**Завдання.**

- Створіть просту веб-програму Java Spring, де кінцеві користувачі можуть виконувати замовлення, які в свою чергу, оброблятимустья персоналом (наприклад: виконати замовлення, або відхилити).

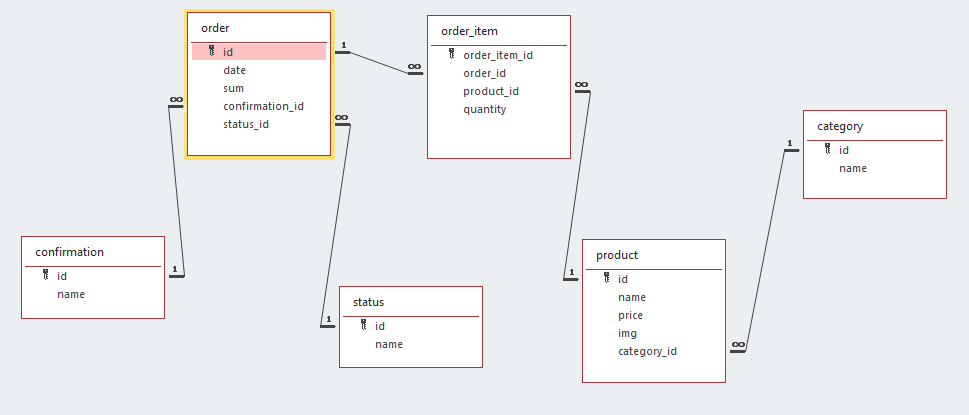
- Функція «Замовлення» має працювати у режимі реального часу (як тільки користувачі розмістили замовлення, воно повинне автоматично відобразитися у списку замовлень для персоналу)

- Покрийте принаймні кілька основних класів та/або методів юніт тестами.

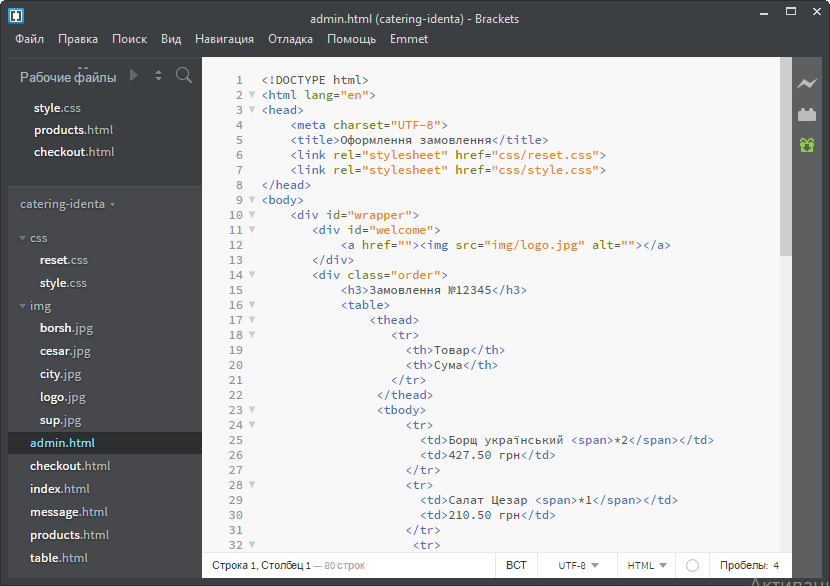
**Презентація.**

Формування цього додатка переслідувало такі кроки:

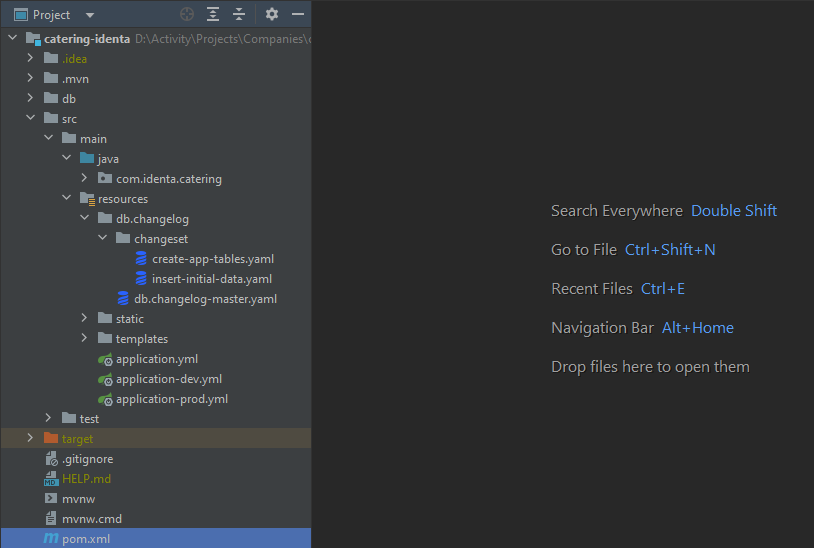
1. Аналіз завдання та формування схеми бази даних



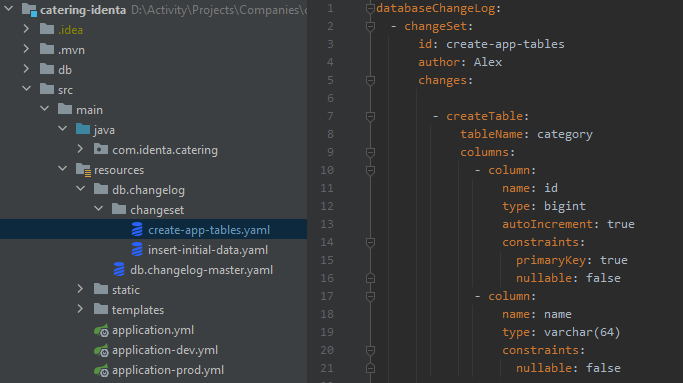
2. Верстання сайту за допомогою html, css, js в середовищі Brackets



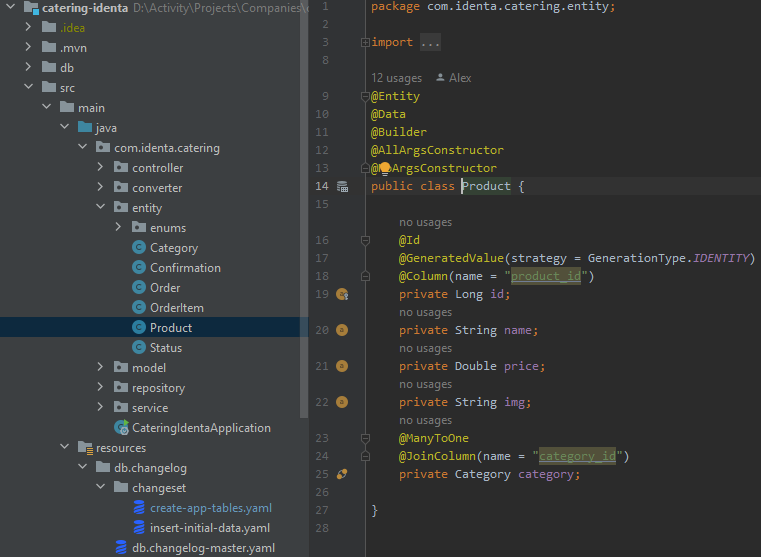
3. Налаштування додатка: Spring Boot, залежності, система міграції Liquibase та інші



4. Додавання liquibase changeset-ів зі: структурою таблиць БД та їх зв'язків, а також початкових даних. Так як використовується дві СУБД (MySQL у продакшині, H2DB для тестування), було вирішено використовувати систему міграції, яка полегшує розробку.



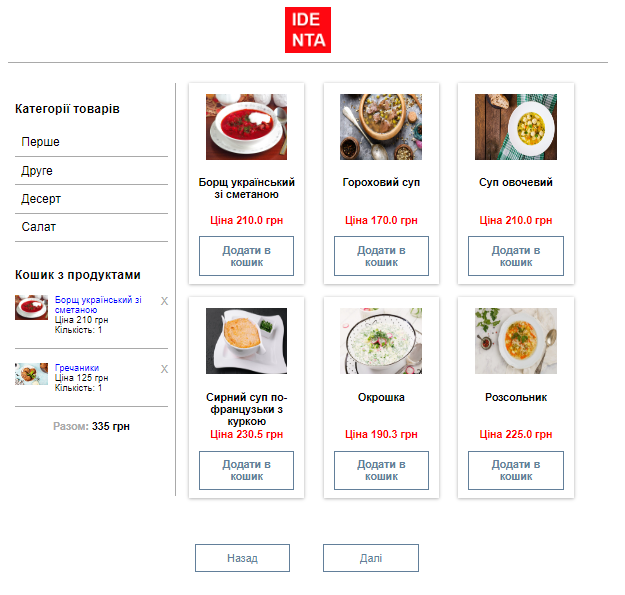
5. Формування сутностей для відображення даних з таблиць БД.



6. Покрокове додавання зверстаних раніше html,css файлів; заміна статичного коду в динамічний. Для цього були створені: контролери, які виконують логіку навігації; сервіси, які виконують бізнес-логіку; репозиторії, які дістають з БД потрібні дані. В першу чергу при запуску додатка та переході за адресою: localhost:8080, користувач потрапляє на index-сторінку, де є єдина кнопка при натискання якої користувач починає процес покупки.

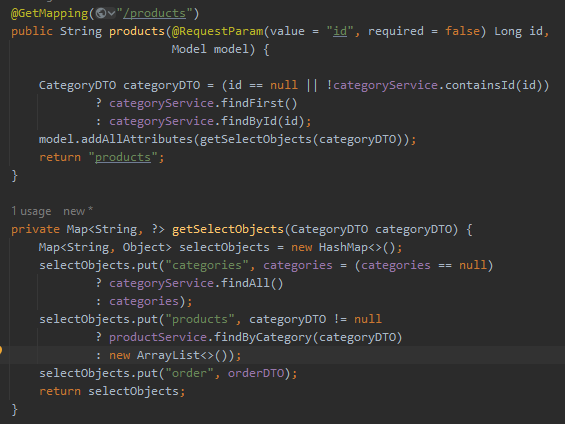


Далі користувач потрапляє на сторінку з вибором страв. Страви добавляються в кошик за допомогою кнопки "Додати в кошик" та тимчасово зберігються в кошику (зліва внизу). Товари також можно фільтрувати за категоріями нажавши на певну категорію зліва. Для того щоб побачити остаточну покупку потрібно нажати на кнопку «Далі», повернутися до головної сторінки – «Назад».

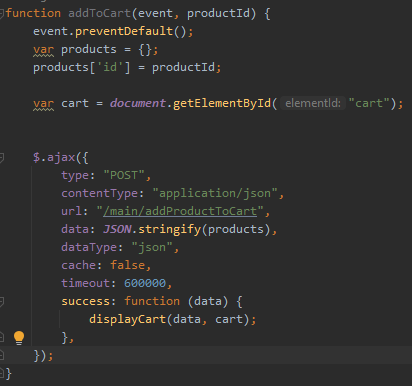


Яка логіка цієї сторінки?

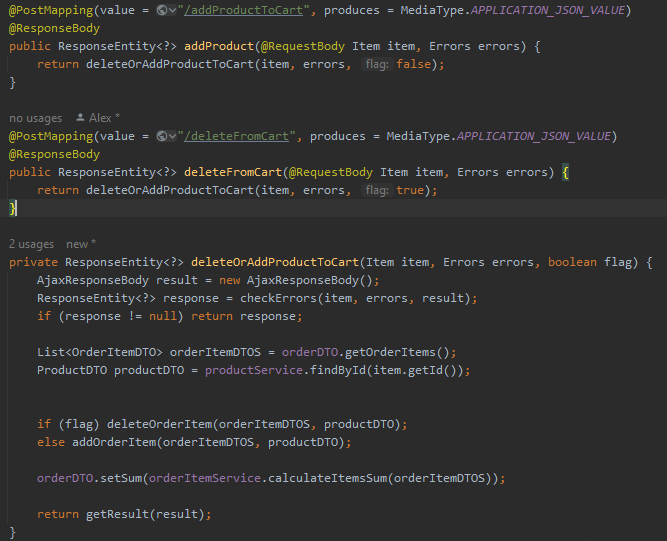
1. При виборі категорії браузер відправляє на сервер за поточним адресом, ID параметр категорії. Цей параметр приймає метод контролеру, який спочатку перевіряє валідність переданих даних, а далі повертає страви за цією категорією.



1. Додавання в кошик та видалення з кошику виконується з схожою процедурою, але тут використовується трохи інша технологія – AJAX (допомагає відправляти та отримувати дані без оновлення сторінки). Дані, у вигляді JSON, відправляються за допомогою JavaScript та бібліотеки JQuery.

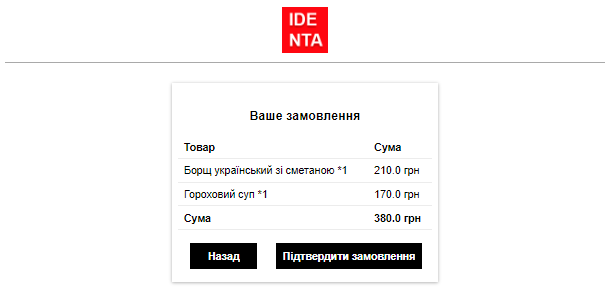


Метод контролеру перевіряє дані на валідність, потім видаляє або додає страву до замовлення. В кінці перераховує суму та повертає змінене тимчасове замовлення.



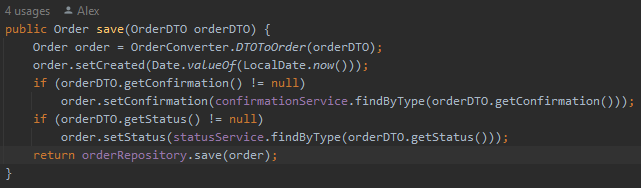
1. Сервіси в свою чергу обробляють (н/р виконують якусь логіку, перетворює сутність в модель, перевіряє чи є якась сутність в БД ) та повертають дані з БД.

Остаточні страви можна побачити на сторінці “checkout”, натиснувши на кнопку «Далі» на сторінці «products».



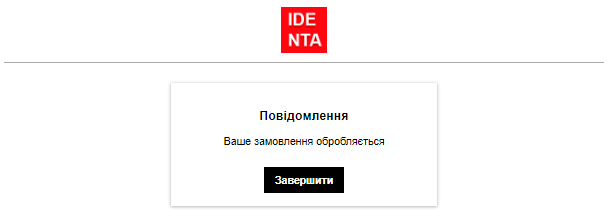
Логіка сторінки «checkout».

1. При натисканні на кнопку «Підтвердити замовлення» браузер відправляє запит контролеру, метод якого спочатку перевіряє замовлення на валідність, а далі відправляє об’єкт сервісу для зберігання, та стирає тимчасове замовлення.
2. Сервіс остаточно формує сутність типу замовлення та відправляє репозиторію для зберігання в БД.

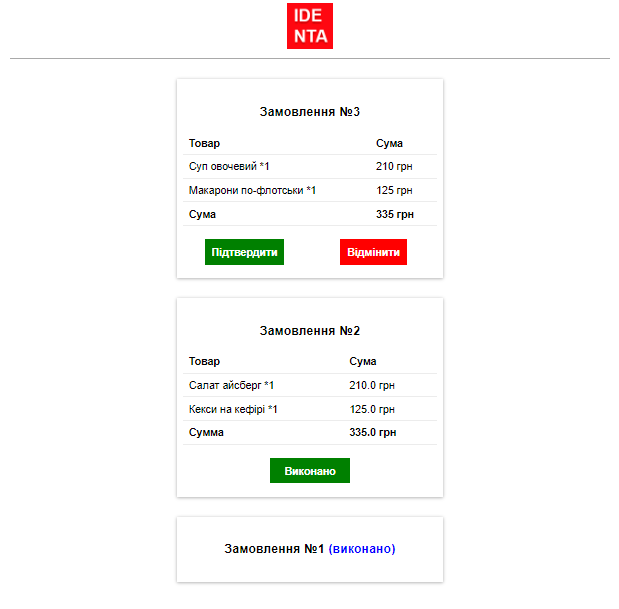


1. Контролер перенаправляє на сторінку з повідомленням

Натиснувши на кнопку «Завершити», користувач переходить на початкову сторінку.

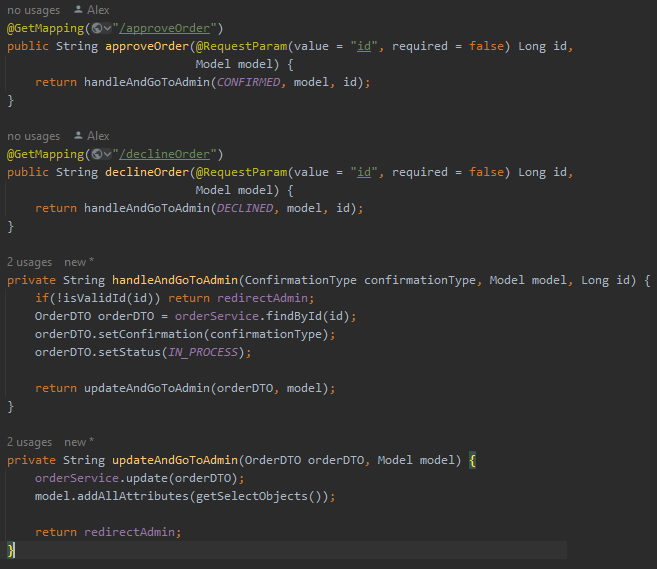


**Адмін**

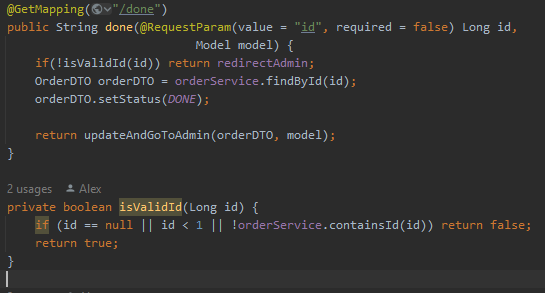
****

На сторінці *localhost:8080/admin* обробляються замовлення. Кожне замовлення має 3 стадії:

1. Перша – замовлення появляться автоматично майже одразу після того як користувач підтвердив замовлення. Автоматичне відображення відбувається завдяки технології AJAX, js-функція відправляє кожні 10 секунд запит на сервері для перевірки, чи не появилось якесь нове замовлення, якщо так – замовлення відображається на сторінці. Оператор може підтвердити замовлення або відмінити, при підтвердженні замовлення переходить до другої стадії, при скасуванні – зникає. Методи контролеру:



1. Друга – стадія виконання. Після того як замовлення виконано оператор натискає на кнопку виконано і замовлення переходить до третьої стадії. Метод контролеру:



1. Виконано – відображається тільки код замовлення, та повідомлення – виконано.

Методи сервісу де зберігається замовлення, пошук не підтверджених та не підтверджених:



Також були покриті всі сервіси unit-тестами, а також контролери – integration-тестами.