

Виконала

студентка групи ПМОм-11

Кравець Ольга

Веб-магазин

Веб-сайт, який дозволить користувачеві обирати товари, згідно їх опису, переглядати їх характеристики, робити замовлення, оплачувати його та відслідковувати статус замовлення.

Вступ

Мета документу

Цей документ містить опис архітектури веб-магазину, визначає основні компоненти системи, їх взаємодію та технологічний стек для реалізації проєкту.

Область застосування проєкту

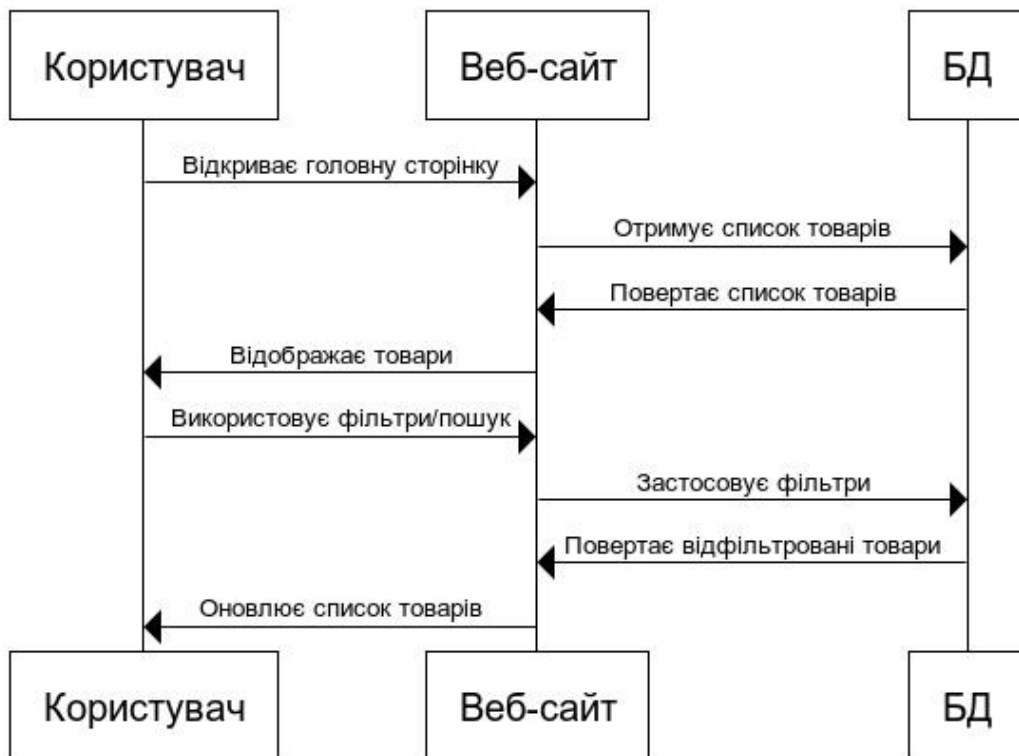
Веб-магазин дозволяє користувачам переглядати каталог товарів, оформлювати замовлення, здійснювати оплату та відстежувати статус доставки.

Основні сценарії

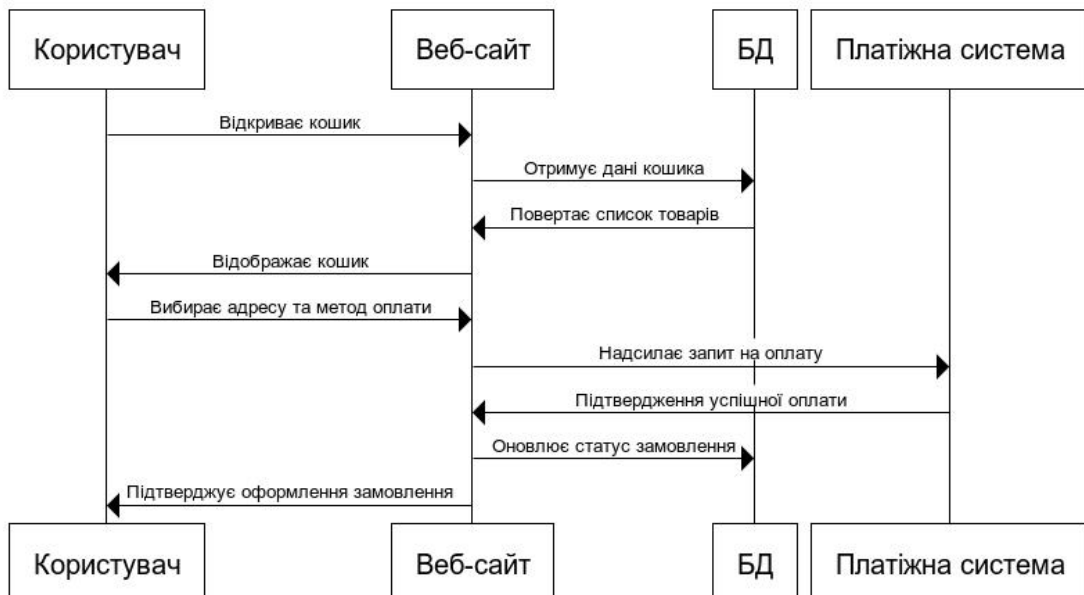
Реєстрація та авторизація користувача



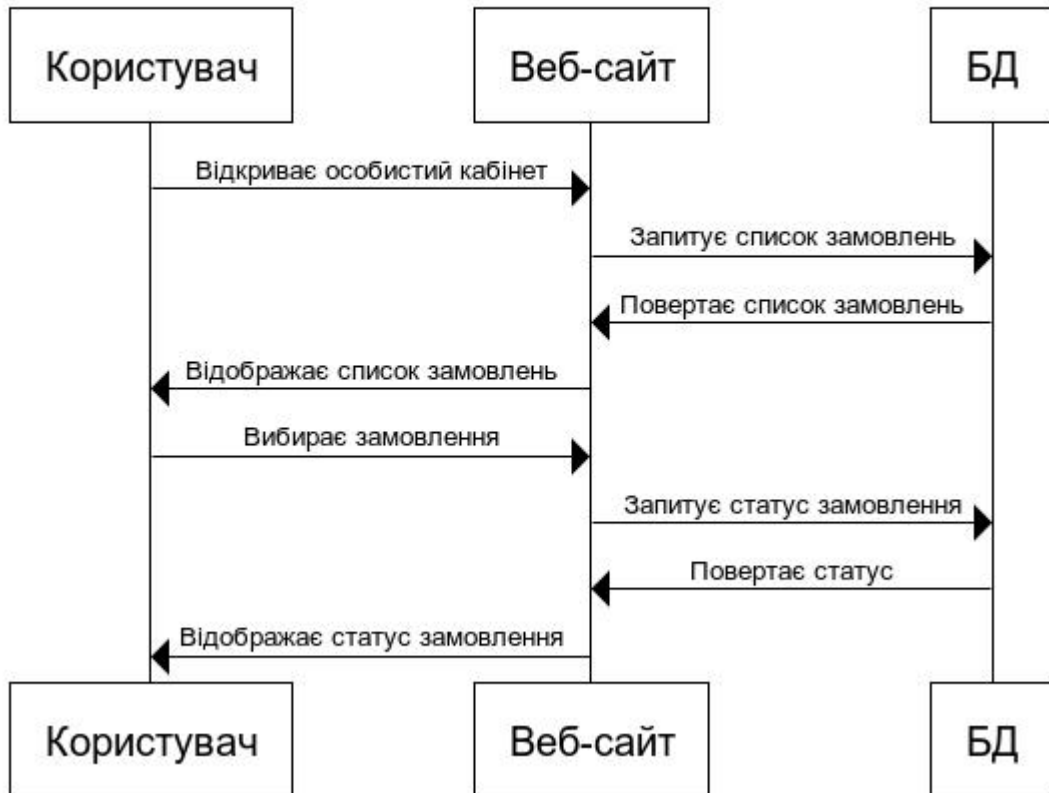
Перегляд каталогу товарів



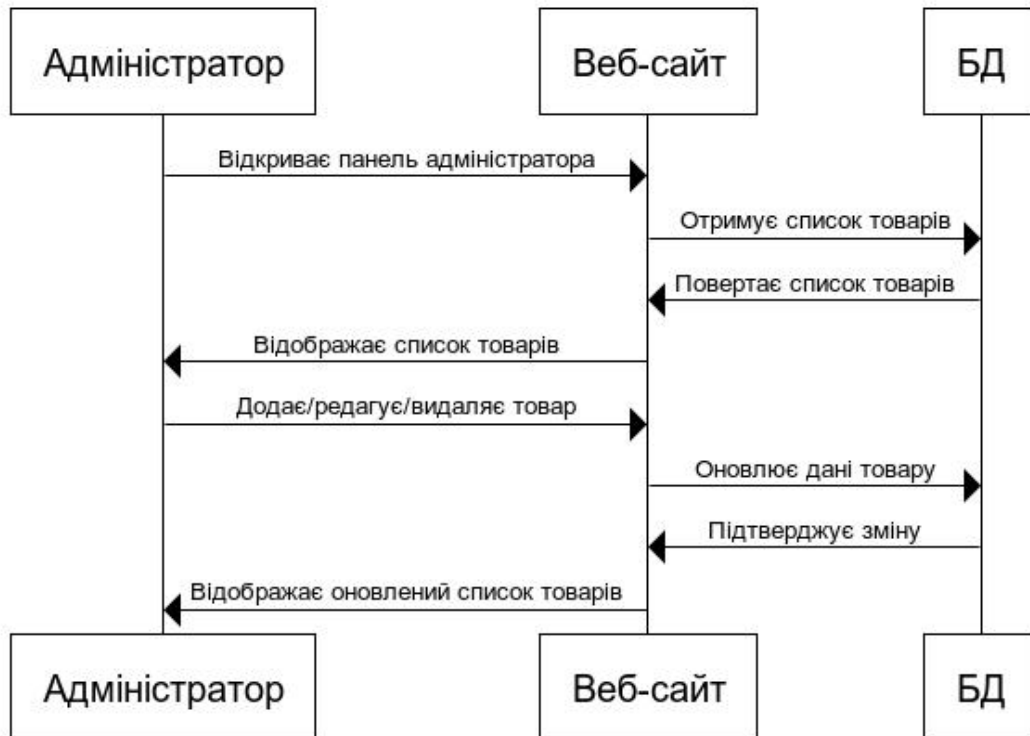
Оформлення замовлення



Відстеження статусу замовлення



Адміністрування товарів



Огляд дизайну

Архітектурний стиль

Система базується на клієнт-серверній архітектурі та використовує архітектурний патерн “N-tier Architecture” з розподілом на такі рівні:

1. Presentation Layer (UI/Клієнтська частина) – відповідає за взаємодію з користувачем.
2. Business Logic Layer (Серверна частина) – обробляє бізнес-логіку, обробку замовлень та перевірку даних.
3. Data Access Layer (Робота з БД) – відповідає за збереження та отримання даних.
4. External Services (Зовнішні сервіси) – інтеграція з платіжними системами, логістичними службами тощо.

Архітектура системи

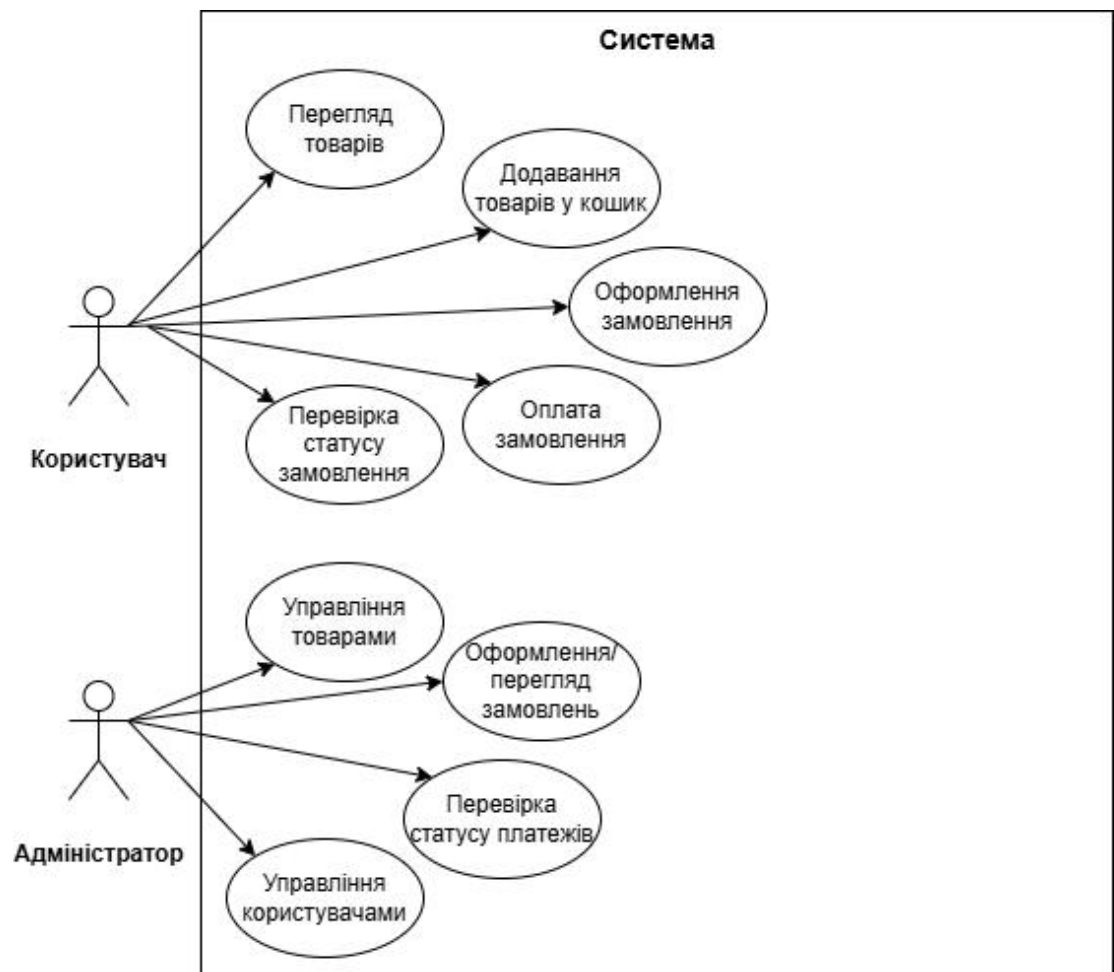
Загальний огляд

Система складається з трьох основних частин:

- Клієнтська частина (Front-end) - реалізована на Angular + Angular Material, відповідає за інтерфейс користувача. Взаємодіє з бекендом через HTTP-запити (REST API). Використовує Angular Router для маршрутизації
- Серверна частина (Back-end) - реалізована на ASP.NET Core Web API, відповідає за бізнес-логіку. Використовує MVC для обробки запитів. Реалізує авторизацію та аутентифікацію (JWT Token). Взаємодіє з базою даних через Entity Framework Core.
- База даних (Database) - MS SQL Server, використовується для зберігання інформації про товари, користувачів та замовлення. Використовує ORM Entity Framework Core для взаємодії з даними

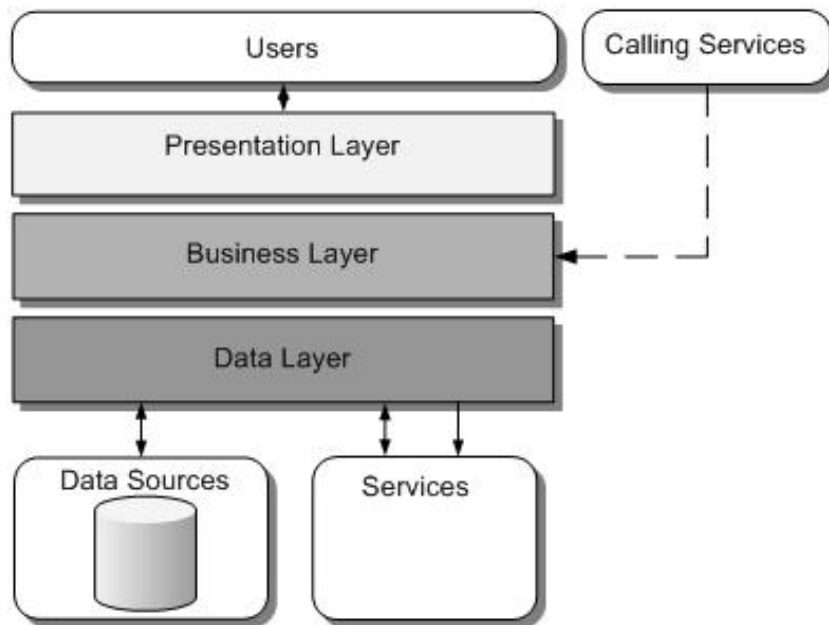
Детальний дизайн системи

Use Case Diagram



Проект складається з 4 модулів

1. Presentation Layer
2. Business Logic Layer
3. Data Access Layer
4. External Services



1. Presentation Layer (UI/Клієнтська частина)

Presentation Layer є основним рівнем взаємодії з кінцевим користувачем. Цей рівень відповідає за відображення даних користувачеві та обробку його запитів. Веб-клієнт реалізований за допомогою Angular з використанням Angular Material для забезпечення інтуїтивно зрозумілого та функціонального інтерфейсу. Користувач може переглядати товари, додавати їх до кошика, оформлювати замовлення, перевіряти статус замовлення та здійснювати оплату. Кожен запит, що надходить від користувача, обробляється контролерами, які взаємодіють з серверною частиною через REST API, отримуючи необхідні дані або повідомлення про помилки у разі невдалих запитів. Клієнтська частина також

забезпечує перевірку даних на стороні користувача перед їх відправкою на сервер.

2. Business Logic Layer (Серверна частина)

Business Logic Layer інкапсулює бізнес-логіку системи. Він обробляє запити, що надходять від Presentation Layer, виконуючи операції, пов'язані з управлінням замовленнями, перевіркою даних та їх обробкою. Наприклад, коли користувач оформляє замовлення, цей рівень перевіряє наявність товарів у кошику, застосовує знижки або промокоди, обчислює загальну суму замовлення та визначає статус. Крім того, бізнес-логіка забезпечує валідацію даних, таких як перевірка правильності введених користувачем платіжних реквізитів та адреси доставки. Сервіси, що реалізують бізнес-логіку, використовують дані з Data Access Layer для отримання або збереження інформації в базі даних. Усі сервіси розроблені так, щоб кожен із них мав конкретну функціональність і відповідав за обробку певної сутності, такої як товар, користувач чи замовлення.

3. Data Access Layer (Робота з БД)

Data Access Layer відповідає за доступ до бази даних і управління даними. Цей рівень використовує технологію Entity Framework Core, що дозволяє працювати з базою даних у стилі Code First. Спочатку визначаються моделі даних і їхні відношення, а на основі цих моделей генерується база даних і таблиці в PostgreSQL. Data Access Layer реалізує всі необхідні операції для збереження, оновлення, видалення та отримання даних з бази, при цьому абстрагується від деталей SQL-запитів, надаючи доступ до даних через об'єкти C#. Для запитів до бази даних використовується LINQ

to Entities, що дозволяє виконувати запити до бази на рівні об'єктів. Цей рівень також обробляє транзакції та забезпечує цілісність даних.

4. External Services (Зовнішні сервіси)

External Services інтегрують систему з зовнішніми сервісами, що дозволяє виконувати операції, які не можуть бути здійснені всередині системи. Наприклад, для обробки онлайн-платежів використовується інтеграція з платіжними системами, такими як Stripe чи PayPal, що дозволяє користувачам здійснювати безпечні покупки. Крім того, цей рівень обробляє інтеграцію з логістичними компаніями для відстежування статусу доставки замовлень і надання користувачам актуальної інформації про місцезнаходження їхніх посилок. External Services забезпечують також доступ до сторонніх API для перевірки наявності товарів на складі або використання зовнішніх баз даних для обробки даних про користувачів чи продукти. Взаємодія з зовнішніми системами здійснюється через HTTP-запити та API, і цей рівень обробляє помилки, що можуть виникати при відсутності з'єднання з сервісами чи інших збоїв.