МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра інформаційних технологій

**КУРСОВА РОБОТА**

**З ДИСЦИПЛІНИ**

**«Бази даних та інформаційні системи»**

**за темою «Розробка інформаційної системи для предметної області»**

Виконав студент 3 курсу, групи 6.1229

спецiальностi 122 Комп’ютернi науки

освiтньо-професiйної програми Комп’ютернi науки

Вовк Назарiй Володимирович

Прийняв Єрмолаєв В. А.

Запоріжжя – 2021

**РЕФЕРАТ**

Курсова робота містить: 19 сторінок; 8 ілюстрацій; 2 літературних джерела.

Об'єкт дослідження − Бази данних та Системи Управління Базами Данних.

Мета роботи: вивчити можливості SQL, навчитися працювати з СУБД, проектувати бази данних згідно з предметною областю.

ЗМIСТ

[Вступ 4](#_Toc89853828)

[Етап 1 Специфікація предметної області 5](#_Toc89853829)

[Етап 2 Концептуальне проектування предметної області. Побудова ER–діаграми 8](#_Toc89853830)

[Етап 3 Перетворення ER-діаграми в реляційну схему даних 9](#_Toc89853831)

[Етап 4 Розробка SQL-скрипта для створення бази даних 10](#_Toc89853832)

[Етап 5 Розробка системи запитів. Оцінка швидкості виконання запитів. Використання індексів 11](#_Toc89853833)

[Етап 6 Забезпечення контролю за даними, що вводяться. 13](#_Toc89853834)

[Етап 7 Визначення прав доступу до бази даних 17](#_Toc89853835)

[Висновок 18](#_Toc89853836)

[Список літератури 19](#_Toc89853837)

# Вступ

Предметна область роботи – «Пекарня». Система розрахована на користування виключно співробітниками пекарні.

Обрана СУБД – SQLite. У розробці було використано такі інструменти як DB Browser for SQLite, Prisma та інструменти коммандної строки SQLite3.

Для розробки web-інтерфейсу для взаємодії з базою данних використано React, Next.js, sqlite3 на мові програмування TypeScript.

Репозиторій проекту - <https://github.com/nazarvovk/coursework-databases>

# Етап 1 Специфікація предметної області

Проаналізувати інформаційні вимоги, що надходять від замовника інформаційної системи і дати опис предметної області, що включать опис виявлених об'єктів і їхніх зв'язків, а також опис необхідних операцій для роботи з даними.

Специфіка застосування інформаційної системи - облік користувачів, продукції та замовлень.

Співробітники мають мати можливість додавати користувачів, з прив’язкою до номеру телефону. Має бути можливість додавати та видаляти вироби зі списку доступної для замовлення продукції. Основною фунцією ж є можливість створювати замовлення з фіксованою ціною та можливість зміни статусу замовлення під час роботи над ним (Замовлено -> Готується -> Очікує передачі кур'єру -> Доставляється -> Доставлено).

Передбачається, що одне замовлення може мати декілька позицій, кожна з вказаною кількістю. Ціна кожної позиції замовлення може бути змінена незалежно. Система має підраховувати вартість замовлення.

Також, замовлення може бути передвчасно скасоване. Окрім завчасно передбачених, має бути можливість встановити будь який статус.

Клієнт може мати безліч замовлень. Адреса доставки встановлюється окремо для кожного замовлення.

У даної предметної області типові операції обробки даних можуть бути такими:

1. Додавання нового клієнта
2. Створення нового замовлення
3. Зміна статусу замовлення
4. Створeння та видалення виробів
5. Отримання списку клієнтів
6. Отримання списку товарів
7. Отримання списку замовлень

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операція | Тип І–інтеракт. В–пакетн. | Частота виконання | Використовувані об'єкти | Скільки екземплярів бере участь | Тип доступу (R – читання, W – запис) |
| 1 | І | Декілька разів на день, із часом – менше. | Номер телефону | 1 | W |
| Ім’я | 1 | W |
| 2 | І | Десятки разів на день. | Клієнт | 1 | R |
| Статус | 1 | W |
| Час створення | 1 | W |
| Адреса | 1 | W |
| Вартість | 1 | W |
| Товари | Безліч | W |
| 3 | І | Сотні разів на день. | Номер Замовлення | 1 | R |
| Статус | 1 | W |
| 4 | І | 2 / місяць | Назва товару | 1 | W |
| Ціна | 1 | W |
| Опис | 1 | W |
| 5 | І | 1000/день | Клієнт | Безліч | R |
| 6 | І | 1000/день | Товар | Безліч | R |
| 7 | І | 1000/день | Замовлення | Безліч | R |

Перетворення інформаційних вимог у форму, зручну для проведення аналізу.

|  |  |  |
| --- | --- | --- |
| № | Найменування | Визначення |
| 1 | Номер телефону | Номер телефону, за яким ідентифікується клієнт |
| 2 | Ім’я | Ім’я клієнта |
| 4 | Статус | Статус опрацювання замовлення |
| 5 | Час створення | Дата коли замовлення було внесено у базу |
| 6 | Адреса | Адреса доставки замовлення |
| 7 | Вартість | Підрахована вартість замовлення. |
| 8 | Товари | Список товарів у замовленні, з кількістю і індивідуальною ціною |
| 9 | Номер Замовлення | Ідентифікатор, який присвоєно замовленню |
| 11 | Назва товару | Назва товару |
| 12 | Ціна | Ціна за одну одиницю товару за замовчуванням |
| 13 | Опис | Опис товару |

Вхідні данні:

1. Назва, кількість та ціна товару
2. Ім’я та номер телефону клієнта
3. Адреса доставки

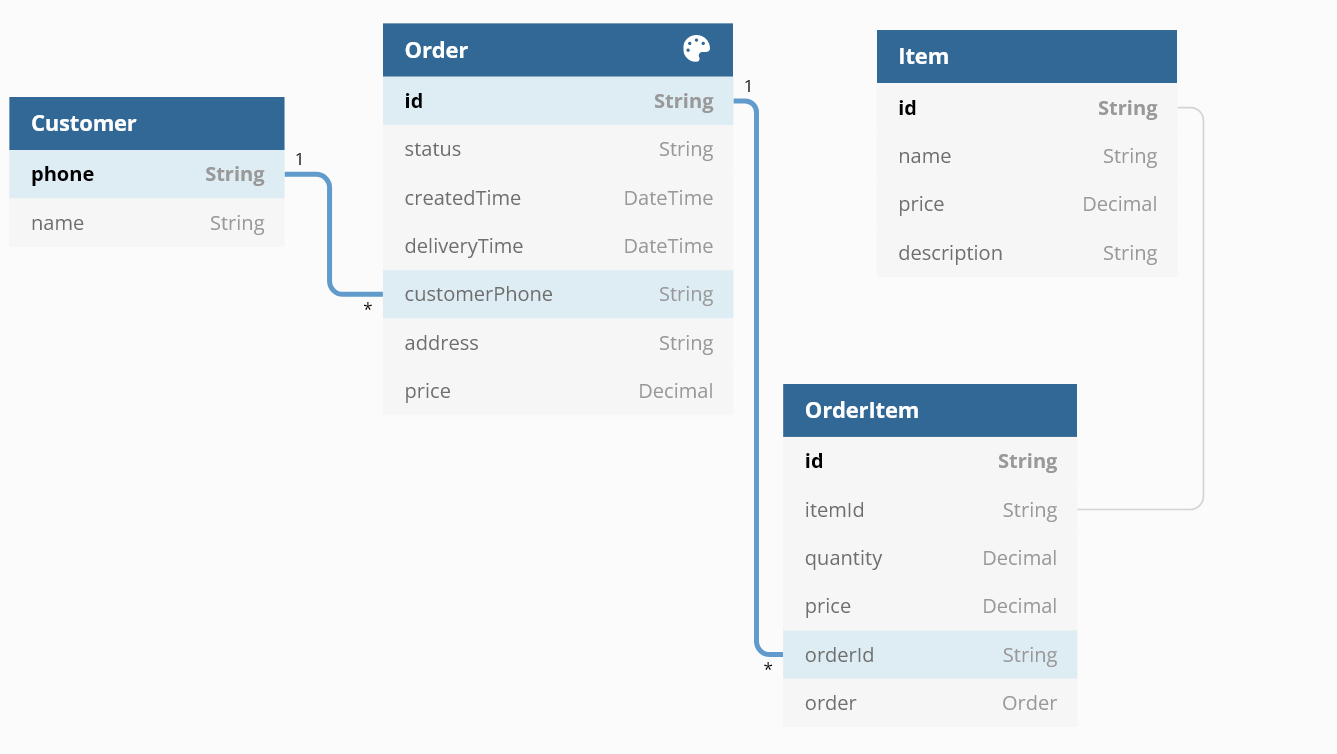
Вихідні данні:

1. Перелік замовленої продукції
2. Вартість замовлення
3. Адреса доставки
4. Статус замовлення
5. Дата та час доставки, якщо замовлення виконано

# Етап 2 Концептуальне проектування предметної області. Побудова ER–діаграми

Познайомитися з методикою ER–діаграм для концептуального проектування баз даних, навчитися будувати коректні ER–діаграми.

ER-діаграма:



Ілюстрація 1

# Етап 3 Перетворення ER-діаграми в реляційну схему даних

Використайте метод перетворення ER-діаграм в реляційні схеми і створіть реляційну схему бази даних Вашої ІС. Оцінюється коректність реляційної схеми, її відповідність до концептуальної схеми, а також до предметної області.

Customer(phone, name)

Order(id, status, createdTime, deliveryTime, customerPhone, address, price)

OrderItem(id, itemId, quantity, price, orderId)

Item(id, name, price, description)

Дана схема знаходиться у 3НФ тому, що всі залежності є залежностями від первинних ключів відповідних таблиць.

# Етап 4 Розробка SQL-скрипта для створення бази даних

Розробити SQL-скрипт у прив’язці до конкретної СУБД, заповнити значеннями всі створені таблиці в кількості достатньої для перевірки ефективності створеної структури даних.

Для створення бази данних було виконано наступні SQL-запити:

CREATE TABLE "Customer" (

"phone" TEXT NOT NULL PRIMARY KEY,

"name" TEXT NOT NULL DEFAULT '---'

);

CREATE TABLE "Order" (

"id" TEXT NOT NULL PRIMARY KEY,

"status" TEXT NOT NULL,

"createdTime" DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

"deliveryTime" DATETIME,

"customerPhone" TEXT NOT NULL,

"address" TEXT NOT NULL,

"price" DECIMAL NOT NULL,

CONSTRAINT "Order\_customerPhone\_fkey" FOREIGN KEY ("customerPhone") REFERENCES "Customer" ("phone") ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE "OrderItem" (

"id" TEXT NOT NULL PRIMARY KEY,

"itemId" TEXT NOT NULL,

"quantity" DECIMAL NOT NULL,

"price" DECIMAL NOT NULL,

"orderId" TEXT NOT NULL,

CONSTRAINT "OrderItem\_itemId\_fkey" FOREIGN KEY ("itemId") REFERENCES "Item" ("id") ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT "OrderItem\_orderId\_fkey" FOREIGN KEY ("orderId") REFERENCES "Order" ("id") ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE "Item" (

"id" TEXT NOT NULL PRIMARY KEY,

"name" TEXT NOT NULL,

"price" DECIMAL NOT NULL,

"description" TEXT

);

# Етап 5 Розробка системи запитів. Оцінка швидкості виконання запитів. Використання індексів

Розробити систему SQL-запитів для аналізу та маніпулювання даними, в якій би виконувалися всі основні операції над даними, що були визначені на етапі аналізу предметної області.

1. Розробіть запити на вибірку ( використовуючи речення SELECT-FROM-WHERE), що стосуються основних операції над даними або задач управління, які були визначені на етапі 1, але не менш п’яти запитів
   1. SELECT \* FROM "Customer";
   2. SELECT \* FROM "Item" WHERE price > 5;
   3. SELECT \* FROM "Order" o join "Customer" c on c.phone=o.customerPhone;
   4. SELECT \*, count(i.id) FROM "Order" o JOIN "OrderItem" i on i.orderId=o.id;
   5. SELECT phone, address FROM "Customer" c JOIN "Order" o ON o.customerPhone=c.phone GROUP BY id, address;
2. Розробіть запити на модифікацію даних ( використовуючи речення   
   INSERT, UPDATE, DELETE), що стосуються основних операції над   
   даними або задач управління, які були визначені на етапі 1
   1. insert into "Customer" (

name,

phone

)

values (?, ?);

* 1. insert into "Item" (id,name,price,description)

values (?, ?, ?, ?) on conflict do nothing;

* 1. update "Order" set status='${body.status}' where id=${body.orderId};
  2. update "Order" set deliveryTime=${Date.now()} where id=${body.orderId};
  3. delete from "Order" where id=?

1. Створіть два представлення для бази даних (використовуючи речення CREATE VIEW). Створіть один чи більше запитів, що звертаються до представлень.
   1. Унікальні адреси клієнтів:

CREATE VIEW IF NOT EXISTS UniqueAdresses

AS

SELECT phone, address FROM "Customer" c

JOIN "Order" o ON o.customerPhone=c.phone

GROUP BY id, address;

* 1. CREATE VIEW IF NOT EXISTS CountCustomers AS

SELECT count(\*) FROM "Customer";

SELECT \* FROM “UniqueAdresses”

1. Створіть декілька корисних індексів, щоб індексувати поля, що найчастіше з’являються в запитах
   1. CREATE INDEX OrderAddress ON “Order”(address)
   2. CREATE INDEX OrderItemsByItemId ON “OrderItem”(itemId)
   3. CREATE INDEX OrderItemsByOrderId ON “OrderItem”(orderId)

# Етап 6 Забезпечення контролю за даними, що вводяться.

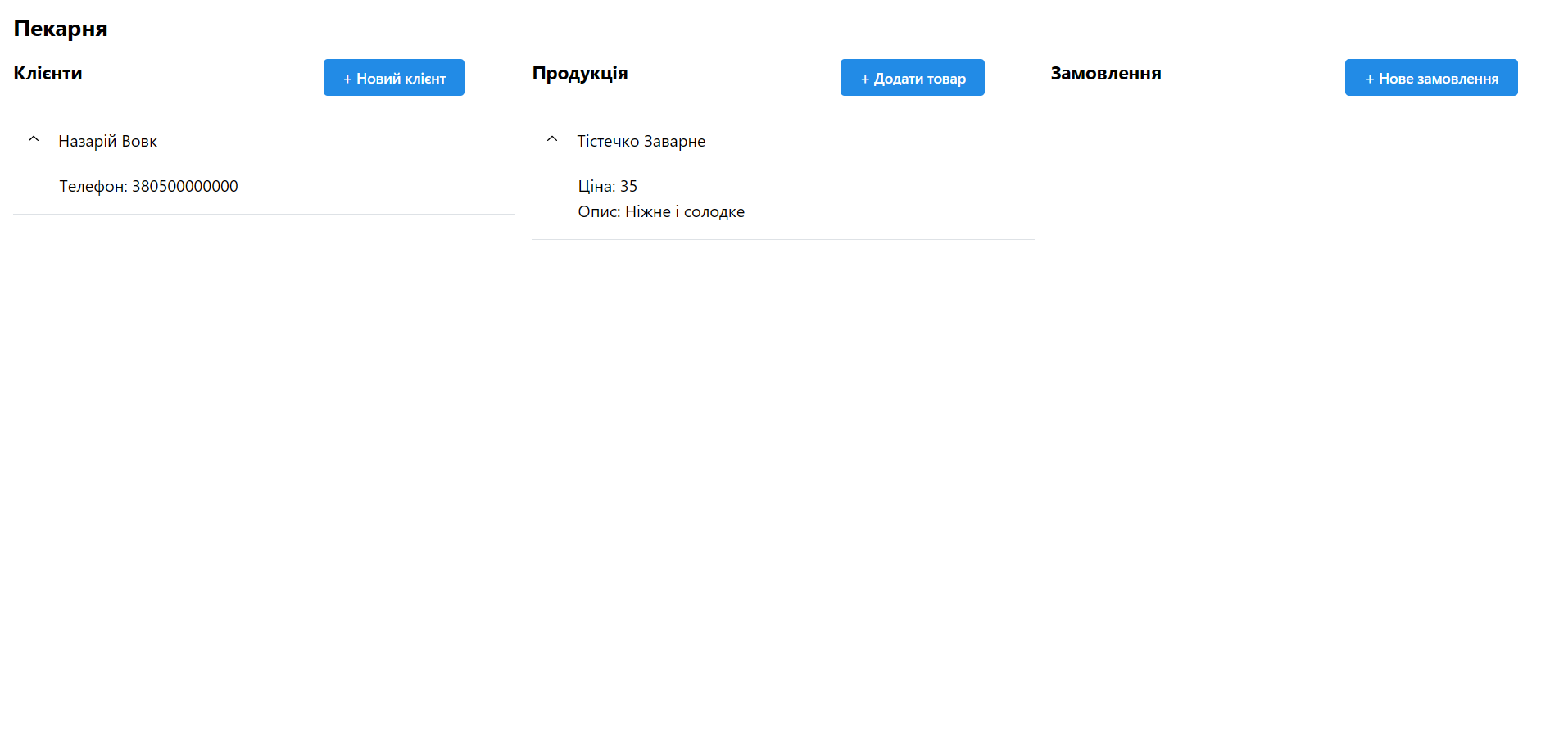
Запровадити в інформаційній системі контроль за введенням/зміною даних для забезпечення цілісності бази даних.

Для обробки і контролю данних було обрано розробити web-інтерфейс за допомогою React, Next.js, sqlite3 та інших бібліотек на мові програмування TypeScript.

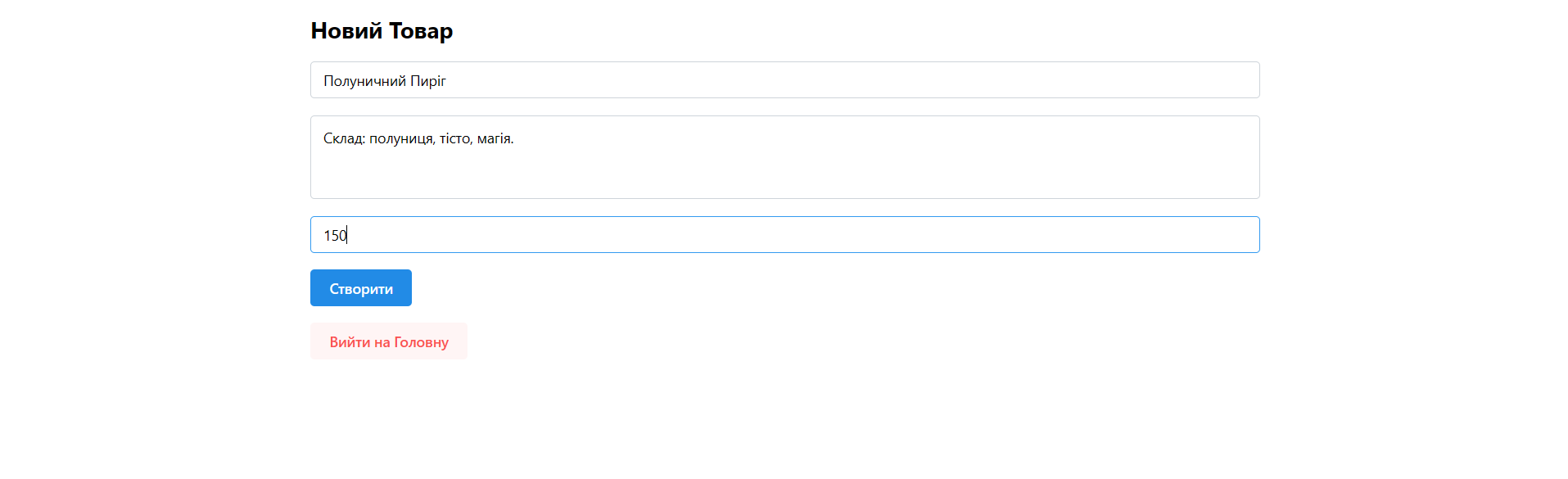
Репозиторій проекту - <https://github.com/nazarvovk/coursework-databases>

Інструкції по запуску у файлі README.md

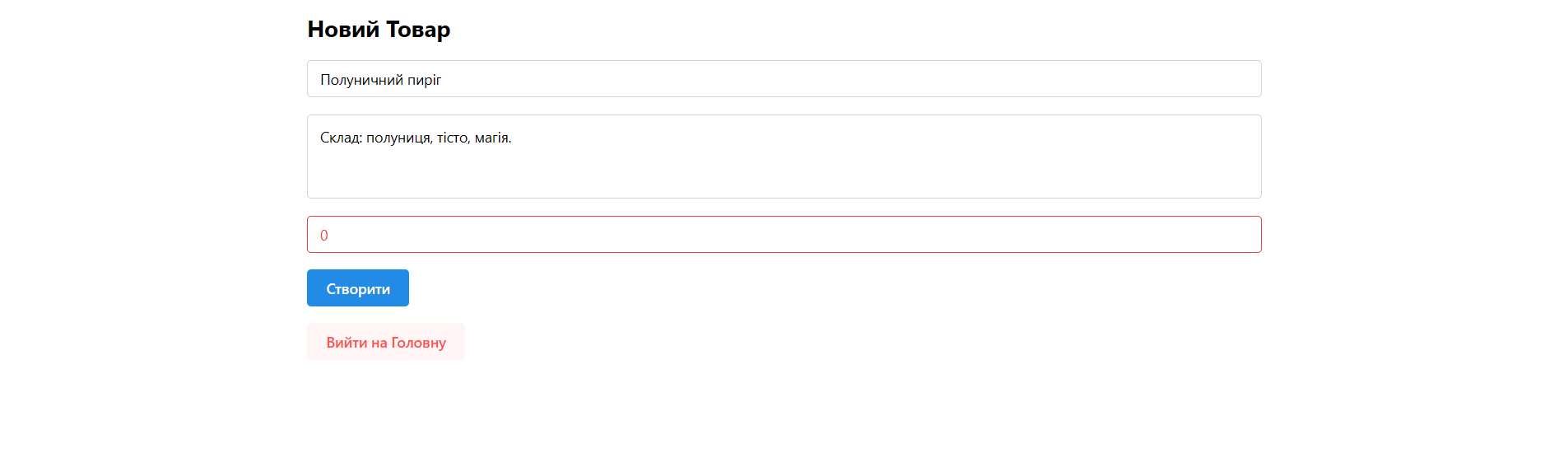
Демонстрація роботи:



Ілюстрація 2

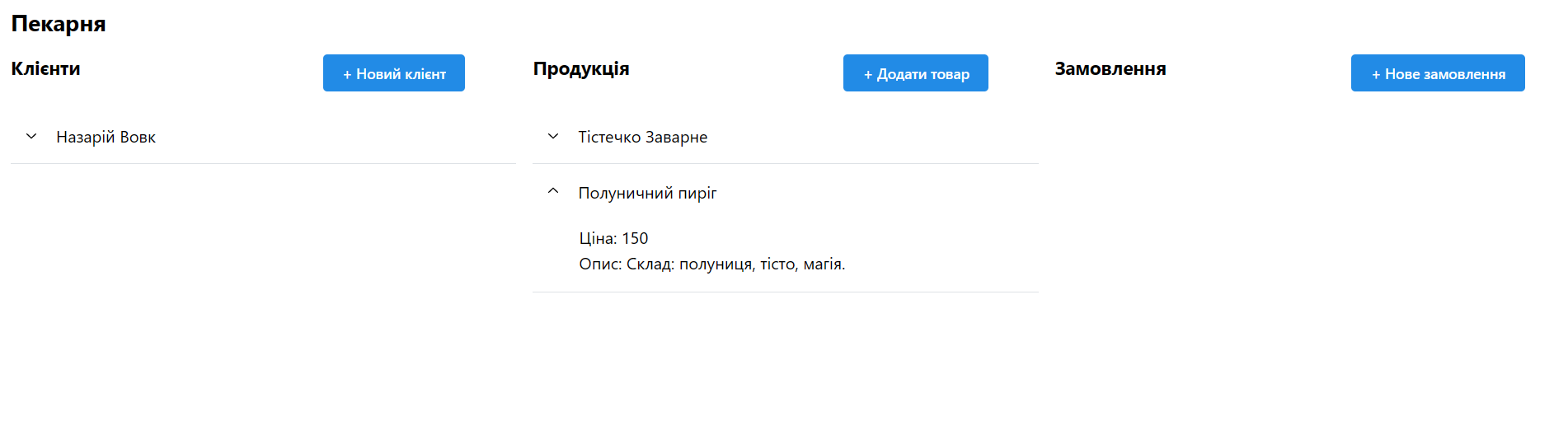


Ілюстрація 3

Форми валідують значення і не пропускають невірні данні:

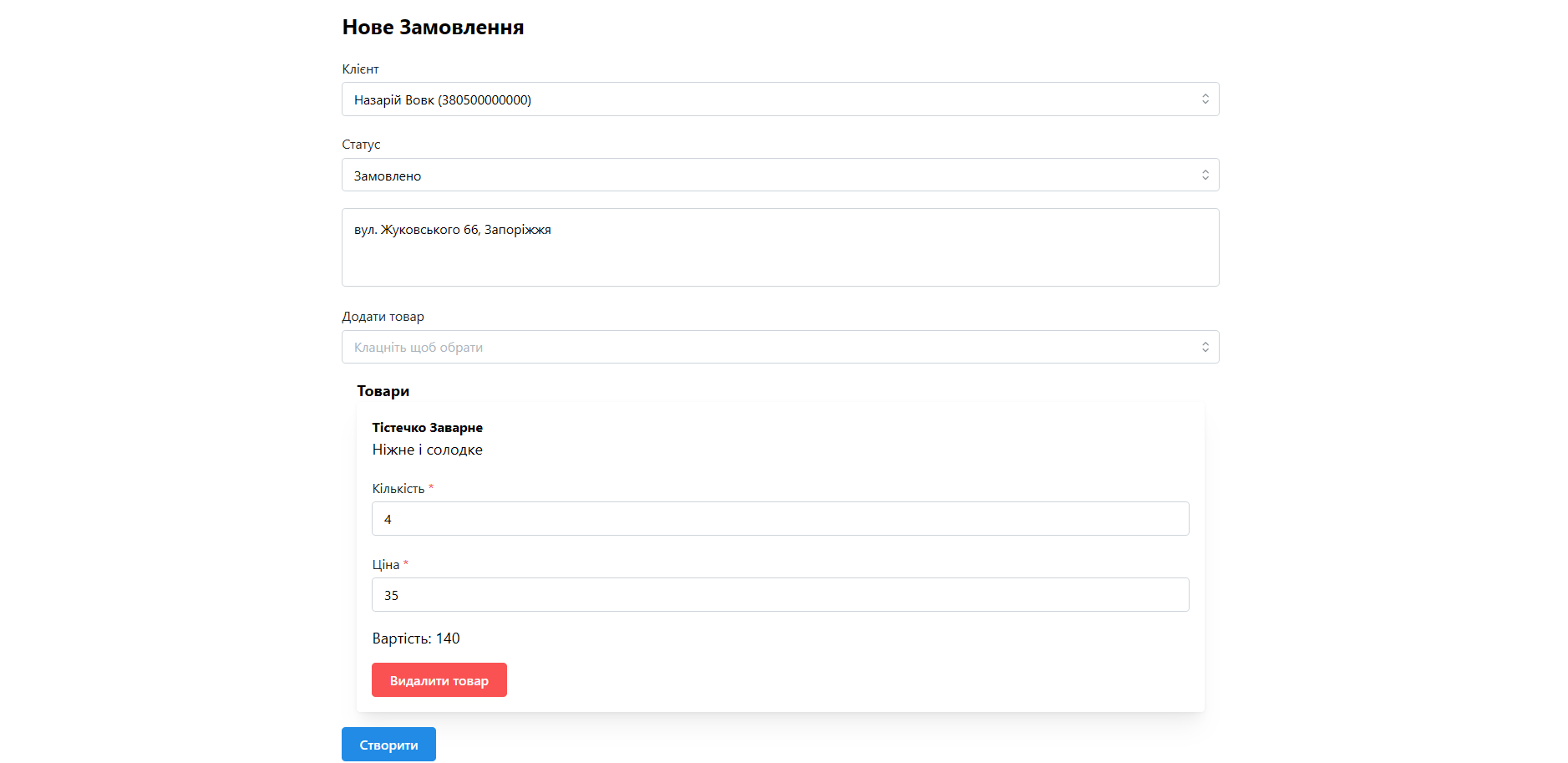
Ілюстрація 4

(тут – ціна має бути більше нуля)



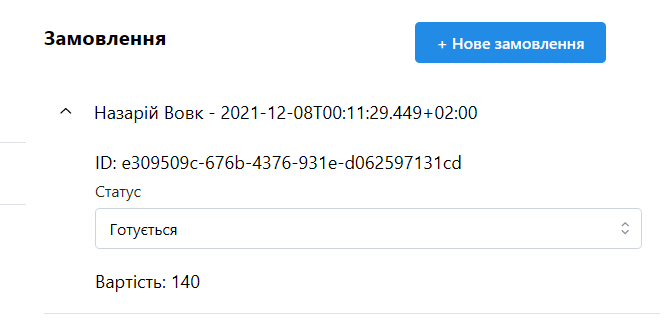
Ілюстрація 5

Форма нового замовлення з вибором клієнта зі значеть в базі данних, так сaмо як і з вибором товарів з можливістю редагування:



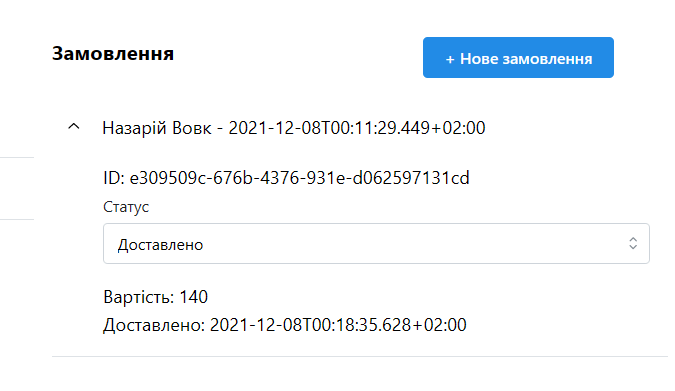
Ілюстрація 6

Після створення, статус замовлення може бути змінено:



Ілюстрація 7

Коли статус змінюється на «Доставлено», автоматично з’являється дата доставки:



Ілюстрація 8

# Етап 7 Визначення прав доступу до бази даних

Розподілити повноваження між користувачами. Перевірити працездатність всієї інформаційної системи з урахуванням обмежень доступу.

Особливість предметної області у тому, що усі користувачі системи мають однакові права. Тому буда обрана СУБД SQLite, яка не передбачає прав доступу окрім прав доступу до файлу бази данних на диску.

# Висновок

В результаті виконаної курсової роботи було створено базу даних пекарні. Використання цієї бази даних має підвищити швидкість та зручність роботи менеджерів, пекарів та кур’єрів.

Додаток дозволяє виконувати пошук клієнтів, товарів та замовлень, а також їх створення та модифікацію.

Під час написання курсової роботи було:

* Оброблено  предметну область;
* Описано реалізацію компонентів системи;
* Розроблено методику взаємодії з програмою.

Створена програма відповідає таким вимогам проектованої системи:

* виведення на екран вихідних форм;
* перехід між формами;
* здатність редагувати дані;
* програма повинна мати зручний, максимально орієнтований на користувача інтерфейс.

# Список літератури

1. https://www.sqlite.org/docs.html
2. <https://www.sqlitetutorial.net/>