простору для зберігання даних та резервних копій. Для SSD-дисків характерна краща продуктивність у порівнянні з HDD.

Ліцензування: MSSQL Server потребує ліцензування, що може включати як безкоштовні, так і платні версії. Для малого бізнесу можна розглянути варіант MSSQL Server Express, який є безкоштовним, але має обмеження на обсяг бази даних та використання апаратних ресурсів.

Даталогічна модель бази (сховища) даних.

Даталогічна модель бази даних визначає фундаментальну структуру даних, що використовується для подальшої фізичної реалізації на рівні конкретної СКБД. Для початку було створено основні таблиці, що відображають основні сутності магазину. Структура бази даних була побудована за допомогою мови опису даних (DDL). Кожна таблиця має свій унікальний ідентифікатор (PRIMARY KEY) та відповідності з іншими таблицями (FOREIGN KEY), забезпечуючи цілісність та зв'язок даних.

```
-- Створення таблиці для категорій продуктів
CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY,
    CategoryName VARCHAR(255) NOT NULL,
    Description TEXT
);
GO

-- Створення таблиці для продуктів
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(255) NOT NULL,
    CategoryID INT,
    Price DECIMAL(10,2),
    StockQuantity INT,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
GO

-- Створення таблиці для клієнтів
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100),
    Phone VARCHAR(20)
);
GO
```

Рисунок 3.1.1. Створення таблиць

```

-- Створення таблиці для замовлень
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
GO

-- Створення таблиці для деталей замовлень
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    Price DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
GO

-- Створення таблиці для корзини
CREATE TABLE ShoppingCart (
    CartID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
GO

```

Рисунок 3.1.2. Створення таблиць

Таблиці бази даних: **1 Categories (Категорії продуктів)**: Ця таблиця містить інформацію про різні категорії продуктів, такі як їжа для котів, іграшки для собак тощо. **2 Products (Продукти)**: У цій таблиці зберігається інформація про окремі продукти, включаючи їх назву, ціну, кількість на складі та категорію, до якої вони належать. **3 Customers (Клієнти)**: Таблиця клієнтів містить дані про клієнтів магазину, такі як їхнє ім'я, прізвище, електронну пошту та номер телефону. **4 Orders (Замовлення)**: Ця таблиця відображає інформацію про замовлення клієнтів, включаючи дату замовлення та загальну суму замовлення. **5 OrderDetails (Деталі замовлень)**: Таблиця містить деталі кожного замовлення, такі як продукти, кількість та ціна. **6 ShoppingCart (Корзина покупок)**: У цій таблиці зберігаються товари, які клієнти додали до своєї корзини, поки не здійснили покупку.

```

-- Додавання даних до таблиці категорій продуктів
INSERT INTO Categories (CategoryID, CategoryName, Description)
VALUES
(1, 'Їжа', 'Категорія, що містить їжу'),
(2, 'Ліки', 'Категорія, що містить ліки для тварин'),
(3, 'Розваги', 'Категорія, що містить товари для розваг');
GO

-- Додавання даних до таблиці продуктів
INSERT INTO Products (ProductID, ProductName, CategoryID, Price, StockQuantity)
VALUES
(1, 'Миска-дозатор', 1, 450, 100),
(2, 'Розумна годівничка wifi', 1, 800, 50),
(3, 'Суперприбираючий килимок', 2, 900, 30);
GO

-- Додавання даних до таблиці клієнтів
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone)
VALUES
(1, 'Іван', 'Петров', 'ivan@example.com', '123-456-7890'),
(2, 'Марія', 'Іванова', 'maria@example.com', '987-654-3210'),
(3, 'Олексій', 'Сидоров', 'oleksiy@example.com', '555-123-4567');
GO

-- Додавання даних до таблиці замовлень
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
(1, 1, '2024-05-23', 450),
(2, 2, '2024-05-24', 800),
(3, 3, '2024-05-25', 900);
GO

-- Додавання даних до таблиці деталей замовлень
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, Price)
VALUES
(1, 1, 1, 2, 450),
(2, 1, 2, 1, 800),
(3, 2, 3, 3, 900);
GO

-- Додавання даних до таблиці кошика
INSERT INTO ShoppingCart (CartID, CustomerID, ProductID, Quantity)
VALUES
(1, 1, 1, 2),
(2, 2, 2, 1),
(3, 3, 3, 3);
GO

```

Рисунок 3.1.3. Додавання даних до таблиць

Наповнення даними. Після створення таблиць вони були наповнені відповідними даними для демонстрації роботи магазину. Для кожної таблиці були додані відповідні дані, такі як категорії продуктів, окремі товари, дані клієнтів, інформація про замовлення та товари в кошику. Ці дані допомагають проілюструвати роботу системи та її взаємодію з різними типами даних, що вона обробляє.

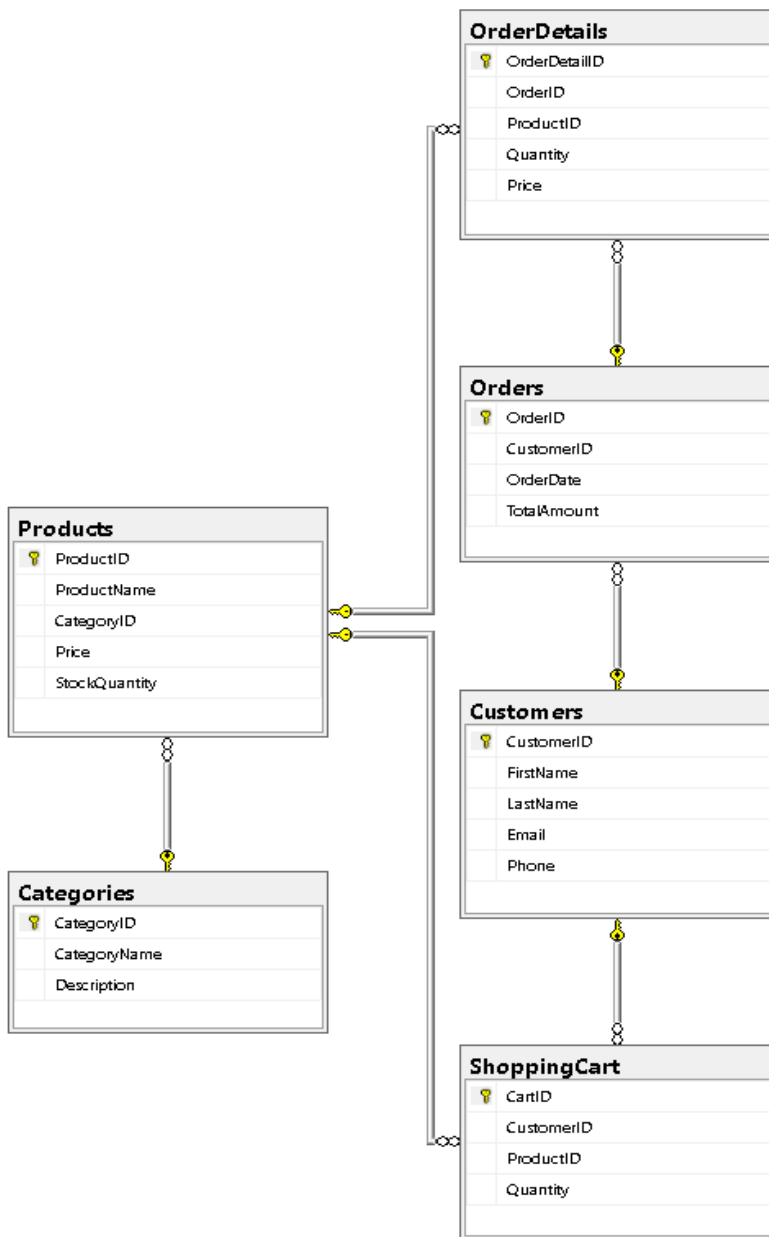


Рисунок 3.1.4. Діаграма бази даних

Діаграма бази даних це відповідне візуальне зображення структури бази даних та відображення взаємозв'язків між її складовими частинами. Ця діаграма використовується для чіткої ілюстрації сутностей, атрибутів, та відносин і ключів у базі даних, що допомагає краще зрозуміти структуру даних та її взаємозв'язки.

3.2. Технічне забезпечення

Загальні положення та схема автоматизації.

Інформаційна система магазину інноваційних продуктів для домашніх тваринро зміщена на безпосередньо віддаленому сервері. На цьому сервері будуть зберігатися база даних з інформацією про товари, клієнтів, замовлення та інші компоненти системи, а також програмне забезпечення для функціонування магазину. Клієнти матимуть доступ до системи через веб-сайт або мобільний додаток, що забезпечує доступ до Інтернету. Таким чином, система буде доступна для використання з будь-якого місця, де є Інтернет-з'єднання.

Нижче представлено схему, яка чітко описує фізичне розташування необхідного обладнання для інформаційної системи магазину інноваційних продуктів для домашніх тварин. Система включає декілька відповідних підсистем, які забезпечують роботу магазину як у фронт-офісі, так і в бек-офісі.

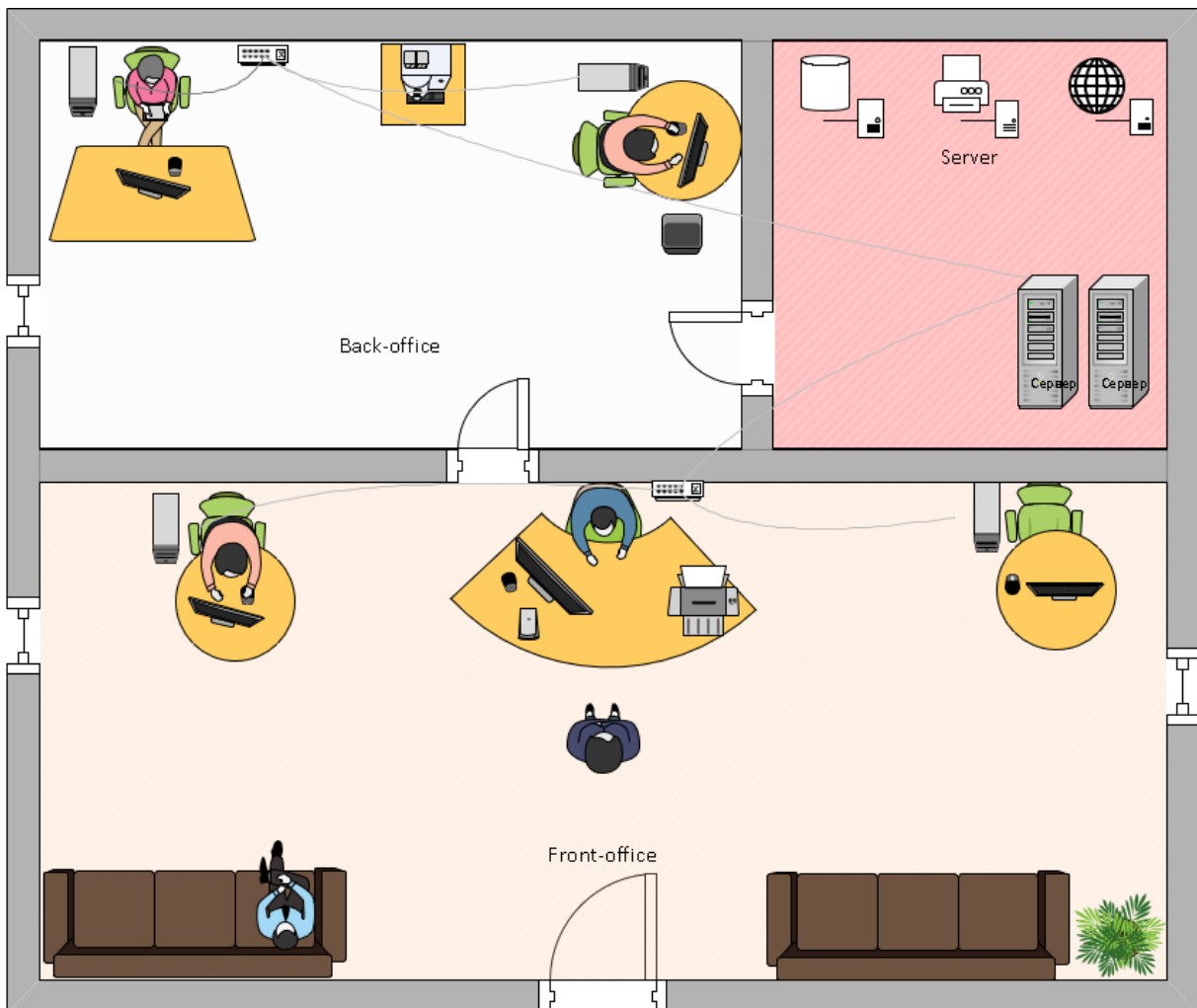


Рисунок 3.2.1. Діаграма бази даних

Опис схеми. **Front-Office** налічує: Комп'ютери для обслуговування клієнтів. Принтери для друку чеків. **Back-Office** містить: Місця з робочими станціями для керівного складу, що забезпечують прямий доступ до управлінської інформації та аналітики. Комп'ютери для безпосереднього ведення бухгалтерського обліку та фінансової звітності. **Серверна кімната** містить основний сервер, на якому розташовано базу даних, ядро системи та всі необхідні файли. Резервні сервери для забезпечення надійності та безперервності роботи системи. Мережеве обладнання: Комутатори та маршрутизатори для забезпечення надійного з'єднання між усіма компонентами системи. Друкарське обладнання: Принтери та сканери для документообігу та друку супровідних матеріалів.

На схемі також показано, як пов'язані між собою різні компоненти системи: Всі робочі станції у фронт-офісі та бек-офісі підключені до центрального сервера через комутатори. Сервер забезпечує централізоване управління базою даних та зберігання даних. Офісні робочі станції взаємодіють з сервером для оновлення інформації про запаси та обробки замовлень.

Структура комплексу технічних засобів;

Для інформаційної системи магазину інноваційних продуктів для домашніх тварин обрано структуру, яка забезпечує надійність, ефективність та безперебійний обмін даними між усіма підрозділами підприємства. Основні технічні рішення по обміну даними з іншими автоматизованими системами включають використання сучасних мережевих технологій та бездротових з'єднань для забезпечення швидкого доступу до інформації.

АРМ по відділах підприємства: Front-Office: 3 робочі станції. Back-Office: Керівництво: 1 робоча станція. Бухгалтерія: 1 робоча станція.

Захист технічних засобів; для забезпечення захисту технічних засобів від механічних, теплових та електромагнітних впливів використовуються наступні рішення: Серверне обладнання розміщується в окремому приміщенні з контрольованим доступом, оснащеному системою кондиціонування та безперебійного живлення (UPS). Засоби апаратного шифрування використовуються для захисту каналів зв'язку між сервером та робочими станціями. Доступ до серверної кімнати контролюється за допомогою ключ-карток та системи відеоспостереження.

Вибір типу ЕОМ і серверів: вибір серверного обладнання та ЕОМ базується на вимогах до продуктивності, надійності та вартості. Використовуються сервери з високою обчислювальною потужністю, що забезпечують швидку обробку транзакцій та зберігання великого обсягу даних. Робочі станції обрані з урахуванням потреб користувачів та типу виконуваних завдань.

Технічні засоби в обчислювальній мережі; до обчислювальної мережі підключені різні периферійні пристрої, які забезпечують збір, контроль та

відображення інформації: Датчики ваги та сканери. Системи відеоспостереження для моніторингу торговельного залу та складських приміщень.

Вибір периферійних технічних засобів; Для забезпечення ефективної роботи використовуються наступні периферійні пристрої: Принтери для друку чеків. Системи відеоспостереження для забезпечення безпеки.

Чисельність персоналу; Функціонування технічних засобів забезпечується наступним персоналом: Адміністратор системи: 1 особа. Технічний персонал для обслуговування обладнання: 3 особи. У випадку використання хостинг-провайдера, технічне обслуговування серверів здійснюється провайдером, що знижує потребу в додатковому персоналі на місці.

Опис автоматизованого робочого місця (АРМ);

Вибір технічних засобів АРМ. Для забезпечення коректної роботи інформаційної системи необхідно обрати відповідні технічні засоби для автоматизованих робочих місць (АРМ). Основні характеристики мінімально допустимих системних вимог для кожного типу АРМ: АРМ Працівника - Процесор: Intel Core i3 або еквівалентний, оперативна пам'ять: 4 GB, об'єм постійної пам'яті на жорсткому диску: 128 GB SSD, операційна система: Windows 10 або Linux. АРМ Адміністратора: Процесор: Intel Core i5 або еквівалентний, Оперативна пам'ять: 8 GB, Об'єм постійної пам'яті на жорсткому диску: 256 GB SSD, операційна система: Windows 10 або Linux. АРМ Бухгалтера: Процесор: Intel Core i5 або еквівалентний, Оперативна пам'ять: 8 GB, об'єм постійної пам'яті на жорсткому диску: 256 GB SSD, операційна система: Windows 10 або Linux.

Вибір периферійних технічних засобів АРМ; Для забезпечення ефективного отримання, контролю, підготовки, збору, реєстрації, передачі та відображення інформації, кожне АРМ обладнане наступними периферійними технічними засобами: АРМ Адміністратора: Монітор: 24" LED, клавіатура та миша: стандартні USB, принтер для документів, гарнітура для голосових комунікацій, веб-камера (за необхідності для відеоконференцій). АРМ Працівника: Монітор: 21.5" LED, Клавіатура та миша: стандартні USB, принтер. АРМ Бухгалтера: Монітор: 24" LED, Клавіатура та миша: стандартні USB, принтер для друку фінансової документації, скринер документів, гарнітура для голосових комунікацій.

Мережеве устаткування: Для забезпечення стабільного та безпечного з'єднання між всіма АРМ використовується наступне мережеве устаткування: Маршрутизатор з підтримкою VPN, керовані комутатори Ethernet, кабелі категорії 6 (CAT6) для підключення робочих станцій, бездротові точки доступу WiFi (за необхідності для мобільних пристроїв), апаратне шифрування каналів зв'язку для забезпечення безпеки даних. Для захисту технічних засобів

використовуються наступні рішення: Використання сейфів для мережевого обладнання, апаратне шифрування каналів зв'язку для захисту даних під час передачі, контроль доступу до приміщень з технічними засобами за допомогою ключ-карток, використання антивірусного програмного забезпечення на всіх робочих станціях.

Схема мережі передачі даних;

Для забезпечення ефективної роботи інформаційної системи обрано наступні засоби телеобробки і передачі даних: Канали зв'язку: Основний канал зв'язку: Ethernet. Резервний канал зв'язку: 4G Мобільний Інтернет. Загальна кількість мережного обладнання: Маршрутизаторів: 1 основний, 1 резервний. Комутаторів: 2. Точок доступу WiFi: 2.

Вибір мережного апаратного забезпечення: Підключення до мережі: Основне підключення: Ethernet. Резервне підключення: 4G Мобільний Інтернет. Кабельне підключення: Вита пара: UTP-8 для основних з'єднань. Оптоволокну: FTTHxxx-SM-02 для високошвидкісного підключення сервера. Маршрутизатори: Основний: Cisco RV340. Резервний: TP-Link Archer MR600 (4G LTE). Комутатори: Основний: Cisco SG350. Допоміжний: Netgear GS308. Точки доступу WiFi: TP-Link EAP225.

Вимоги до каналів зв'язку, що оренднуються: Швидкість каналу: не менше 100 Мбіт/с для основного каналу. Тип IP: статичний IP для основного каналу, динамічний IP для резервного каналу.

Розміщення абонентів і об'ємно-тимчасові характеристики передаваних даних: Бухгалтерія: Кількість користувачів: 1. Тип активності: передача документів, текстові повідомлення, Інтернет-дзвінки. Час роботи: 8 годин на добу, 5 днів на тиждень. Керівництво: Кількість користувачів: 1. Тип активності: управлінська діяльність, електронна пошта, відеоконференції. Час роботи: 8 годин на добу, 5 днів на тиждень. Front-office: Кількість користувачів: 3. Тип активності: допомога клієнтам стосовно замовлень, друк чеків. Час роботи: 12 годин на добу, 7 днів на тиждень.

Основні показники надійності, відмовостійкості, ремонтпридатності, довговічності; Надійність: Гарантійний термін використання обладнання – 3 роки. Відмовостійкість: Підтримка резервного каналу зв'язку 4G для забезпечення безперебійної роботи. Ремонтпридатність: Плановий ремонт та технічне обслуговування – кожні 6 місяців. Довговічність: Обладнання розраховане на експлуатацію не менше 5 років. Достовірність: Використання апаратного шифрування для захисту даних. Стійкість до електромагнітних завад: Всі компоненти мережі мають захист від електромагнітних завад.

3.3. Програмне забезпечення

Структура програмного забезпечення;

Програмне забезпечення включає операційну систему, середовище розробки та інструменти управління базами даних, що забезпечують робоче середовище для прикладних програм. Операційна система для серверів Windows Server 2019, яка забезпечує стабільність, високу безпеку та підтримку широкого спектру програмного забезпечення. На робочих станціях використовуються Windows 10 або macOS, залежно від потреб користувачів. Мовою програмування серверної частини є Node.js, що забезпечує високий рівень продуктивності та масштабованості. Система керування базами даних (СКБД) обирається Microsoft SQL Server 2019 для зберігання даних про клієнтів, товари, замовлення та корзини.

Прикладне або спеціалізоване програмне забезпечення: Прикладне програмне забезпечення включає веб-додаток, що має користувацький і адміністративний інтерфейси, реалізовані за допомогою сучасних веб-технологій. Веб-застосунок розробляється за допомогою React, що забезпечує інтерактивність та динамічність веб-застосунку. Для створення структури, стилізації та функціональності веб-сторінок використовуються HTML, CSS та JavaScript.

Системне програмне забезпечення;

Операційні системи є основою будь-якої інформаційної системи, забезпечуючи платформу для запуску прикладних програм та управління апаратними ресурсами. Для серверів обирається Windows Server 2019, яка забезпечує високу стабільність, безпеку та підтримку сучасних веб-технологій. Для робочих станцій обираються Windows 10. Для розробки й оформлення графічних схем, діаграм і специфікацій моделей баз даних використовуються сучасні CASE-пакети та засоби проектування. Основними інструментами є Microsoft SQL Server Management Studio (SSMS) для адміністрування та візуального моделювання баз даних. Веб-сайт створений з використанням HTML, CSS, JavaScript (React), Node.js та Microsoft SQL Server. Для аналізу неструктурованих даних та гіпертекстової інформації використовуються пакети, такі як Deductor, NeuroPro, MineSet. Ці інструменти допомагають обробляти великі обсяги даних, виконувати їхню класифікацію та візуалізацію для подальшого аналізу. Сервісні програми включають відповідні оболонки, утиліти та антивірусні засоби, які чітко забезпечують підтримку та захист комп'ютерних систем. Наприклад, антивірусне програмне забезпечення Norton або McAfee забезпечує гарний захист від шкідливого ПЗ, тоді як утиліти типу CCleaner допомагають підтримувати систему в чистоті та чудово оптимізувати її роботу. Програми технічного обслуговування, такі як тестові програми та програми

контролю, відповідно використовуються для регулярного тестування апаратного та програмного забезпечення. Це забезпечує стабільність та надійність роботи системи. Для захисту інформації використовується програмне забезпечення, таке як Open VPN для забезпечення безпечного з'єднання та шифрування переданих даних. Додаткові засоби, такі як BitLocker, використовуються для прямого шифрування даних на жорстких дисках. Для обробки зображень та графічного дизайну використовуються програми Adobe Photoshop та Adobe Illustrator. Ці інструменти дозволяють створювати та редагувати графічні матеріали для веб-сайту та інших маркетингових матеріалів. Програми обробки мовних сигналів, такі як Adobe Audition або Audacity, використовуються для редагування та обробки звукових файлів. Ці інструменти дозволяють записувати, редагувати та покращувати якість звуку. Інструментальне ПЗ включає транслятори мов програмування, компілятори, інтерпретатори та асемблери. Для розробки програмного забезпечення використовується Visual Studio Code, який підтримує численні мови програмування та має зручний інтерфейс. Node.js використовується для серверної частини веб-застосунку, що забезпечує швидке виконання JavaScript коду на сервері.

Прикладне програмне забезпечення;

Для створення та функціонування інформаційної системи необхідно розробити різноманітні прикладні програмні засоби. Ось приклади таких засобів: **ПЗ загального призначення:** Включає редактори коду та системи керування версіями. **Методо-орієнтоване ПЗ:** Використовується фреймворк React для розробки веб-інтерфейсу та Express.js для побудови серверної частини додатку на Node.js. **Проблемно-орієнтоване ПЗ:** Включає систему управління базами даних Microsoft SQL Server для зберігання та керування інформацією про продукти та клієнтів. **ПЗ для роботи глобальних мереж:** Використовується бібліотека Axios для взаємодії з сервером з використанням HTTP-запитів. **ПЗ для організації (адміністрування) обчислювального процесу:** Включає пакети pm2 для управління залежностями та скриптами проекту, а також інструменти для автоматизації розгортання, наприклад, Docker або Kubernetes.

При виборі засобів важливо врахувати їх функціональні можливості та інтегрованість з обраною системою керування базами даних. Також потрібно переконатися, що вони відповідають вимогам категорії RAD (Rapid Application Development), маючи інтегроване середовище розробки та засоби для швидкого візуального проектування компонентів.

Програмна документація;

Програмна документація для інформаційної системи включає наступні складові:

1. Формуляр: він має назву ІС веб-застосунку магазину інноваційних продуктів для домашніх тварин. Мова програмування: JavaScript (React.js, Node.js). База даних: Microsoft SQL Server. Операційна система: Windows. Тип ліцензії: МІТ. Комплектність документації: Інсталяційний пакет. Керівництва користувача та системного адміністратора. Документація API. Тестові дані та контрольні приклади.

2. Відомості про призначення програми та області використання для розв'язання завдань: Головна мета - надати користувачам зручний інтерфейс для перегляду та придбання інноваційних товарів для їх домашніх улюбленців. Області використання включають управління товарами, обробку замовлень, аналіз продажів та надання підтримки клієнтам.

3. Керівництво системного програміста: Керівництво системного програміста містить відповідні докладні вказівки та відомості для перевірки, забезпечення функціонування і налаштування програми згідно з конкретними потребами та умовами застосування. Воно охоплює такі аспекти, як інсталяція та налаштування середовища розробки, встановлення необхідних залежностей, налаштування бази даних та забезпечення безперебійної роботи веб-застосунку.

4. Обмеження на застосування мінімальної конфігурації технічних засобів: Мінімальна конфігурація серверу: Процесор Intel Core i5 або еквівалентний. Оперативна пам'ять 8 ГБ. Місце на диску 100 ГБ SSD. Операційна система Windows Server 2019. Мінімальна конфігурація клієнтського комп'ютера: Процесор Intel Core i3 або еквівалентний. Оперативна пам'ять 4 ГБ. Місце на диску 20 ГБ HDD/SSD. Операційна система Windows 10. Браузер Google Chrome, Mozilla Firefox, Microsoft Edge.

Текст програми, приклад React компонента:

```
import React from 'react';

const Product = ({ product }) => {

  return (

    <div>

      <h2>{product.name}</h2>

      <p>{product.description}</p>

      <p>Ціна: ${product.price}</p>

      <p>Кількість на складі: {product.stock}</p>
```

```
    </div>
  );
};
export default Product;
```

Приклад Node.js коду для з'єднання з базою даних MSSQL Server

```
const sql = require('mssql');
const config = {
  user: 'username',
  password: 'password',
  server: 'localhost',
  database: 'database_name'
};
sql.connect(config).then(pool => {
  return pool.request().query('SELECT * FROM Products');
}).then(result => {
  console.log(result);
}).catch(err => {
  console.error('Помилка при з'єднанні з базою даних:', err);
});
```

3.4. Результати реалізації інформаційної підсистеми

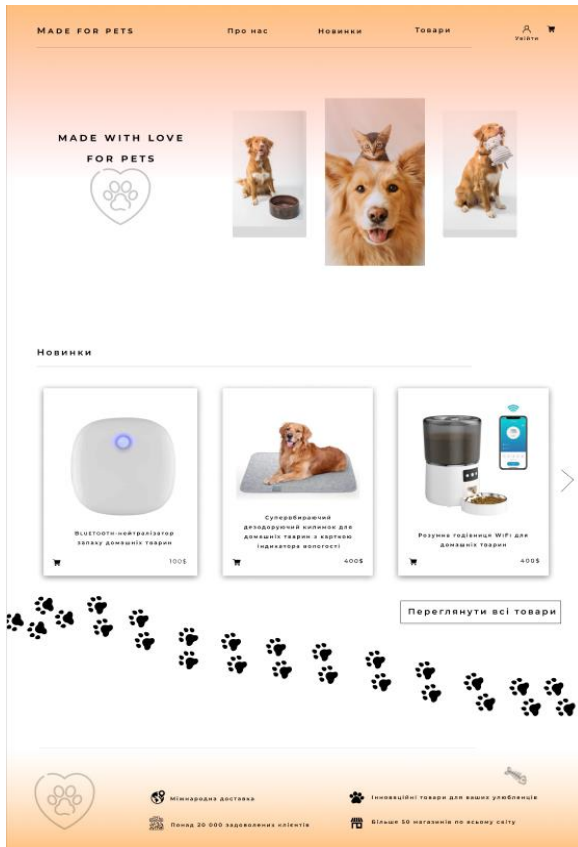


Рисунок 3.4.1 Основна сторінка сайту

На головній сторінці додатку є можливість переглянути новинки магазину, глянути весь каталог чи авторизуватися на сайті. Крім того є можливість відразу додати товар в кошик або перейти на його сторінку для більшої інформації про цей товар. Також можна прочитати коротку інформацію про магазин.

Основна сторінка має зручність використання та викликає бажання вивчати деталі та переглядати інноваційні товари. Вона створена з урахуванням потреб користувачів та надає зручний стартовий пункт для експлорації всього додатку.

Я вірю, що кожен елемент основної сторінки має свою вагомість та цінність, спрямовану на створення позитивного враження від перебування в додатку. Враховуючи каталог інноваційних товарів, зручну навігацію та вишукане оформлення, додаток готовий забезпечити чудовий досвід покупок.

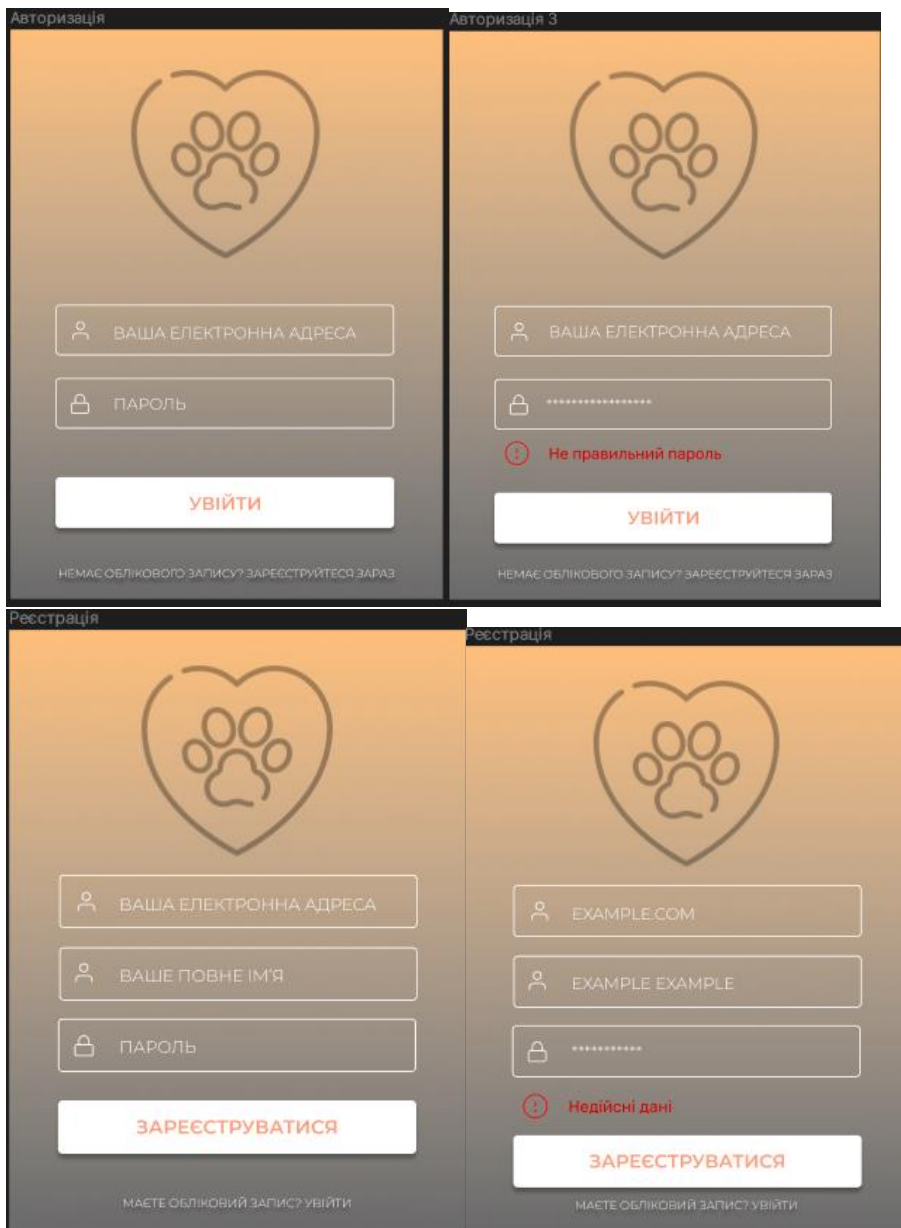


Рисунок 3.4.2 - 3.4.5 Панель авторизації та панель реєстрації

Авторизація: В цій панелі можна авторизуватися на сайт при умові коректних введених даних користувача, або у випадку якщо обліковий запис не створений, є можливість його створити. Панель авторизації визначається як ключовий фрагмент веб-магазину, призначений для спрощення вхідного процесу та забезпечення безпеки даних користувачів. Цей елемент системи розроблений із спрощеними функціональностями для комфортного користування.

Реєстрація: В цій панелі можна зареєструвати свій обліковий запис. Панель реєстрації відіграє ключову роль у взаємодії користувачів з веб-магазином, пропонуючи простий та зручний спосіб приєднатися до спільноти

користувачів. Цей інструмент розроблено з метою надання новим користувачам безпроблемного та інтуїтивно зрозумілого досвіду реєстрації.



Рисунок 3.4.6 Головна сторінка після реєстрації

Після успішної авторизації нас перекидає на основну сторінку на якій з'являється можливість увійти у свій профіль

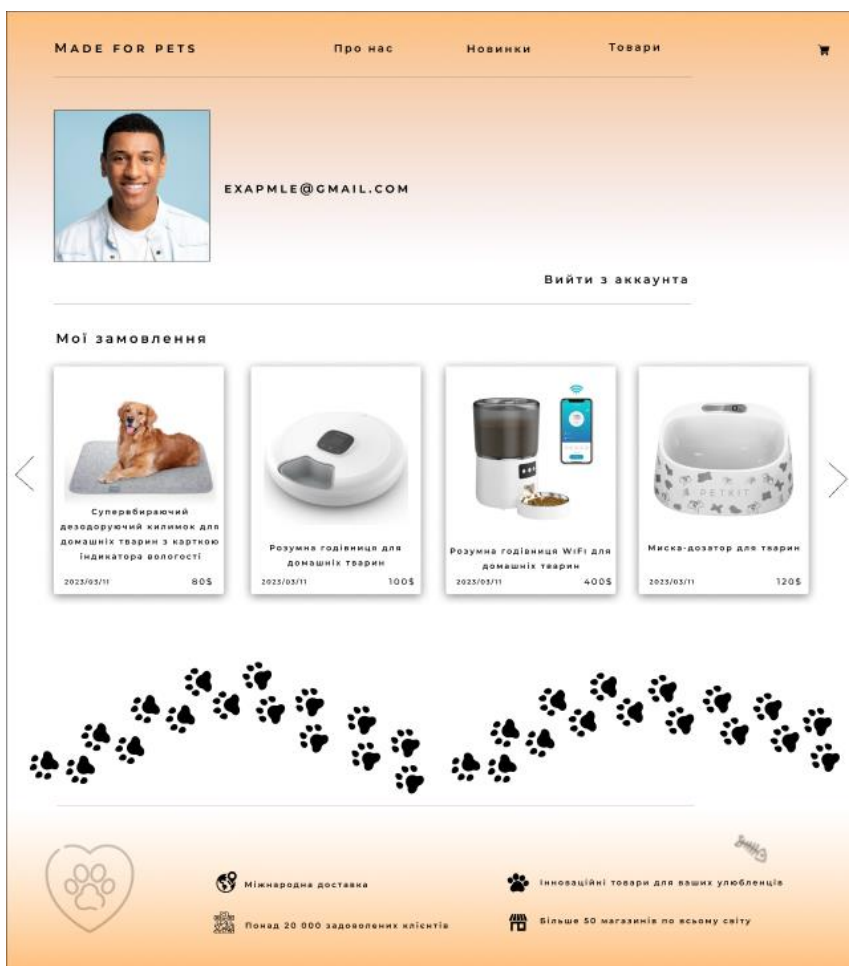


Рисунок 3.4.7 Особистий кабінет

В особистому профілі можна переглянути товари які були раніше замовлені користувачем. Це інструмент, що дає повний контроль над вашими

покупками та забезпечує зручність взаємодії з веб-магазином. Також є можливість вийти з облікового запису.

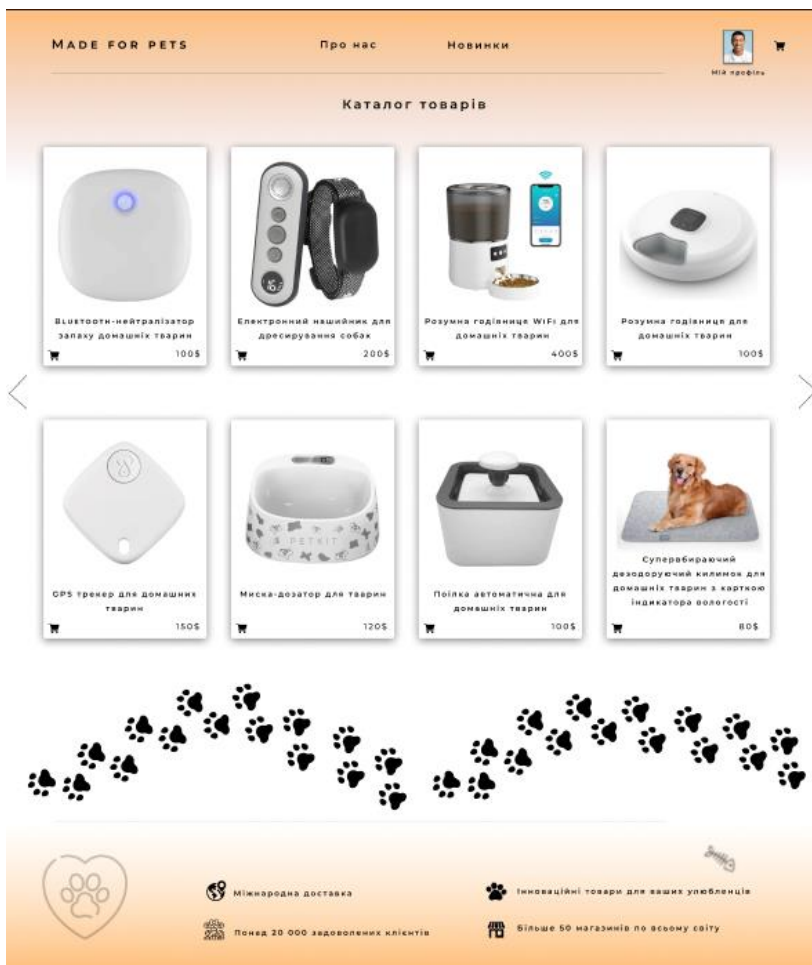


Рисунок 3.4.8 Каталог товарів

На сторінці каталогу товарів можна переглянути усі товари, замовити їх або дізнатися про них детальніше. Загальний вигляд сторінки вражає зручністю та лаконічністю, що сприяє позитивному користувацькому досвіду. Інтуїтивно зрозуміле розташування елементів та зручна навігація дозволяють користувачам легко взаємодіяти з асортиментом товарів.

MADE FOR PETS Про нас Новинки Товари  Мій профіль

Розумна автоматична годівниця для тварин WI-Fi 4 літри



400\$

Додати до кошику

Тепер вам буде дуже просто збалансувати харчування вашого улюбленця 🐾 а ще більше не доведеться стежити за часом та вагою порції 🍲

- ✓ Функція WI-Fi та зрозуміле налаштування режимів живлення за допомогою програми на смартфоні (Можливо налаштувати до 9-ти прийомів їжі в день з регулюванням часу та ваги порції)
- ✓ Запис голосу та інфрачервоний датчик блокування подачі корму, коли тварина поряд.
- ✓ Антиковзне дно буде складно зрушити з місця навіть найжазвішим улюбленцем 🐾
- ✓ Знімне металеве блюдо
- ✓ Живлення від мережі 220В або батарейок



 Міжнародна доставка

 Понад 20 000 задоволених клієнтів

 Інноваційні товари для ваших улюбленців

 Більше 50 магазинів по всьому світу

Рисунок 3.4.9 Сторінка товару

На сторінці товару викладено простий та зрозумілий дизайн, спрямований на забезпечення зручного взаємодії користувача з товаром. На сторінці можна детальніше дізнатися про цей товар та при бажанні замовити його. Також загальний вигляд сторінки товару демонструє практичність та функціональність.

Додані товари

✗		Товар	100\$
✗		Товар	80\$
✗		Товар	400\$
✗		Товар	120\$

700\$

Рисунок 3.4.10 Кошик



Рисунок 3.4.11 Кошик

Структурований та інформативний вигляд сторінки Корзини спрощує взаємодію користувача з процесом покупок. Завдяки корзині можна переглянути які товари були додані для купівлі та відповідно замовити ці товари.

Меню Оплати

ІМ'Я ВЛАСНИКА КАРТКИ

НОМЕР КАРТКИ

ТЕРМІН ПРИДАТНОСТІ

CVV

АДРЕСА ВІДДІЛУ НОВОЇ ПОЧТИ

ВАШ НОМЕР ТЕЛЕФОНУ

ОПЛАТИТИ ТА ЗАМОВИТИ

Рисунок 3.4.12 Меню оплати.

Меню Оплати

ІМ'Я ВЛАСНИКА КАРТКИ

НОМЕР КАРТКИ

ТЕРМІН ПРИДАТНОСТІ

CVV

АДРЕСА ВІДДІЛУ НОВОЇ ПОЧТИ

ВАШ НОМЕР ТЕЛЕФОНУ

! Ой ой, недійсні дані, перевірте введені деталі

ОПЛАТИТИ ТА ЗАМОВИТИ

Рисунок 3.4.13 Меню невдалої оплати.

В меню оплати можна ввести дані картки та адреси доставки та відповідно замовити товар. При недійсних даних клієнт отримає відповідне повідомлення.

ВИСНОВОК:

У цьому дипломному проекті була розроблена інформаційна система (ІС) веб-застосунку магазину інноваційних продуктів у світі домашніх тварин. Використано сучасні інформаційні технології, такі як React.js для фронтенду, Node.js для бекенду та Microsoft SQL Server для зберігання даних, щоб забезпечити ефективну роботу системи.

Розроблено зручний та інтуїтивно зрозумілий інтерфейс для користувачів магазину. Використання React.js забезпечило динамічний і швидкий відгук веб-застосунку, що покращило загальний користувацький досвід. На серверній частині використано Node.js з Express.js, що дозволило створити надійний сервер для обробки запитів користувачів та управління даними. Для зберігання інформації про продукти, замовлення та користувачів використано Microsoft SQL Server. Це забезпечило надійне і швидке зберігання даних, а також легкий доступ до них. Проект продемонстрував важливі особливості та переваги розробленої ІС, а саме інтуїтивно зрозумілий інтерфейс покращив користувацький досвід. Надійна база даних забезпечила швидкий доступ до інформації.

Однак, були виявлені деякі недоліки, а саме складність налаштування системи на локальному середовищі. Для подальшого вдосконалення системи рекомендовано провести додаткові тести на стійкість системи під високим навантаженням та додатково оптимізувати додаток для роботи з великими обсягами даних.

Розроблена ІС веб-застосунку магазину інноваційних продуктів для домашніх тварин готова до впровадження на практиці і має потенціал для подальшого розвитку та масштабування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вікіпедія – вільна енциклопедія [Електронний ресурс]. – <http://wikipedia.org>.
2. React – A JavaScript library for building user interfaces [Електронний ресурс]. – <https://reactjs.org>.
3. Node.js – JavaScript runtime built on Chrome's V8 JavaScript engine [Електронний ресурс]. – <https://nodejs.org>.
4. Microsoft SQL Server – Database Management System [Електронний ресурс]. – <https://www.microsoft.com/en-us/sql-server>.
5. Express – Fast, unopinionated, minimalist web framework for Node.js [Електронний ресурс]. – <https://expressjs.com>.
6. MDN Web Docs – HTML, CSS, and JavaScript documentation [Електронний ресурс]. – <https://developer.mozilla.org>.
7. Встановлення Node.js та npm [Електронний ресурс]. – Режим доступу до ресурсу: <https://nodejs.dev/learn/how-to-install-nodejs>.
8. Завантаження SQL Server Management Studio [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>.
9. JavaScript – Synchronous and asynchronous programming [Електронний ресурс]. – <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous>.
10. CSS-Tricks – A Complete Guide to Flexbox [Електронний ресурс]. – <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>.
11. W3Schools – SQL Tutorial [Електронний ресурс]. – <https://www.w3schools.com/sql/>.
12. GitHub – Реєстрація та створення репозиторіїв [Електронний ресурс]. – <https://github.com>.

ДОДАТКИ

Фрагменти коду:

React:

```
import React, { useState, useEffect } from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
import axios from 'axios';
```

```
// КОМПОНЕНТИ
```

```
import HomePage from './components/HomePage';
import CatalogPage from './components/CatalogPage';
import ProductPage from './components/ProductPage';
import LoginPage from './components/LoginPage';
import RegisterPage from './components/RegisterPage';
import CartPage from './components/CartPage';
import CheckoutPage from './components/CheckoutPage';
import ProfilePage from './components/ProfilePage';
```

```
const App = () => {
  const [loggedInUser, setLoggedInUser] = useState(null);
  const [products, setProducts] = useState([]);
  const [cart, setCart] = useState([]);

  useEffect(() => {
    // Отримання списку товарів під час завантаження сторінки
    fetchProducts();
  }, []);

  const fetchProducts = async () => {
    try {
      const response = await axios.get('/api/products');
      setProducts(response.data);
    } catch (error) {
      console.error('Помилка отримання списку товарів:', error);
    }
  };

  const handleLogin = async (userData) => {
    try {
      const response = await axios.post('/api/login', userData);
      setLoggedInUser(response.data);
    }
  };
};
```

```

    } catch (error) {
      console.error('Помилка входу:', error);
    }
  };

const handleRegister = async (userData) => {
  try {
    const response = await axios.post('/api/register', userData);
    console.log(response.data);
  } catch (error) {
    console.error('Помилка реєстрації:', error);
  }
};

const addToCart = (product) => {
  setCart([...cart, product]);
};

const removeFromCart = (product) => {
  const updatedCart = cart.filter((item) => item.goodsId !== product.goodsId);
  setCart(updatedCart);
};

const handleCheckout = async () => {
  try {
    const response = await axios.post('/api/checkout', { userId: loggedInUser.userId,
items: cart });
    console.log(response.data);
  } catch (error) {
    console.error('Помилка при оформленні замовлення:', error);
  }
};

return (
  <Router>
    <Switch>
      <Route exact path="/" component={HomePage} />
      <Route exact path="/catalog" render={(props) => <CatalogPage {...props}
products={products} /> />
      <Route exact path="/product/:id" render={(props) => <ProductPage {...props}
addToCart={addToCart} /> />
      <Route exact path="/login" render={(props) => <LoginPage {...props}
handleLogin={handleLogin} /> />

```

```

    <Route exact path="/register" render={(props) => <RegisterPage {...props}
handleRegister={handleRegister} /> } />
    <Route exact path="/cart" render={(props) => <CartPage {...props} cart={cart}
removeFromCart={removeFromCart} /> } />
    <Route exact path="/checkout" render={(props) => <CheckoutPage {...props}
cart={cart} handleCheckout={handleCheckout} /> } />
    <Route exact path="/profile" render={(props) => <ProfilePage {...props}
loggedInUser={loggedInUser} /> } />
  </Switch>
</Router>
);
};

export default App;

```

Nodejs:

```

const express = require('express');
const bodyParser = require('body-parser');
const sql = require('mssql');

const app = express();
const port = 3001;

app.use(bodyParser.json());

const config = {
  user: 'your_username',
  password: 'your_password',
  server: 'your_server',
  database: 'your_database',
  options: {
    enableArithAbort: true,
  },
};

// Маршрутизація для реєстрації нового користувача
app.post('/api/register', async (req, res) => {
  const { username, password, email } = req.body;
  try {
    const request = new sql.Request();
    await request.query(`INSERT INTO Users (username, password, email) VALUES

```

```

('${username}', '${password}', '${email}')`);
  res.status(200).send('Користувач успішно зареєстрований.');
```

```

} catch (error) {
  console.error('Помилка реєстрації:', error);
  res.status(500).send('Помилка сервера');
}
});

// Маршрутизація для авторизації користувача
app.post('/api/login', async (req, res) => {
  const { username, password } = req.body;
  try {
    const request = new sql.Request();
    const result = await request.query(`SELECT * FROM Users WHERE username =
    '${username}' AND password = '${password}'`);
    if (result.recordset.length > 0) {
      res.status(200).json({ userId: result.recordset[0].userId, username:
result.recordset[0].username });
    } else {
      res.status(401).send('Невірний логін або пароль.');
```

```

    }
  } catch (error) {
    console.error('Помилка входу:', error);
    res.status(500).send('Помилка сервера');
  }
});

// Маршрут для оформлення замовлення
app.post('/api/checkout', async (req, res) => {
  const { userId, items } = req.body;
  try {
    const request = new sql.Request();
    const totalPrice = items.reduce((sum, item) => sum + item.price, 0);
    await request.query(`INSERT INTO Orders (userId, totalPrice) VALUES
    (${userId}, ${totalPrice})`);
    const orderIdResult = await request.query('SELECT SCOPE_IDENTITY() AS
orderId');
    const orderId = orderIdResult.recordset[0].orderId;
    for (const item of items) {
      await request.query(`INSERT INTO OrderItems (orderId, goodsId) VALUES
    (${orderId}, ${item.goodsId})`);
    }
    res.status(200).send('Замовлення успішно оформлено.');
```

```

  } catch (error) {

```

```
    console.error('Помилка при оформленні замовлення:', error);
    res.status(500).send('Помилка сервера');
  }
});
```

```
// Маршрут для отримання списку товарів
app.get('/api/products', async (req, res) => {
  try {
    const request = new sql.Request();
    const result = await request.query('SELECT * FROM Goods');
    res.status(200).json(result.recordset);
  } catch (error) {
    console.error('Помилка отримання списку товарів:', error);
    res.status(500).send('Помилка сервера');
  }
});
```

```
// Маршрут для отримання корзини користувача
app.get('/api/cart/:userId', async (req, res) => {
  const userId = req.params.userId;
  try {
    const request = new sql.Request();
    const result = await request.query(`SELECT * FROM Cart WHERE userId =
${userId}`);
    res.status(200).json(result.recordset);
  } catch (error) {
    console.error('Помилка отримання корзини користувача:', error);
    res.status(500).send('Помилка сервера');
  }
});
```

```
// Маршрут для отримання даних користувача для особистого кабінету
app.get('/api/profile/:userId', async (req, res) => {
  const userId = req.params.userId;
  try {
    const request = new sql.Request();
    const result = await request.query(`SELECT * FROM Users WHERE userId =
${userId}`);
    res.status(200).json(result.recordset);
  } catch (error) {
    console.error('Помилка отримання даних користувача для особистого
кабінету:', error);
    res.status(500).send('Помилка сервера');
  }
});
```

```

});

app.listen(port, () => {
  console.log(`Сервер запущено на порту ${port}`);
});

sql.connect(config)
  .then(() => console.log('Підключено до бази даних'))
  .catch((err) => console.log('Помилка підключення до бази даних:', err));

```

Microsoft SQL server:

```

-- Створення таблиці для категорій продуктів
CREATE TABLE Categories (
  CategoryID INT PRIMARY KEY,
  CategoryName VARCHAR(255) NOT NULL,
  Description TEXT
);
GO

-- Створення таблиці для продуктів
CREATE TABLE Products (
  ProductID INT PRIMARY KEY,
  ProductName VARCHAR(255) NOT NULL,
  CategoryID INT,
  Price DECIMAL(10,2),
  StockQuantity INT,
  FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
GO

-- Створення таблиці для клієнтів
CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  Email VARCHAR(100),
  Phone VARCHAR(20)
);
GO

```

```
-- Створення таблиці для замовлень
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
GO
```

```
-- Створення таблиці для деталей замовлень
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    Price DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
GO
```

```
-- Створення таблиці для корзини
CREATE TABLE ShoppingCart (
    CartID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
GO
```

```
-- Додавання даних до таблиці категорій продуктів
INSERT INTO Categories (CategoryID, CategoryName, Description)
VALUES
    (1, 'Їжа', 'Категорія, що містить їжу'),
    (2, 'Ліки', 'Категорія, що містить ліки для тварин'),
    (3, 'Розваги', 'Категорія, що містить товари для розваг');
GO
```

```
-- Додавання даних до таблиці продуктів
```

```
INSERT INTO Products (ProductID, ProductName, CategoryID, Price,
StockQuantity)
VALUES
(1, 'Миска-дозатор', 1, 450, 100),
(2, 'Розумна годівничка wifi', 1, 800, 50),
(3, 'Суперприбираючий килимок', 2, 900, 30);
GO
```

-- Додавання даних до таблиці клієнтів

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone)
VALUES
(1, 'Іван', 'Петров', 'ivan@example.com', '123-456-7890'),
(2, 'Марія', 'Іванова', 'maria@example.com', '987-654-3210'),
(3, 'Олексій', 'Сидоров', 'oleksiy@example.com', '555-123-4567');
GO
```

-- Додавання даних до таблиці замовлень

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
(1, 1, '2024-05-23', 450),
(2, 2, '2024-05-24', 800),
(3, 3, '2024-05-25', 900);
GO
```

-- Додавання даних до таблиці деталей замовлень

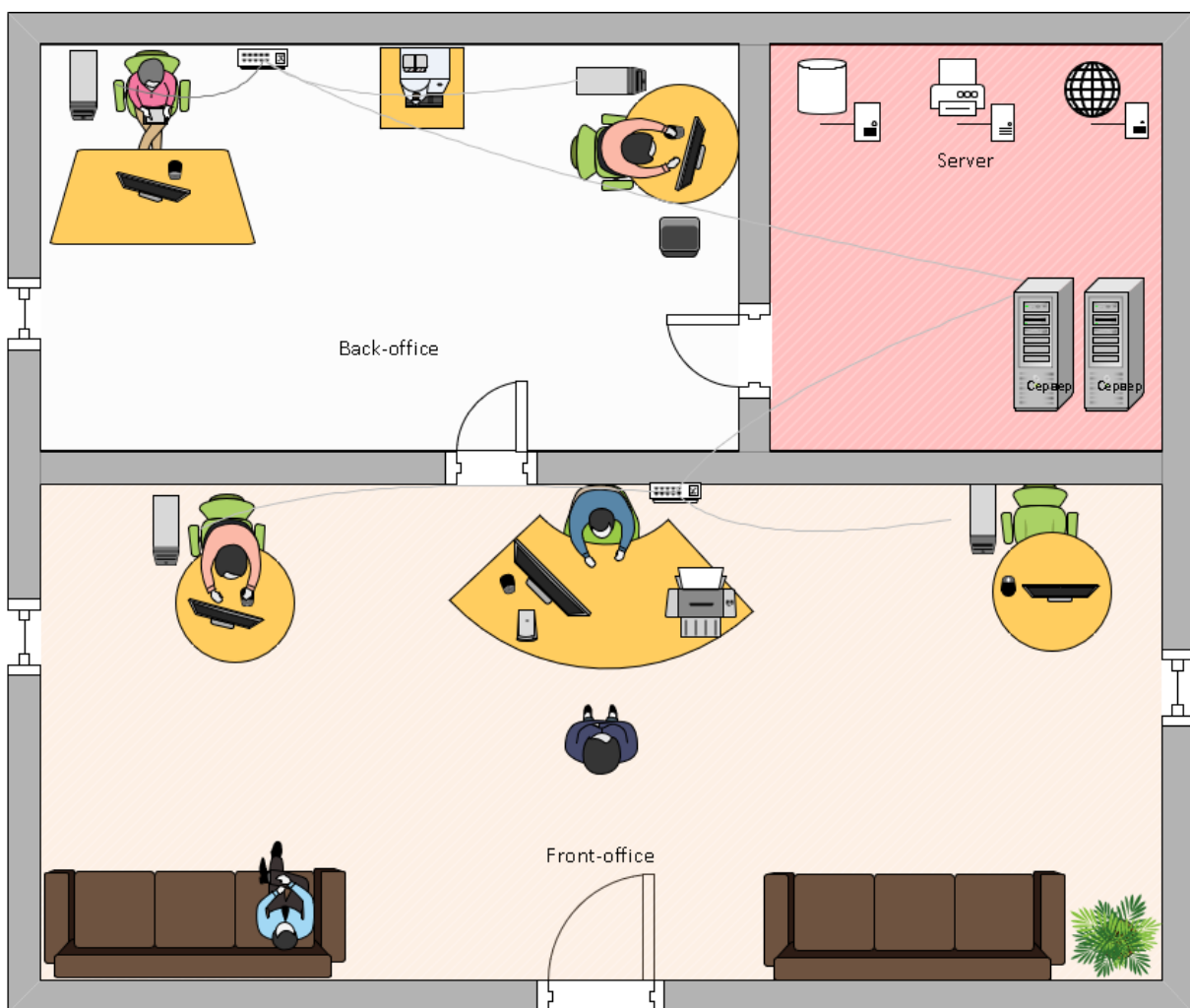
```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, Price)
VALUES
(1, 1, 1, 2, 450),
(2, 1, 2, 1, 800),
(3, 2, 3, 3, 900);
GO
```

-- Додавання даних до таблиці корзини

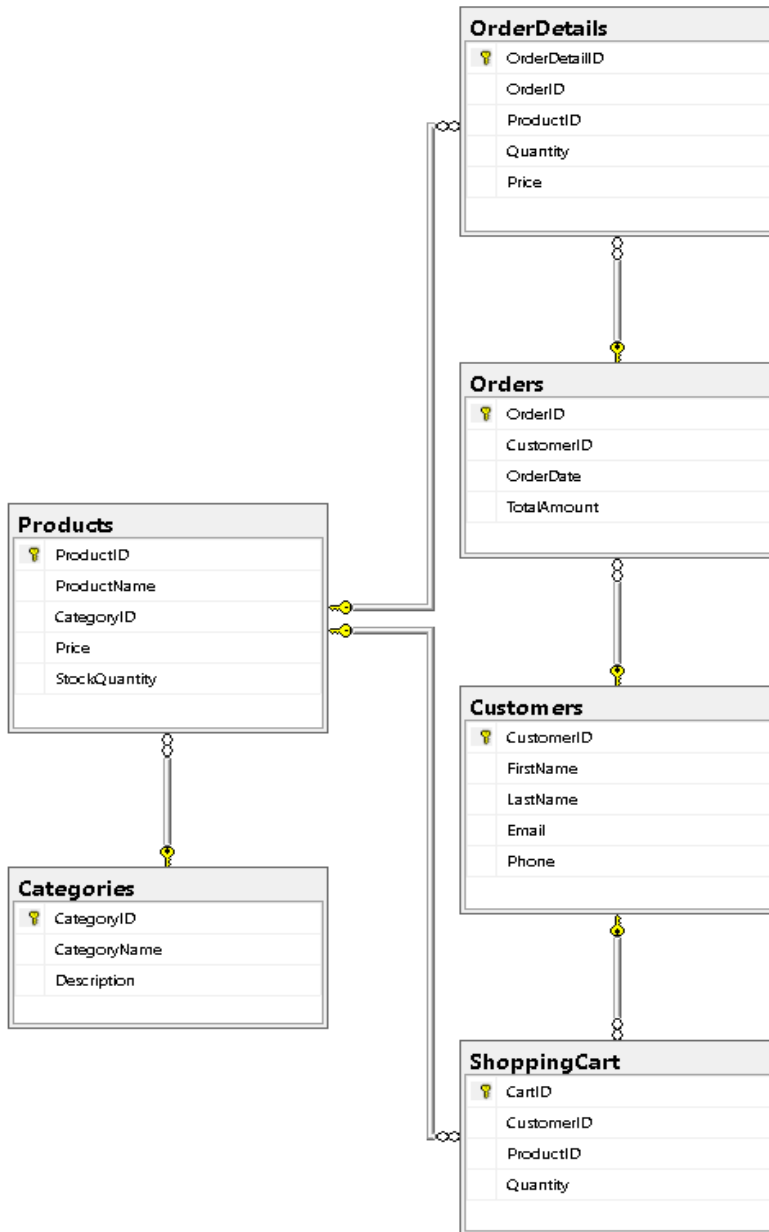
```
INSERT INTO ShoppingCart (CartID, CustomerID, ProductID, Quantity)
VALUES
(1, 1, 1, 2),
(2, 2, 2, 1),
(3, 3, 3, 3);
GO
```

ДОДАТОК Б Графічні зображення

План-схема об'єкта інформаційної діяльності



Діаграма бази даних:



ЗВІТ ПОДІБНОСТІ



Ім'я користувача: Інформаційних систем в економіці Шкуратовська Те... ID перевірки: 1016323883
Дата перевірки: 05.06.2024 15:15:30 EEST Тип перевірки: Doc vs Internet + Library
Дата звіту: 06.06.2024 08:07:31 EEST ID користувача: 100005745

Назва документа: Коломієць_Д_ІН-401

Кількість сторінок: 69 Кількість слів: 9172 Кількість символів: 79991 Розмір файлу: 2.13 MB ID файлу: 1016122437

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.86% Схожість

Найбільша схожість: 2.89% з джерелом з Бібліотеки (ID файлу: 1008077495)

3.52% Джерела з Інтернету	295	Сторінка 71
9.38% Джерела з Бібліотеки	460	Сторінка 73

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	45
Підозріле форматування	15 сторінок