

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

Допущено до захисту

Завідувач кафедри комп'ютерних наук
доктор технічних наук, професор

_____ Андрій БОНДАРЧУК

« ____ » _____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»

Спеціальність 122 Комп'ютерні науки

Освітня програма 122.00.01 Інформатика

**Тема: ВЕБДОДАТОК РЕСТОРАННОГО БІЗНЕСУ
З ВИКОРИСТАННЯМ ASP .NET CORE I VUE**

Виконав
студент групи ІНб-2-22-4.0д
Бабенко Артем Романович

(підпис)

Науковий керівник
кандидат технічних наук
доцент кафедри комп'ютерних наук
Рзаєва Світлана Леонідівна

(підпис)

Київ – 2026

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри комп'ютерних наук
доктор технічних наук, професор

_____ Андрій БОНДАРЧУК
« ____ » _____ 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

**«Вебдодаток ресторанного бізнесу
з використанням ASP .NET Core і Vue»**

Виконавець – студент групи ІНБ-2-22-4.0д,
спеціальності 122 Комп'ютерні науки,
освітньої програми 122.00.01 Інформатика
Бабенко Артем Романович

1. Вихідні дані: Інформаційна система, що дозволяє автоматизувати внутрішні бізнес-процеси ресторану, включаючи управління меню, замовленнями та взаємодію з клієнтами.
2. Основні завдання: Проаналізувати існуючі вебдодатки для ресторанного бізнесу та визначити їх функціональні можливості; Обрати стек технологій для розробки вебдодатку; Визначити вимоги до функціоналу системи, включно з адміністративною панеллю та клієнтським інтерфейсом; Спроектувати архітектуру системи; Реалізувати користувацький інтерфейс та серверний додаток; Провести тестування вебдодатку.
3. Пояснювальна записка: Обсяг – до 45 стор. формату А4 комп'ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри.
4. Графічні матеріали: не передбачено.
5. Додатки: лістинг програми.
6. Строк подання роботи на кафедру: « 4 » червня 2026 р.

Науковий керівник
Рзаєва Світлана Леонідівна
доцент кафедри комп'ютерних наук

« ____ » _____ 2026 р.

Виконавець:
Бабенко Артем Романович

« ____ » _____ 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапу дипломної роботи	Строк виконання етапу роботи	Примітка
1	Затвердження теми кваліфікаційної роботи бакалавра	31.10.25	Виконано
2	Аналіз проблеми, вивчення аналогів	01.11.25 - 11.01.26	Виконано
3	Формування попереднього змісту та плану-графіка виконання	26.01.26 - 15.03.26	Виконано
4	Пошук джерел для дослідження	15.03.26 - 31.03.26	Виконано
5	Створення вступної частини	01.04.26 - 10.04.26	Виконано
6	Написання першого розділу	11.04.26 - 25.04.26	Виконано
7	Написання другого розділу	26.04.26 - 15.05.26	Виконано
8	Написання третього розділу	16.05.26 - 31.05.26	Виконано
9	Підготовка пояснювальної записки, графічних матеріалів, додатків	01.06.26 - 15.06.26	Виконано
10	Захист кваліфікаційної роботи	19.06.26	Виконано

Здобувач _____

Керівник роботи _____

РЕФЕРАТ

Кваліфікаційна робота: 45 с., 16 рис., 1 табл., 28 літературних джерел.

Актуальність: обумовлена необхідністю автоматизації процесів обслуговування клієнтів та управління закладами ресторанного бізнесу. Використання вебдодатків, зокрема електронних меню з доступом через QR-код, дозволяє підвищити ефективність роботи закладу, зменшити навантаження на персонал та покращити якість обслуговування клієнтів.

Об'єкт дослідження: вебтехнології у сфері ресторанного бізнесу.

Предмет дослідження: процеси проєктування та розроблення вебдодатка ресторанного бізнесу з використанням технологій ASP.NET Core і Vue.

Мета роботи: розробка вебдодатку ресторанного бізнесу з використанням ASP.NET Core і Vue.js.

Завдання:

1. провести аналіз предметної області та існуючих рішень;
2. визначити функціональні та нефункціональні вимоги до системи;
3. спроектувати архітектуру вебдодатку та базу даних; реалізувати серверну частину;
4. розробити клієнтську частину та адміністративну панель;
5. провести тестування та оцінку ефективності системи.

Результати роботи: розроблено вебдодаток ресторанного бізнесу, який включає клієнтську частину у вигляді QR-меню та адміністративну панель для управління закладом.

Практичне значення: полягає у можливості використання розробленого вебдодатку в реальних закладах ресторанного бізнесу для автоматизації процесів обслуговування клієнтів, управління меню та замовленнями.

Ключові слова: ВЕБДОДАТОК, РЕСТОРАННИЙ БІЗНЕС, QR-МЕНЮ, ASP.NET CORE, VUE.JS, АВТОМАТИЗАЦІЯ, REST API.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

SPA - Single Page Application (односторінковий застосунок) - веб-застосунок, який працює без перезавантаження сторінки.

UI - User Interface (інтерфейс користувача) - візуальна частина онлайн-меню.

UX - User Experience (досвід користувача) - зручність і ефективність взаємодії користувача з меню.

CSS - Cascading Style Sheets (каскадні таблиці стилів) - використовується для оформлення інтерфейсу.

JavaScript (JS) - мова програмування для реалізації логіки клієнтської частини.

CRUD - Create, Read, Update, Delete - базові операції роботи з даними (наприклад, для страв у меню).

ASP.NET Core - фреймворк для створення серверної частини вебзастосунку.

Entity Framework Core (EF Core) - ORM для роботи з базою даних у .NET.

CRM - Customer Relationship Management - система для управління взаємодією з клієнтами.

DBMS / СУБД - Database Management System - система управління базами даних.

ORM - Object-Relational Mapping - технологія відображення об'єктів у реляційні дані.

Vite - інструмент для збірки та запуску фронтенд-застосунків.

Docker - платформа для контейнеризації та ізоляції середовища виконання.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1	9
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ У СФЕРІ РЕСТОРАННОГО БІЗНЕСУ	9
1.1 Огляд вебдодатків ресторанного бізнесу та їх функціональності	9
1.2 Аналіз систем електронного меню та управління закладом	10
1.3 Аналіз функціональних можливостей аналогів	13
1.4 Визначення вимог до вебдодатку ресторанного бізнесу	15
Висновки до розділу 1	17
РОЗДІЛ 2	18
ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБДОДАТКУ	18
2.1 Вибір технологічного стеку	18
2.2 Проєктування архітектури системи	19
2.3 Проєктування бази даних	21
2.4 Опис реалізованих модулів системи	23
2.5 Розробка серверної частини	25
2.6 Розробка клієнтської частини (QR-меню)	27
2.7 Розробка адміністративної панелі	28
Висновки до розділу 2	30
РОЗДІЛ 3	31
ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ	31
3.1 Тестування функціональності вебдодатку	31
3.2 Аналіз продуктивності системи	32
3.3 Оцінка зручності використання	33
3.4 Перспективи розвитку та масштабування системи	34
Висновки до розділу 3	35
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	38

ВСТУП

Інформаційні технології активно впливають на різні сфери бізнесу, зокрема і на ресторанний бізнес. Багато закладів вже використовують веб- та мобільні рішення для того, щоб спростити обслуговування клієнтів, зменшити навантаження на персонал і зробити роботу більш ефективною. Одним із таких рішень є електронні меню, онлайн-замовлення та адміністративні панелі для управління закладом.

Якщо подивитися на сучасний ринок, то можна побачити, що вже існує багато готових систем, таких як Choice, SmartSpot або Expienza. Вони дозволяють переглядати меню, оформлювати замовлення та керувати роботою закладу. Але на практиці ці рішення не завжди підходять повністю. У багатьох із них є обмеження - наприклад, складно щось змінити під свої потреби, є залежність від сервісу, або потрібно платити за додатковий функціонал. Через це виникає потреба у створенні власного рішення, яке можна налаштувати під конкретний заклад. Такий вебдодаток має бути достатньо гнучким, щоб враховувати різні сценарії використання, і при цьому залишатися зручним як для клієнтів, так і для персоналу.

Актуальність роботи полягає у необхідності автоматизації процесів обслуговування клієнтів та управління закладами ресторанного бізнесу, що зумовлено розвитком цифрових технологій та зростанням вимог до якості сервісу. Використання сучасних вебтехнологій, зокрема ASP.NET Core та Vue.js, дозволяє створювати швидкі, зручні та масштабовані системи [10].

Метою роботи є розробка вебдодатку ресторанного бізнесу з використанням ASP.NET Core і Vue.js, який забезпечує зручний доступ до меню для клієнтів та надає інструменти для управління закладом.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. провести аналіз існуючих рішень у сфері автоматизації ресторанного бізнесу;

2. визначити основні вимоги до системи;
3. спроектувати архітектуру вебдодатку;
4. реалізувати серверну частину;
5. розробити клієнтську частину;
6. створити адміністративну панель;
7. провести тестування та оцінку роботи системи.

Об'єктом дослідження є вебтехнології у сфері ресторанного бізнесу.

Предметом дослідження є процеси проєктування та розроблення вебдодатка ресторанного бізнесу з використанням технологій ASP.NET Core і Vue.

Методи дослідження включають аналіз існуючих рішень, проєктування програмних систем, використання об'єктно-орієнтованого підходу, моделювання баз даних та тестування програмного забезпечення.

Практичне значення отриманих результатів полягає у можливості використання розробленого вебдодатку в реальних закладах ресторанного бізнесу для автоматизації роботи, управління меню та замовленнями.

Результати роботи апробовано шляхом публікації тез доповіді на конференції Інформаційні технології – 2026: зб. тез XII Всеукраїнської науково-практичної конференції молодих учених, 14 трав. 2026 р., м. Київ / Київський столичний університет імені Бориса Грінченка [29].

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ У СФЕРІ РЕСТОРАННОГО БІЗНЕСУ

1.1 Огляд вебдодатків ресторанного бізнесу та їх функціональності

На сьогодні ресторанний бізнес активно використовує інформаційні технології для покращення обслуговування клієнтів та оптимізації роботи закладу. Одним із таких рішень є вебдодатки, які дозволяють автоматизувати процес прийому замовлень, управління меню та взаємодію з клієнтами. Незважаючи на це, у багатьох закладах досі використовуються звичайні підходи до обслуговування. Це часто призводить до певних проблем, наприклад, повільного обслуговування або помилок при передачі замовлення. Також паперові меню не дуже зручні, оскільки їх складно швидко оновлювати - потрібно витрачати час і кошти на друк нових.

Одним із сучасних рішень цієї проблеми є електронне меню, яке відкривається через QR-код [22]. Клієнт просто сканує код зі столу і одразу бачить меню на своєму телефоні. Це зручно, швидко і не потребує участі персоналу на цьому етапі. QR-меню має ряд переваг у порівнянні з традиційним. По-перше, інформацію можна змінити буквально за кілька хвилин. По-друге, це зменшує контакт між людьми, що теж може бути важливим. Крім того, у таких меню можна реалізувати пошук, фільтри або, наприклад, показ популярних страв.

Якщо говорити загалом, то системи автоматизації ресторанного бізнесу бувають різні. Це можуть бути окремі системи для меню, POS-системи для роботи із замовленнями або CRM-системи для роботи з клієнтами [23]. У сучасних рішеннях часто ці можливості об'єднуються в одну систему. Зазвичай вебдодатки ресторанного бізнесу складаються з двох частин. Перша - це клієнтська частина, яку бачить відвідувач (наприклад, QR-меню). Друга - це адміністративна панель, через яку персонал керує системою [24]. Через

адміністративну панель можна редагувати меню, змінювати ціни, додавати нові страви або переглядати замовлення. По суті, це основний інструмент для роботи із системою. Якщо адміністративна панель реалізована незручно, це може ускладнювати роботу персоналу, навіть за наявності необхідного функціоналу. Деякі системи також мають додаткові можливості, наприклад аналітику продажів або інтеграцію з платіжними сервісами. Це дозволяє не тільки автоматизувати процеси, але й краще розуміти, як працює заклад.

Отже, можна сказати, що використання вебдодатків у ресторанному бізнесі значно спрощує роботу персоналу і покращує обслуговування клієнтів. Такі рішення допомагають зменшити кількість помилок, пришвидшити обслуговування та зробити сам процес більш зручним як для клієнтів, так і для працівників.

1.2 Аналіз систем електронного меню та управління закладом

Існує велика кількість сервісів, які дозволяють реалізувати електронне меню та автоматизувати роботу ресторану. Найбільш поширеними серед них є Choice, SmartSpot та Expienza. Вони пропонують схожий базовий функціонал, однак відрізняються за можливостями, складністю та підходом до використання. Дані системи дозволяють клієнтам отримувати доступ до меню через QR-код, переглядати страви, їх опис та ціни, а іноді - навіть оформлювати замовлення без участі офіціанта. Це зменшує навантаження на персонал та робить обслуговування швидшим.

Choice (рис. 1.1) - це сервіс, який в основному орієнтований на створення електронного меню. Переваги: простота використання, швидке створення меню, швидкий запуск системи, наявність QR-генерації. Недоліки: обмежений функціонал управління замовленнями, відсутність розширеної аналітики, низький рівень кастомізації. Заклад може досить швидко створити своє меню, додати категорії, страви та отримати QR-код для розміщення на столах. З функціональної точки зору, Choice більше підходить для невеликих закладів, яким потрібно просто показувати меню клієнтам.


Продукти ▾ Рішення ▾ Тарифи Кейси Блог Партнери Співпраця ▾ Вакансії
Спробувати безкоштовно
Увійти

CHOICE — СИСТЕМА №1 ДЛЯ ВЛАСНОЇ ДОСТАВКИ З РЕСТОРАНІВ

Запускайте **власну доставку** легко та без зайвого клопоту!



CHOICE — ЦЕ ГОТОВЕ РІШЕННЯ, ЩО ПОЄДНУЄ:



Сучасний сайт для замовлень — адаптований під мобільні, реклами й SEO.



Усі замовлення в одному місці — доставка, самовивіз і маркетплейси.



Аналітику та базу клієнтів для повторних продажів і збільшення прибутку.



Інтеграції з кур'єрськими сервісами або можливість працювати з власними кур'єрами.

З Choice ви контролюєте все — від першого кліку до задоволеного гостя ❤️

Рисунок 1.1. Choice

SmartSpot (рис. 1.2) має більш широкий функціонал у порівнянні з простими рішеннями. Переваги: підтримка замовлень, базова аналітика, ширші можливості управління закладом. Недоліки: складніший інтерфейс, вища складність впровадження, частково обмежена кастомізація. Окрім електронного меню, система дозволяє працювати із замовленнями, а також надає базові інструменти аналітики. Інтерфейс системи більш складний, ніж у простих рішень, але водночас і більш функціональний. SmartSpot може підійти для

закладів середнього рівня, яким вже недостатньо просто меню, а потрібен певний контроль за замовленнями та клієнтами [7].

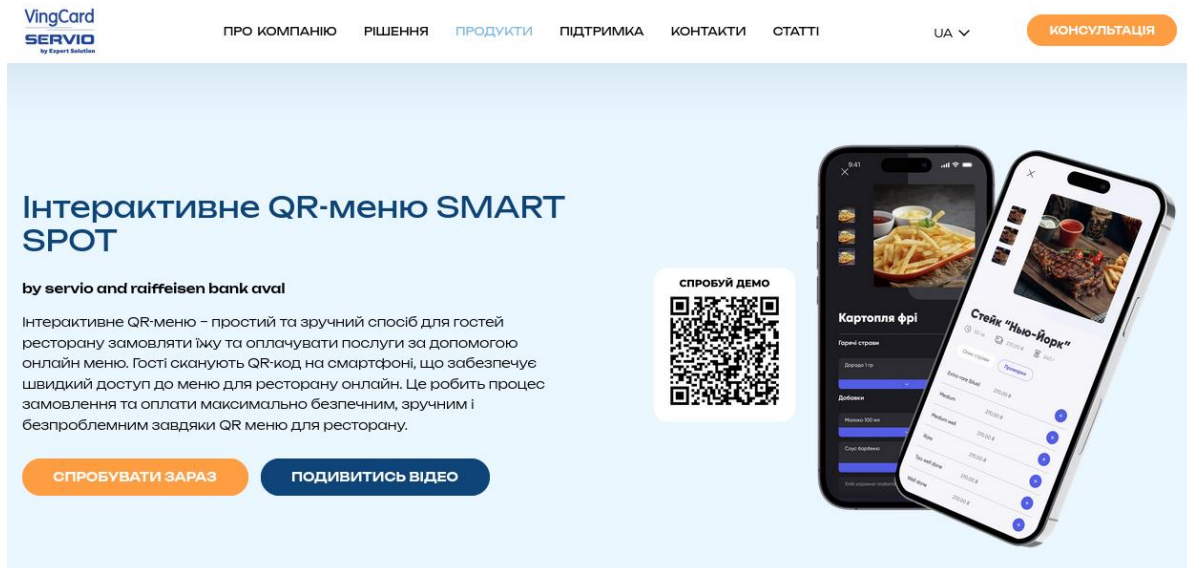


Рисунок 1.2. SmartSpot

Expirenza (рис. 1.3) є більш комплексним рішенням, яке охоплює не лише електронне меню, але й управління замовленнями, аналітику та взаємодію з клієнтами [8]. Переваги: повний цикл управління (меню, замовлення, аналітика), розвинена система взаємодії з клієнтами, широкий функціонал. Недоліки: складність використання, платні функції, обмежена гнучкість налаштувань під конкретний заклад. Це система, яка вже наближена до повноцінної платформи для ресторанного бізнесу. Разом з тим, її використання ускладнюється підвищеною складністю впровадження та необхідністю використання платних тарифів для доступу до розширених функцій. Крім того, як і більшість готових SaaS-рішень, система має обмежені можливості кастомізації, що може бути недоліком для закладів з унікальними бізнес-процесами.

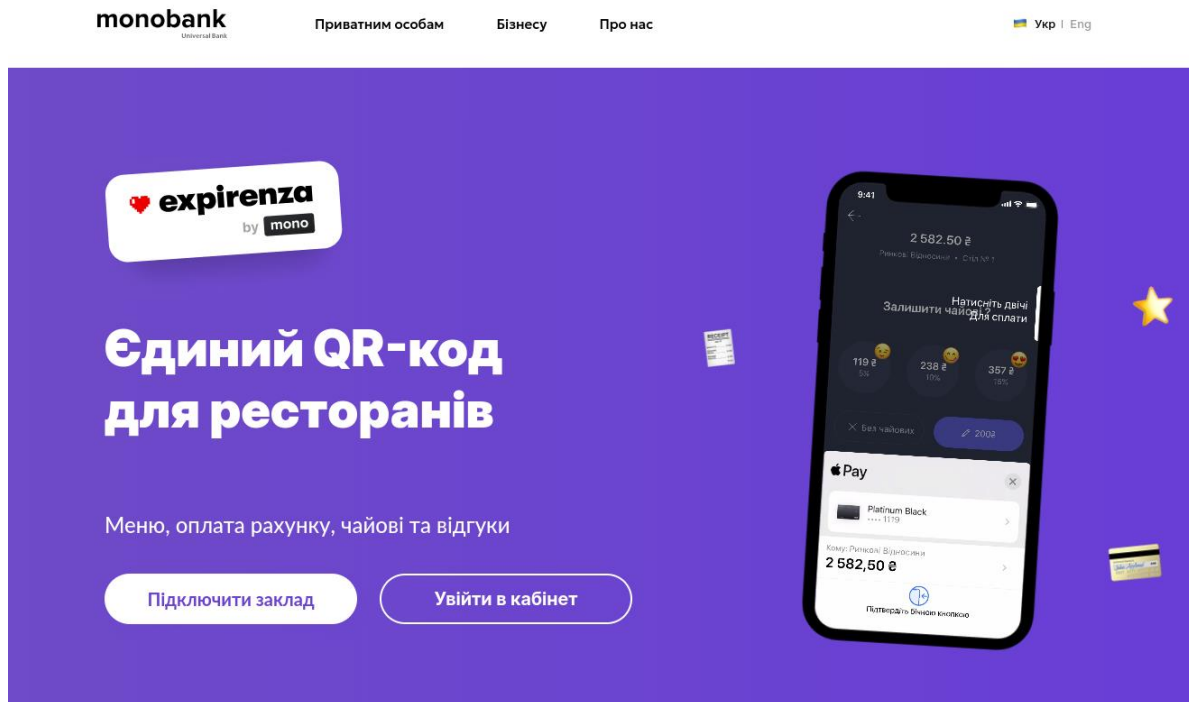


Рисунок 1.3. Expirenza

1.3 Аналіз функціональних можливостей аналогів

Для того щоб краще зрозуміти, які можливості мають сучасні системи електронного меню, доцільно розглянути їх функціонал більш детально. Більшість таких сервісів мають схожий базовий набір функцій, але відрізняються за рівнем складності та додатковими можливостями.

До основних функцій, які зустрічаються майже у всіх системах, можна віднести відображення електронного меню, поділ страв на категорії, а також показ інформації про страву - опису, ціни та зображення. Доступ до меню зазвичай здійснюється за допомогою QR-коду, що значно спрощує взаємодію клієнта із закладом. Окрім базового функціоналу, деякі системи пропонують додаткові можливості. Наприклад, клієнт може самостійно оформити замовлення зі свого пристрою, а персонал - отримати його в системі та обробити [23]. Також у більш розвинених рішеннях присутня аналітика продажів, яка дозволяє оцінювати популярність страв або загальну ефективність роботи закладу. До розширених можливостей також можна віднести інтеграцію з платіжними сервісами, управління користувачами та ролями, а також контроль

замовлень у реальному часі [24]. Проте такі функції зазвичай доступні не у всіх системах або потребують додаткової оплати.

Загалом можна сказати, що не всі системи реалізують повний набір функцій. Деякі з них орієнтовані лише на відображення меню і мають мінімальний функціонал. Інші, навпаки, пропонують багато можливостей, але можуть бути складними у використанні або занадто дорогими для невеликих закладів.

Для того щоб краще побачити різницю між основними системами, доцільно порівняти їх функціональні можливості.

Таблиця 1.1

Порівняльний аналіз систем

Функціональність	Choice	SmartSpot	Expirenza
QR-меню	+	+	+
Відображення страв	+	+	+
Оформлення замовлення	- / частково	+	+
Управління замовленнями	-	+	+
Аналітика	базова	базова	розширена
Адміністративна панель	+	+	+
Гнучкість налаштування	середня	низька	середня
Вартість	28-70\$	35\$-50\$	1.3%

Проведений аналіз показує, що всі розглянуті системи мають базовий функціонал електронного меню, проте відрізняються за рівнем складності та можливостями. Прості рішення є зручними у використанні, але обмежені у функціональності, тоді як більш складні системи надають ширші можливості, але потребують більших витрат і часу на впровадження. Крім того, більшість систем є закритими платформами, що обмежує можливості їх адаптації під конкретні потреби закладу. Це означає, що бізнес змушений підлаштовуватися під систему, а не навпаки.

1.4 Визначення вимог до вебдодатку ресторанного бізнесу

На основі проведеного аналізу предметної області, існуючих аналогів та сучасних підходів до автоматизації ресторанного бізнесу було визначено основні вимоги до вебдодатку. Визначені вимоги формують основу для проєктування системи та дозволяють забезпечити ефективну взаємодію між клієнтами, персоналом закладу та адміністрацією.

Основним призначенням системи є автоматизація процесу обслуговування клієнтів за допомогою електронного QR-меню, спрощення процесу оформлення замовлень, а також забезпечення зручного управління закладом ресторанного бізнесу. Система повинна забезпечувати швидкий доступ користувачів до меню через QR-код без необхідності встановлення додаткового програмного забезпечення. Після сканування QR-коду клієнт має отримувати доступ до меню конкретного столу та мати можливість оформити замовлення через веб-інтерфейс. Інтерфейс повинен бути простим, зрозумілим та адаптованим для мобільних пристроїв.

До основних функціональних вимог системи належать:

1. можливість перегляду електронного меню через QR-код;
2. підтримка структури меню з категоріями товарів;
3. відображення детальної інформації про страви та напої;
4. можливість оформлення замовлення клієнтом;
5. передача замовлення до адміністративної системи закладу;
6. підтримка історії статусів замовлення;
7. можливість призначення кількох виконавців на одне замовлення;
8. розмежування ролей персоналу (адміністратор, офіціант, кухар тощо);
9. управління користувачами та правами доступу;
10. підтримка механізму запрошення нових працівників до системи;
11. підтримка роботи мережі закладів;
12. створення та управління столами закладу;

13. генерація унікальних QR-кодів для столів;
14. управління товарами та категоріями;
15. можливість встановлення різних цін товарів залежно від закладу;
16. підтримка доступності окремих товарів лише у визначених закладах;
17. збереження даних у базі даних;
18. авторизація та автентифікація користувачів;

Важливою частиною системи є адміністративна панель, яка повинна забезпечувати зручне управління всіма компонентами системи. Адміністратор повинен мати можливість редагувати меню, змінювати ціни, управляти замовленнями, контролювати користувачів та переглядати інформацію щодо діяльності закладу.

Окрему увагу необхідно приділити модулю аналітики. Система повинна забезпечувати можливість перегляду інформації про кількість продажів, популярність товарів та прибуток закладу за певний період. Дані мають відображатися у зручному для сприйняття вигляді, зокрема у вигляді діаграм чи графіків.

Крім функціональних можливостей, система повинна відповідати ряду нефункціональних вимог, які визначають якість та стабільність її роботи.

До основних нефункціональних вимог належать:

1. висока швидкість завантаження сторінок;
2. стабільна робота системи при одночасному використанні кількох користувачами;
3. адаптивність інтерфейсу для смартфонів, планшетів та персональних комп'ютерів;
4. зручний та інтуїтивно зрозумілий інтерфейс;
5. масштабованість системи та можливість розширення функціоналу;
6. надійність збереження інформації;
7. забезпечення базового рівня інформаційної безпеки;
8. підтримка сучасних веб-браузерів;
9. можливість супроводу та подальшої модифікації системи.

Також система повинна бути достатньо гнучкою для адаптації під різні типи закладів ресторанного бізнесу. Це дозволяє використовувати розроблений вебдодаток як для окремих ресторанів чи кафе, так і для мережі закладів із різною структурою меню та особливостями організації обслуговування.

Висновки до розділу 1

У цьому розділі проаналізовано стан ресторанного бізнесу та існуючі програмні рішення для автоматизації обслуговування клієнтів і управління закладом. Визначено основні можливості систем електронного меню та адміністративних панелей, а також їх переваги і недоліки.

Серед переваг - базовий функціонал для клієнтів і персоналу, інтеграція з платіжними сервісами та аналітика продажів. Недоліки включають обмежену гнучкість, залежність від сторонніх сервісів і наявність додаткової плати за розширений функціонал.

На основі аналізу сформульовані функціональні та нефункціональні вимоги до вебдодатку, включаючи перегляд меню через QR-код, оформлення замовлень, управління меню та замовленнями, а також вимоги до швидкодії, безпеки та адаптивності інтерфейсу.

Результати цього розділу створюють основу для подальшого проєктування та розробки вебдодатку, який буде зручним для клієнтів та ефективним для персоналу.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБДОДАТКУ

2.1 Вибір технологічного стеку

Для розробки вебдодатку ресторанного бізнесу було обрано стек технологій, який дозволяє зручно реалізувати як серверну, так і клієнтську частину системи.

Серверна частина реалізована на ASP.NET Core. Дана технологія добре підходить для створення вебдодатків і API, оскільки забезпечує достатню швидкість роботи та гнучкість у розробці [10]. Крім того, вона підтримує кросплатформеність, що дозволяє запускати додаток на різних операційних системах [30].

Для роботи з базою даних використовується Entity Framework Core (EF Core). Це ORM-бібліотека, яка дозволяє працювати з базою даних через об'єкти та класи, замість написання великої кількості SQL-запитів [11]. Це спрощує розробку, робить код більш зрозумілим і дозволяє швидше вносити зміни у структуру бази даних. Для зберігання даних використовується база даних PostgreSQL. Вона є надійною та широко використовується у веб-розробці, добре підходить для роботи з реляційними даними, такими як меню, замовлення та користувачі [12].

Для безпеки особистих даних користувачів використовується криптографічна бібліотека BCrypt [16]. Він застосовується для хешування паролів перед їх збереженням у базі даних. Це означає, що навіть у випадку доступу до бази даних паролі не зберігаються у відкритому вигляді, що значно підвищує рівень безпеки системи.

Клієнтська частина реалізована за допомогою Vue.js [9]. Цей фреймворк дозволяє зручно будувати інтерфейс у вигляді компонентів, що спрощує розробку та подальшу підтримку додатку [31]. Також він добре підходить для створення динамічних інтерфейсів, таких як QR-меню. Для взаємодії між

клієнтом і сервером використовується бібліотека Axios. Вона дозволяє виконувати HTTP-запити до API та отримувати дані у форматі JSON [15]. Це спрощує обмін даними між фронендом і бекендом. Для збірки та запуску клієнтської частини використовується Vite. Це сучасний інструмент, який забезпечує швидкий запуск проєкту та оновлення сторінки під час розробки [14]. Завдяки цьому процес розробки стає швидшим і зручнішим.

Для розгортання та ізоляції середовища використовується Docker. Це дозволяє запускати додаток у контейнерах та уникати проблем із налаштуванням середовища на різних машинах [13].

Обраний стек є достатньо простим у використанні, але водночас дозволяє реалізувати всі необхідні функції для даного проєкту. Він забезпечує зручність розробки, масштабованість та можливість подальшого розширення системи.

2.2 Проєктування архітектури системи

Архітектура вебдодатку побудована за клієнт-серверним принципом, де основна логіка зосереджена на сервері, а клієнтська частина відповідає за відображення даних і взаємодію з користувачем [3]. У межах проєкту реалізовано два окремих вебінтерфейси: клієнтський сайт у вигляді QR-меню та адміністративна панель для управління закладом. Вони виконують різні функції, але працюють як частини однієї системи.

Клієнтська частина у вигляді QR-меню орієнтована на відвідувачів закладу. Вона дозволяє переглядати меню, отримувати інформацію про страви та взаємодіяти із системою безпосередньо зі смартфона. Адміністративна панель, у свою чергу, використовується персоналом для управління меню, редагування позицій, зміни цін та роботи із замовленнями. Такий поділ дозволяє зробити інтерфейси більш зручними, оскільки кожен з них розрахований на конкретний тип користувача. Кожен із цих інтерфейсів взаємодіє зі своїм сервером, проте обидві серверні частини побудовані на спільній логіці та використовують одну базу даних (рис. 2.1). Це означає, що всі дані в системі є узгодженими: наприклад, якщо змінити меню в адміністративній панелі, ці зміни одразу будуть доступні у

QR-меню. Використання спільного ядра дозволяє уникнути дублювання функціоналу та спрощує підтримку системи [2].

Серверна частина відповідає за обробку запитів від клієнтів, виконання бізнес-логіки та взаємодію з базою даних. Обмін даними між клієнтом і сервером відбувається через REST API, що дозволяє чітко розділити фронтенд і бекенд та зробити систему більш гнучкою.

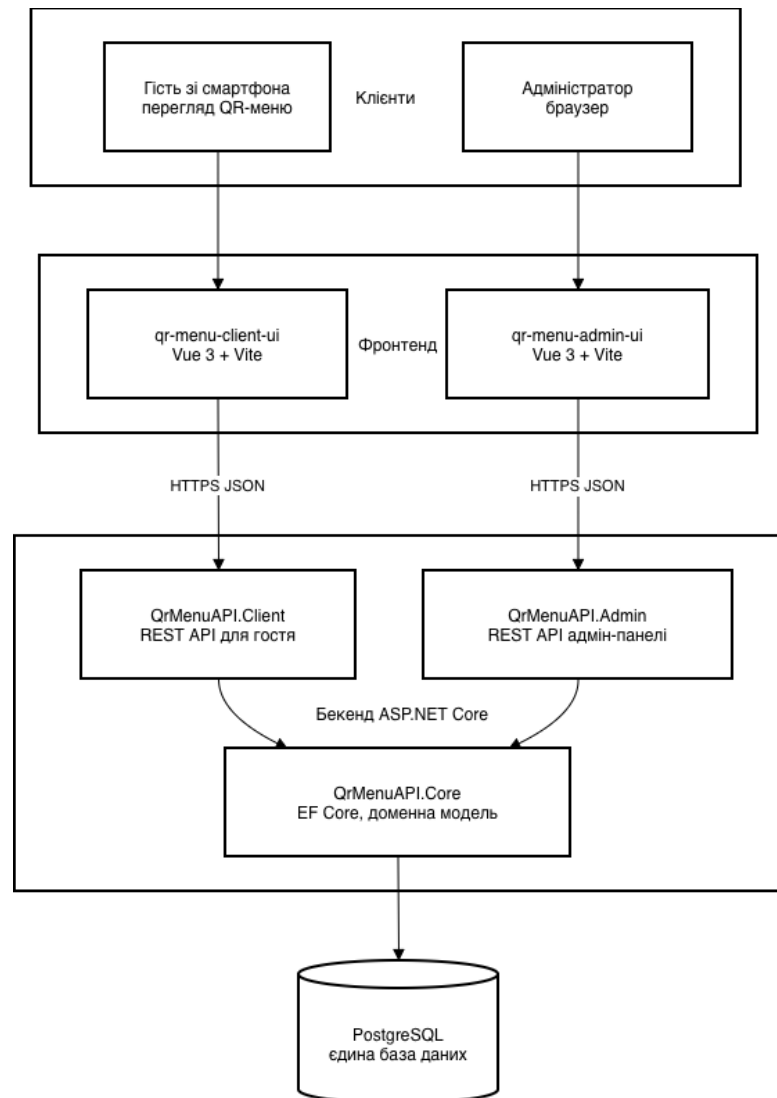


Рисунок 2.1. Модель архітектури системи

У цілому система побудована за багатошаровим підходом, де окремо виділяється рівень інтерфейсу, рівень бізнес-логіки та рівень роботи з даними [5]. Така організація дозволяє легше розширювати функціонал у майбутньому, а також спрощує внесення змін у різні частини системи без порушення її роботи.

2.3 Проєктування бази даних

Для реалізації вебдодатку була спроектована реляційна база даних з використанням СУБД PostgreSQL [12]. Основною задачею при проєктуванні було забезпечити зручне зберігання даних, їх цілісність, а також можливість масштабування системи у майбутньому.

База даних побудована за принципами нормалізації, що дозволяє уникнути дублювання даних та забезпечує логічну структуру зв'язків між таблицями (рис 2.2).

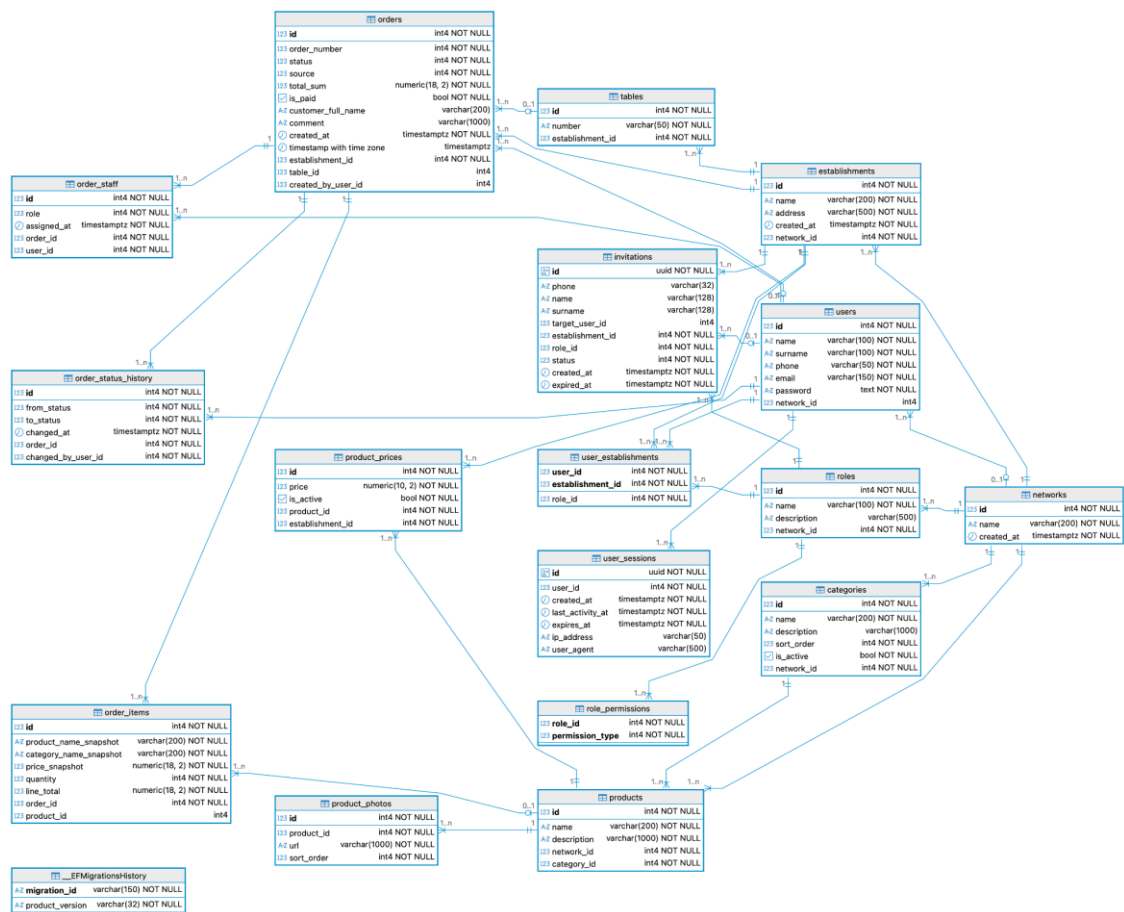


Рисунок 2.2. Фізична модель бази даних системи

Основні сутності системи:

- Мережа закладів (networks) - це верхній рівень, який об'єднує всі інші сутності. Одна мережа може мати декілька закладів, користувачів, ролей та товарів.
- Заклади (establishments) - представляють конкретні точки (наприклад, ресторан або кафе), які належать до певної мережі.

- Категорії (categories) - використовуються для групування товарів (наприклад, напої, десерти). Кожна категорія прив'язана до мережі.
- Товари (products) - містять інформацію про позиції меню. Вони належать до категорій і також прив'язані до мережі.
- Ціни товарів (product_prices) - реалізовані окремою таблицею, що дозволяє задавати різні ціни на один і той самий товар у різних закладах.
- Фото товарів (product_photos) - дозволяють зберігати декілька зображень для одного товару з можливістю сортування.

Користувачі та ролі:

- Користувачі (users) - містять базову інформацію (ім'я, телефон, email, пароль).
- Ролі (roles) - визначають права доступу в межах мережі.
- Права доступу (role_permissions) - реалізують набір дозволів для кожної ролі.
- Зв'язок користувачів із закладами (user_establishments) - дозволяє одному користувачу працювати в кількох закладах з різними ролями.

Для роботи з замовленнями використовується кілька таблиць:

- Замовлення (orders) - основна таблиця, що містить інформацію про замовлення (статус, сума, джерело, чи оплачено).
- Позиції замовлення (order_items) - містять список товарів у замовленні. Використовується підхід зі збереженням "snapshot" (назва, ціна, категорія), щоб дані не змінювались після редагування товару.
- Історія статусів (order_status_history) - дозволяє відстежувати зміну статусів замовлення.
- Персонал замовлення (order_staff) - зберігає інформацію про працівників, які працювали із замовленням.

Додаткові сутності:

- Столи (tables) - використовуються для ідентифікації столів у закладі (актуально для QR-меню).

- Сесії (user_sessions) - реалізують механізм авторизації та збереження сесій.

У базі даних використано зовнішні ключі (FOREIGN KEY), що забезпечують цілісність даних. Наприклад:

- один заклад належить до однієї мережі;
- товар належить до категорії;
- замовлення прив'язане до закладу та, опційно, до столу;
- користувач може бути прив'язаний до кількох закладів.

Для оптимізації запитів створені індекси, а також унікальні обмеження (наприклад, унікальність назви товару в межах мережі або номера столу в межах закладу).

Таким чином, спроектована база даних покриває всі основні бізнес-процеси системи: управління закладами, меню, користувачами та замовленнями. Структура є гнучкою, масштабованою та дозволяє легко додавати новий функціонал у майбутньому.

2.4 Опис реалізованих модулів системи

У розробленому вебдодатку реалізовано набір модулів, які забезпечують повноцінну роботу системи для ресторанного бізнесу. Кожен модуль відповідає за окрему частину функціоналу та взаємодіє з іншими модулями.

Одним із основних є модуль замовлень. Він відповідає за створення, обробку та зберігання замовлень користувачів. Користувач може сформувати замовлення через QR-меню, після чого воно потрапляє в систему та обробляється працівниками закладу. У межах цього модуля також реалізовано історію статусів замовлення. Це означає, що кожне замовлення проходить певні етапи (наприклад: створено, в обробці, готово, завершено), і всі зміни статусу зберігаються. Це дозволяє відстежувати повний життєвий цикл замовлення.

Додатково реалізована можливість призначення кількох виконавців на одне замовлення. Наприклад, кухар може відповідати за приготування страв, а офіціант - за їх подачу. Виконавці розподіляються за ролями, що дозволяє чітко

розмежувати відповідальність між працівниками. Також у системі передбачена історичність даних, тобто зберігається інформація про зміни, які відбувалися із замовленням або іншими сутностями. Це дозволяє аналізувати роботу системи в минулому.

Наступним важливим є модуль управління мережею закладів. Система дозволяє працювати не з одним рестораном, а з мережею закладів. Для кожного закладу можна створювати окремі столи, які використовуються для обслуговування клієнтів. Для кожного столу генерується унікальний QR-код. Відсканувавши цей код, клієнт отримує доступ до меню та може зробити замовлення безпосередньо зі свого пристрою. Це значно спрощує процес обслуговування та зменшує навантаження на персонал.

Окремо реалізовано модуль користувачів та ролей. У системі передбачено різні ролі, наприклад адміністратор, офіціант або кухар. Кожна роль має свій набір прав доступу до функціоналу системи. Це дозволяє обмежити доступ до певних дій та забезпечити безпеку даних. Управління користувачами включає створення, редагування та видалення облікових записів.

Також реалізовано механізм запрошень. Адміністратор може запросити нового користувача до системи, після чого той отримує можливість зареєструватися та приєднатися до конкретного закладу або мережі. Це зручно для додавання персоналу без ручного створення кожного облікового запису.

Модуль реєстрації та входу забезпечує авторизацію користувачів у системі. Користувачі можуть створювати облікові записи та входити в систему за допомогою логіна і пароля. Це дозволяє ідентифікувати користувачів і надавати їм доступ до відповідного функціоналу залежно від їх ролі.

Ще одним важливим модулем є управління товарами та категоріями. У системі можна створювати категорії товарів, наприклад напої, основні страви або десерти, та додавати до них відповідні позиції. Для кожного товару можна вказати назву, опис та інші характеристики. Також реалізовано можливість встановлення різної ціни для одного й того ж товару залежно від закладу. Крім цього, товар може бути доступний лише у певних закладах, а в інших - відсутній.

Це дозволяє гнучко налаштовувати меню для кожного закладу окремо та враховувати особливості мережі.

Для аналізу роботи закладу реалізовано модуль аналітики. Він дозволяє переглядати інформацію про кількість проданих товарів та загальний прибуток. Дані відображаються у вигляді діаграм, що спрощує їх сприйняття та дозволяє швидко оцінити ситуацію. Наприклад, можна побачити, які товари продаються найкраще або як змінюється прибуток за певний період.

Загалом можна сказати, що реалізовані модулі охоплюють основні потреби ресторанного бізнесу: від прийому замовлень і управління меню до контролю персоналу та аналізу діяльності. Всі модулі взаємодіють між собою, що забезпечує цілісну та зручну роботу системи.

2.5 Розробка серверної частини

Серверна частина реалізована у вигляді двох окремих серверних додатків і спільного ядра. Такий підхід використаний для того, щоб розділити функціонал для різних типів користувачів і не змішувати все в одному місці [3]. Серверна частина розроблена з використанням технології ASP.NET Core [10].

Перший серверний додаток працює з QR-меню, яким користуються відвідувачі закладу. Він відповідає за отримання списку категорій і товарів, показ інформації про заклад і створення замовлення. Структура API контролерів онлайн меню наведена на рисунку 2.3. Також через QR-код передається інформація про стіл, щоб розуміти, звідки зроблено замовлення. Цей додаток зроблений простішим, без складної логіки та без адміністративного функціоналу.

Другий серверний додаток використовується для адміністративної панелі. Через нього працівники керують товарами, категоріями, закладами, користувачами і ролями. Також тут відбувається робота із замовленнями - їх перегляд і зміна статусів. UML-діаграма класів контролерів API адміністративної панелі наведена на рисунку 2.4.

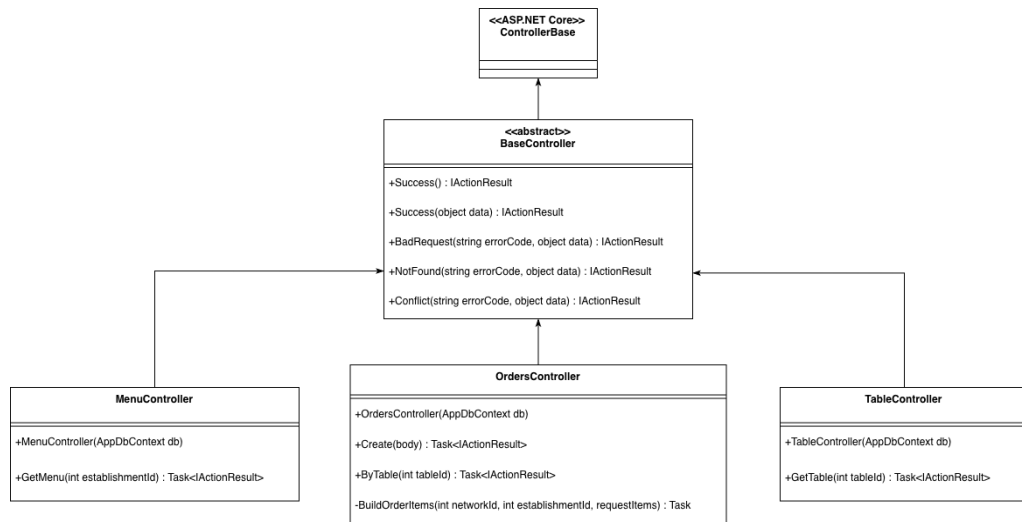


Рисунок 2.3. Модель класів контролерів онлайн меню

У цьому додатку реалізована перевірка прав доступу, щоб кожен користувач бачив і міг виконувати лише дозволені дії. Для безпеки паролі користувачів зберігаються у захешованому вигляді.

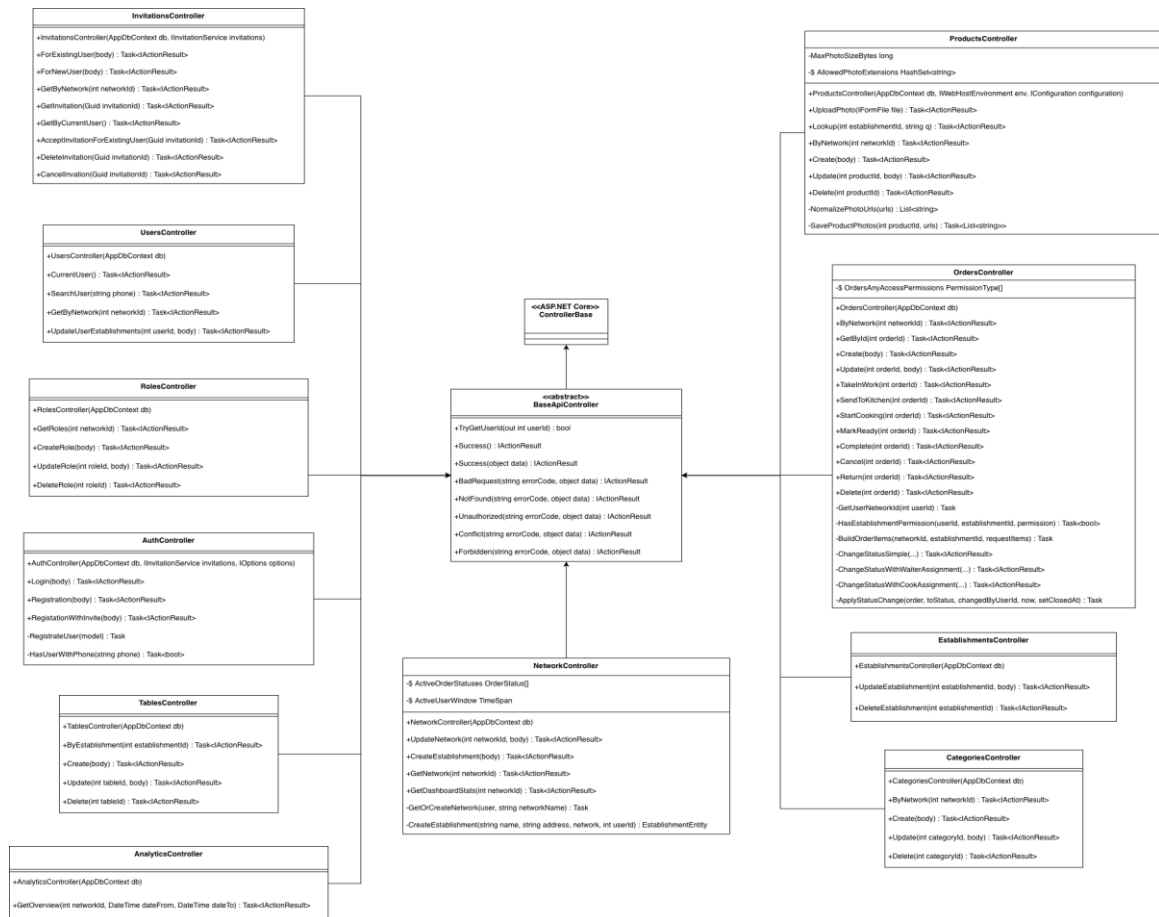


Рисунок 2.4. Модель класів контролерів адмін. панелі

Щоб не дублювати код, створено спільне ядро для обох серверних додатків. У ньому знаходяться сутності, робота з базою даних і загальна логіка [4]. Якщо щось змінюється, це одразу відображається в обох додатках.

Обидва серверні додатки працюють з однією базою даних. Для роботи з нею використовується ORM Entity Framework Core, тому замість написання SQL-запитів використовується робота через об'єкти. Це робить код простішим і більш зрозумілим [4].

Взаємодія між клієнтською частиною та сервером відбувається через HTTP API з використанням формату JSON.

2.6 Розробка клієнтської частини (QR-меню)

Клієнтська частина системи реалізована у вигляді вебдодатку з використанням фреймворку Vue.js [9]. Даний додаток призначений для відвідувачів закладу і відкривається після сканування QR-коду, розміщеного на столі. Основною його задачею є надання доступу до меню та можливість створення замовлення без взаємодії з персоналом.

Додаток має просту структуру та складається з чотирьох основних розділів. Перший розділ - головна сторінка. На ній відображається інформація про обраний стіл, а також розміщені основні елементи навігації, які дозволяють перейти до інших частин додатку (рис Б.1). Другий розділ - меню. У ньому користувач може переглядати категорії товарів, наприклад напої або страви, а також список товарів у кожній категорії (рис Б.1). Для кожного товару відображається назва та ціна. Користувач має можливість додавати товари до кошика для подальшого оформлення замовлення. Третій розділ - кошик. У цьому розділі відображаються обрані товари, які користувач планує замовити (рис Б.2). Також передбачені поля для введення імені користувача та коментаря до замовлення. Після заповнення цих даних користувач може підтвердити створення замовлення. Четвертий розділ - замовлення. Тут користувач може переглядати свої створені замовлення, а також їх поточний статус (рис Б.2). Наприклад, замовлення може бути у стані «готується» або «завершено». Крім

статусу, відображається склад замовлення, тобто перелік товарів, які були замовлені. Навігаційна модель клієнтського додатку у вигляді діаграми станів та переходів між сторінками наведена на рисунку 2.5.

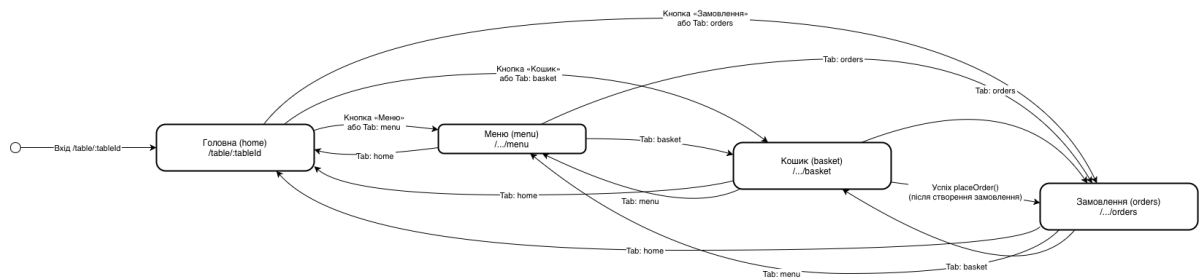


Рисунок 2.5. Діаграма станів та переходів у онлайн меню

Додаток розроблений як простий та зручний у використанні інтерфейс без зайвого функціоналу. Основний акцент зроблено на швидкість роботи та зрозумілість для користувача. Інтерфейс адаптований під мобільні пристрої, оскільки передбачається, що більшість користувачів будуть працювати з ним зі смартфонів.

2.7 Розробка адміністративної панелі

Адміністративна панель реалізована у вигляді вебдодатку, який використовується працівниками закладу для управління системою. Даний додаток надає доступ до основного функціоналу, пов'язаного з налаштуванням меню, обробкою замовлень, управлінням закладами та користувачами.

У панелі передбачено декілька основних розділів. Головна сторінка використовується для швидкого перегляду загальної інформації та навігації по системі (рис А.1). Вона дозволяє швидко перейти до інших розділів та отримати базове уявлення про стан системи.

Розділ меню призначений для управління товарами та категоріями (рис А.2). У ньому можна створювати, редагувати та видаляти категорії, а також додавати товари, змінювати їх характеристики та ціни. Це дозволяє гнучко налаштовувати меню для кожного закладу.

Розділ замовлень використовується для перегляду та обробки замовлень (рис А.3). Працівники можуть бачити список замовлень, їх склад та поточний статус. Також є можливість змінювати статус замовлення, наприклад позначати його як «готується» або «завершено».

У розділі закладів здійснюється управління мережею закладів (рис А.5). Тут можна створювати нові заклади, редагувати їх інформацію, а також налаштовувати столи. Для кожного столу може бути згенерований QR-код, який використовується клієнтами для доступу до меню.

Розділ користувачів призначений для управління обліковими записами працівників (рис А.6). У ньому можна переглядати список користувачів, призначати їм ролі та керувати доступом до функціоналу системи. У системі реалізовано механізм контролю доступу. Для кожного користувача визначається роль, яка впливає на доступ до функціоналу. Обмеження застосовуються не лише на рівні розділів адміністративної панелі, а й на рівні окремих дій. Це означає, що користувач може мати доступ, наприклад, до перегляду даних, але не мати можливості їх змінювати. Такий підхід дозволяє гнучко налаштовувати права доступу та забезпечує безпечну роботу системи.

Розділ аналітики дозволяє переглядати дані про роботу закладу (рис А.4). У ньому відображається інформація про кількість проданих товарів та отриманий прибуток. Дані представлені у вигляді діаграм, що спрощує їх аналіз.

Окрім основних розділів, реалізовано сторінки для роботи з обліковими записами. Користувач може зареєструватися та увійти в систему за допомогою сторінок реєстрації та логіну. Після входу в систему користувач потрапляє на сторінку онбордингу. На ній пропонується або приєднатися до існуючого закладу за запрошенням, або створити власний заклад. Це дозволяє зручно розпочати роботу в системі.

Реалізовано механізм реєстрації за запрошенням. Користувач отримує спеціальне посилання, за яким переходить до сторінки реєстрації. На цій сторінці він самостійно вводить свої дані та встановлює пароль. Це підвищує зручність і безпеку створення облікових записів.

Адміністративна панель підтримує зміну кольорової теми, що дозволяє користувачам змінити її на світлу і темну. Крім цього, інтерфейс адаптований під різні типи пристроїв, зокрема персональні комп'ютери, планшети та мобільні телефони. Це забезпечує зручну роботу з системою незалежно від розміру екрана та умов використання.

Висновки до розділу 2

У даному розділі було розглянуто процес проєктування та розробки вебдодатку для ресторанного бізнесу. Було обрано технологічний стек, спроектовано архітектуру системи та структуру бази даних. Також визначено основні модулі системи та їх взаємодію між собою. Реалізовано серверну частину з API та механізмами контролю доступу, а також клієнтський додаток у вигляді QR-меню й адміністративну панель для управління системою. Розроблені модулі забезпечують роботу із замовленнями, меню, користувачами та закладами. У результаті створено систему, яка дозволяє автоматизувати основні процеси ресторанного закладу та забезпечує зручну взаємодію як для відвідувачів, так і для персоналу.

РОЗДІЛ 3

ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

3.1 Тестування функціональності вебдодатку

Після завершення основної розробки було проведено тестування функціональності вебдодатку з метою перевірки коректності роботи всіх його складових. Основна увага приділялась перевірці ключових можливостей системи як для клієнтської частини, так і для адміністративної панелі, а також взаємодії між ними.

Тестування клієнтської частини (QR-меню) включало перевірку відображення меню, коректності завантаження даних із сервера та загальної роботи інтерфейсу на різних типах пристроїв. Було перевірено, що користувач може перейти за QR-кодом, отримати доступ до меню конкретного закладу та столу, переглянути категорії і товари, а також додавати позиції до кошика без помилок. Особлива увага приділялась зручності взаємодії та швидкості відгуку інтерфейсу, оскільки додаток орієнтований на використання з мобільних пристроїв. Також було протестовано процес створення замовлення, включаючи введення даних користувача та передачу інформації на сервер.

В адміністративній панелі було перевірено роботу всіх основних розділів системи. Зокрема, протестовано створення, редагування та видалення товарів і категорій, зміну цін, а також налаштування доступності товарів для різних закладів. Окремо перевірялась робота з замовленнями, включаючи їх перегляд, зміну статусів та відображення актуальної інформації. Було впевнено встановлено, що всі зміни, виконані в адміністративній частині, коректно відображаються у клієнтському додатку, що підтверджує правильну синхронізацію даних.

Також проведено тестування механізмів авторизації та розмежування доступу. Перевірено, що користувачі можуть виконувати лише ті дії, які дозволені їх роллю, а обмеження доступу коректно працюють як на рівні

окремих розділів, так і на рівні дій. Це дозволило переконатися у безпечності роботи системи та захисті від несанкціонованих змін.

Важливою частиною тестування була перевірка взаємодії між клієнтом і сервером. Було протестовано роботу API, обробку HTTP-запитів та передачу даних у форматі JSON. У процесі тестування перевірялось, що система коректно реагує на дії користувача, а всі запити обробляються без помилок. Окремо виконувалась перевірка обробки некоректних або неповних даних, наприклад при введенні неправильних значень або відсутності обов'язкових полів. У таких випадках система повинна коректно повідомляти про помилку та не допускати некоректних операцій.

За результатами проведеного тестування було встановлено, що основний функціонал вебдодатку працює стабільно та відповідає поставленим вимогам. Виявлені незначні помилки були усунені в процесі доопрацювання та не впливають на загальну працездатність і зручність використання системи.

3.2 Аналіз продуктивності системи

Після перевірки функціональності було проведено аналіз продуктивності вебдодатку з метою оцінки швидкодії та стабільності його роботи. Основна увага приділялась швидкості завантаження клієнтської частини, обробці запитів на сервері та ефективності роботи бази даних. Перевірка виконувалась у наближених до реальних умовах використання, зокрема під час перегляду меню, створення замовлень, а також додавання і редагування даних через адміністративну панель.

У процесі аналізу було встановлено, що клієнтська частина працює достатньо швидко. Середній час першого завантаження сторінки становить приблизно 1-2 секунди, залежно від швидкості інтернет-з'єднання. Подальша навігація між розділами відбувається значно швидше - в межах 200-500 мілісекунд, оскільки оновлюються лише необхідні частини сторінки без повного перезавантаження. Взаємодія з інтерфейсом є плавною, затримки при додаванні товарів у кошик або переході між категоріями практично відсутні.

Серверна частина показала стабільну роботу при обробці запитів. Середній час відповіді API становить близько 100-300 мілісекунд для стандартних операцій, таких як отримання списку товарів або створення замовлення. При виконанні більш складних операцій, наприклад обробки великої кількості даних в адміністративній панелі, час відповіді може збільшуватись до 400-600 мілісекунд, що залишається прийнятним для користувача. Система коректно обробляє одночасні запити від декількох користувачів без помітного зниження швидкодії.

Окрему увагу було приділено роботі бази даних. Операції зчитування даних виконуються в середньому за 50-150 мілісекунд, а операції запису - до 200 мілісекунд. Це дозволяє забезпечити швидку обробку замовлень і оновлення інформації в системі. Навіть при одночасному виконанні декількох запитів база даних працює стабільно, без втрати цілісності даних.

Також варто зазначити, що використання контейнеризації за допомогою Docker забезпечує стабільність роботи системи незалежно від середовища запуску. Це дозволяє уникнути проблем, пов'язаних із різницею в налаштуваннях, та спрощує процес розгортання додатку [13].

У результаті проведеного аналізу можна зробити висновок, що вебдодаток має достатній рівень продуктивності для використання у реальних умовах. Система забезпечує швидку обробку даних, стабільну роботу та може ефективно масштабуватись у разі збільшення навантаження.

3.3 Оцінка зручності використання

Оцінка зручності використання базується на здатності системи забезпечувати ефективну, зрозумілу та просту взаємодію користувача з інтерфейсом. Згідно з підходом Дж. Нільсена, основними критеріями зручності використання є навченість, ефективність використання, запам'ятовуваність, рівень помилок та загальна задоволеність користувача [28]. Під час розробки вебдодатку значна увага приділялась забезпеченню високого рівня зручності використання, оскільки система орієнтована як на відвідувачів закладу, так і на

персонал. Інтерфейс побудований таким чином, щоб користувач міг швидко зрозуміти принцип роботи системи без додаткового навчання або інструкцій. Основні сценарії взаємодії, зокрема перегляд меню, додавання товарів до кошика та оформлення замовлення, виконуються за 2-3 дії, що дозволяє зменшити час взаємодії з системою та знизити когнітивне навантаження на користувача.

Клієнтська частина (QR-меню) має мінімалістичний інтерфейс без зайвих елементів. Після сканування QR-коду користувач одразу потрапляє на сторінку з меню конкретного столу без необхідності реєстрації або введення додаткових даних. Категорії товарів відображаються у вигляді списку, що дозволяє швидко перемикається між ними, а додавання товару до кошика відбувається в один клік. У кошику користувач може змінювати кількість позицій та бачити загальну суму замовлення. Процес оформлення замовлення включає лише введення імені та коментаря, що значно спрощує використання.

Адміністративна панель також спроектована з урахуванням зручності роботи користувачів. Основні розділи (меню, замовлення, заклади, користувачі) доступні через навігаційне меню, що дозволяє швидко переходити між ними без зайвих дій. Таблиці з даними підтримують швидке редагування та оновлення інформації, а зміна статусу замовлення виконується без перезавантаження сторінки. Додатково реалізовано перевірки введених даних та повідомлення про помилки, що допомагає уникати некоректних дій.

У результаті можна зробити висновок, що інтерфейс системи є зрозумілим і зручним для користувачів. Використання простих сценаріїв взаємодії, мінімальної кількості дій та адаптації під різні пристрої дозволяє ефективно працювати з додатком як відвідувачам, так і персоналу закладу.

3.4 Перспективи розвитку та масштабування системи

Розроблений вебдодаток можна розширювати та доповнювати новими можливостями. Одним із найбільш очевидних покращень є додавання онлайн-оплати. У майбутньому можна реалізувати оплату замовлень прямо в QR-меню через такі сервіси, як Monobank Pay, Apple Pay або Google Pay. Це дозволить

користувачу не тільки зробити замовлення, а й одразу його оплатити без виклику офіціанта.

Також систему можна розширити в плані зовнішнього вигляду та налаштувань для закладів. Наприклад, додати можливість змінювати кольори, стиль меню, порядок відображення категорій або показувати додаткову інформацію. Окремо можна реалізувати відображення інформації про заклад, такої як фотографії, опис та графік роботи. Це зробить QR-меню більш інформативним і схожим на повноцінний сайт закладу.

З точки зору масштабування система вже підтримує різні варіанти використання. Вона може працювати як для одного закладу, так і для мережі з декількох закладів. При цьому один і той самий користувач може мати різні ролі в різних закладах, наприклад бути адміністратором в одному і офіціантом в іншому. Це зручно для невеликих бізнесів або підприємців, які керують кількома закладами. Важливою перевагою є те, що система не має жорстко заданої логіки під один сценарій роботи. Більшість функціоналу налаштовується, а структура дозволяє додавати нові можливості без значних змін у коді. Це означає, що при зростанні кількості користувачів або розширенні бізнесу систему можна легко допрацьовувати та масштабувати.

Підсумовуючи, вебдодаток вже зараз є гнучким у використанні та має зрозумілі напрямки для подальшого розвитку, що робить його придатним як для невеликих закладів, так і для більших мереж.

Висновки до розділу 3

У даному розділі було проведено тестування вебдодатку та оцінено його ефективність у реальних умовах використання. Перевірка функціональності показала, що всі основні можливості системи працюють коректно як у клієнтській частині, так і в адміністративній панелі, а взаємодія між ними відбувається без помилок. Аналіз продуктивності підтвердив, що додаток має достатню швидкодію: сторінки завантажуються швидко, запити обробляються без значних затримок, а система стабільно працює при одночасному виконанні

декількох операцій. Оцінка зручності використання показала, що інтерфейс є простим і зрозумілим для користувачів, дозволяє швидко виконувати необхідні дії та не потребує додаткового навчання. Також було визначено основні напрямки подальшого розвитку, зокрема додавання онлайн-оплати, розширення налаштувань та покращення клієнтської частини. Загалом можна зробити висновок, що розроблена система є працездатною, зручною у використанні та має потенціал для подальшого розвитку і масштабування.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено вебдодаток для ресторанного бізнесу з використанням технологій ASP.NET Core та Vue.js, який дозволяє автоматизувати процес обслуговування клієнтів і управління закладом. У першому розділі було проведено аналіз предметної області та існуючих рішень, зокрема систем електронного меню та управління закладом. Визначено їх основні функціональні можливості, переваги та обмеження, а також сформульовано функціональні та нефункціональні вимоги до розроблюваної системи. На основі проведеного аналізу було обрано технологічний стек для реалізації вебдодатку.

У другому розділі виконано проєктування та реалізацію вебдодатку. Було спроектовано архітектуру системи та структуру бази даних, а також реалізовано основні модулі системи. Серверна частина побудована у вигляді REST API, клієнтська частина реалізує QR-меню для відвідувачів, а адміністративна панель забезпечує управління товарами, замовленнями, закладами та користувачами.

У третьому розділі проведено тестування функціональності вебдодатку, аналіз продуктивності та оцінку зручності використання системи. Отримані результати підтвердили стабільну роботу додатку, достатню швидкодію та простоту взаємодії для користувачів. Також визначено можливі напрямки подальшого розвитку, зокрема інтеграцію онлайн-оплати та розширення налаштувань.

У результаті виконання кваліфікаційної роботи досягнуто поставленої мети – створено вебдодаток для ресторанного бізнесу, який може бути використаний у реальних умовах та має потенціал для подальшого розвитку і масштабування. Мета досягнута, завдання дослідження – виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Martin, R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Boston: Prentice Hall, 2009. 464 p.
2. Sommerville, I. Software Engineering. 10th ed. Boston: Pearson, 2016. 816 p.
3. Fowler, M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002. 533 p.
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 1994. 395 p.
5. Freeman, E., Robson, E. Head First Design Patterns. Sebastopol: O'Reilly, 2020. 694 p.
6. ChoiceQR. Онлайн QR меню для закладів [Електронний ресурс]. 2025. URL: <https://choiceqr.com/uk/> (дата звернення: 02.05.2026).
7. SMART SPOT - інтерактивне QR-меню для ресторанів [Електронний ресурс]. 2025. URL: <https://expertsolution.com.ua/statti/interaktyvne-qr-menu-smart-spot/> (дата звернення: 02.05.2026).
8. Expirenza by mono [Електронний ресурс]. 2026. URL: <https://expz.monobank.ua> (дата звернення: 02.05.2026).
9. Vue.js Documentation [Електронний ресурс]. 2026. URL: <https://vuejs.org/> (дата звернення: 02.05.2026).
10. ASP.NET Core Documentation [Електронний ресурс]. 2026. URL: <https://learn.microsoft.com/aspnet/core> (дата звернення: 02.05.2026).
11. Entity Framework Core Docs [Електронний ресурс]. 2026. URL: <https://learn.microsoft.com/ef/core> (дата звернення: 02.05.2026).
12. PostgreSQL Documentation [Електронний ресурс]. 2026. URL: <https://www.postgresql.org/docs/> (дата звернення: 02.05.2026).
13. Docker Documentation [Електронний ресурс]. 2026. URL: <https://docs.docker.com/> (дата звернення: 02.05.2026).

14. Vite Documentation [Електронний ресурс]. 2026. URL: <https://vitejs.dev/> (дата звернення: 02.05.2026).
15. Axios GitHub [Електронний ресурс]. 2026. URL: <https://github.com/axios/axios> (дата звернення: 02.05.2026).
16. BCrypt Documentation [Електронний ресурс]. 2026. URL: <https://github.com/kelektiv/node.bcrypt.js> (дата звернення: 02.05.2026).
17. MDN Web Docs [Електронний ресурс]. 2026. URL: <https://developer.mozilla.org/> (дата звернення: 02.05.2026).
18. W3C HTML Standard [Електронний ресурс]. 2026. URL: <https://html.spec.whatwg.org/> (дата звернення: 02.05.2026).
19. CSS Specification [Електронний ресурс]. 2026. URL: <https://www.w3.org/Style/CSS/> (дата звернення: 02.05.2026).
20. REST API Tutorial [Електронний ресурс]. 2026. URL: <https://restfulapi.net/> (дата звернення: 02.05.2026).
21. QR Code Generator Guide [Електронний ресурс]. 2026. URL: <https://www.qr-code-generator.com/> (дата звернення: 02.05.2026).
22. Restaurant Management Systems Overview // TechJournal. 2023.
23. POS Systems Explained [Електронний ресурс]. 2026. URL: <https://www.shopify.com/pos> (дата звернення: 02.05.2026).
24. CRM Systems Overview [Електронний ресурс]. 2026. URL: <https://www.salesforce.com/crm/> (дата звернення: 02.05.2026).
25. OWASP Top 10 [Електронний ресурс]. 2026. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 02.05.2026).
26. JWT Introduction [Електронний ресурс]. 2026. URL: <https://jwt.io/introduction> (дата звернення: 02.05.2026).
27. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Київ, 2015.
28. Nielsen J. Usability Engineering. Boston: Academic Press, 1993. 272 p.

29. Збірник матеріалів Всеукраїнської конференції молодих учених "Інформаційні технології" [Електронний ресурс]. 2026. URL: <https://zcit.kubg.edu.ua/index.php/journal/issue/view/16>.

30. Valiveti S. S. Evolution of ASP.NET to ASP.NET Core: Tools, Strategies, and Implementation Approaches // 2025 IEEE 2nd International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS). - Bangalore, India, 2025. - DOI: 10.1109/ICITEICS64870.2025.11341480

31. Malokhvii E., Buhai V., Molchanov H., Chernykh O. Аналітичний огляд та порівняння сучасних JavaScript рішень для розробки веб-додатків // Системи управління, навігації та зв'язку. - 2021. - Т. 4, № 66. - С. 55. - DOI: <https://doi.org/10.26906/SUNZ.2021.4.055>

ДОДАТОК А

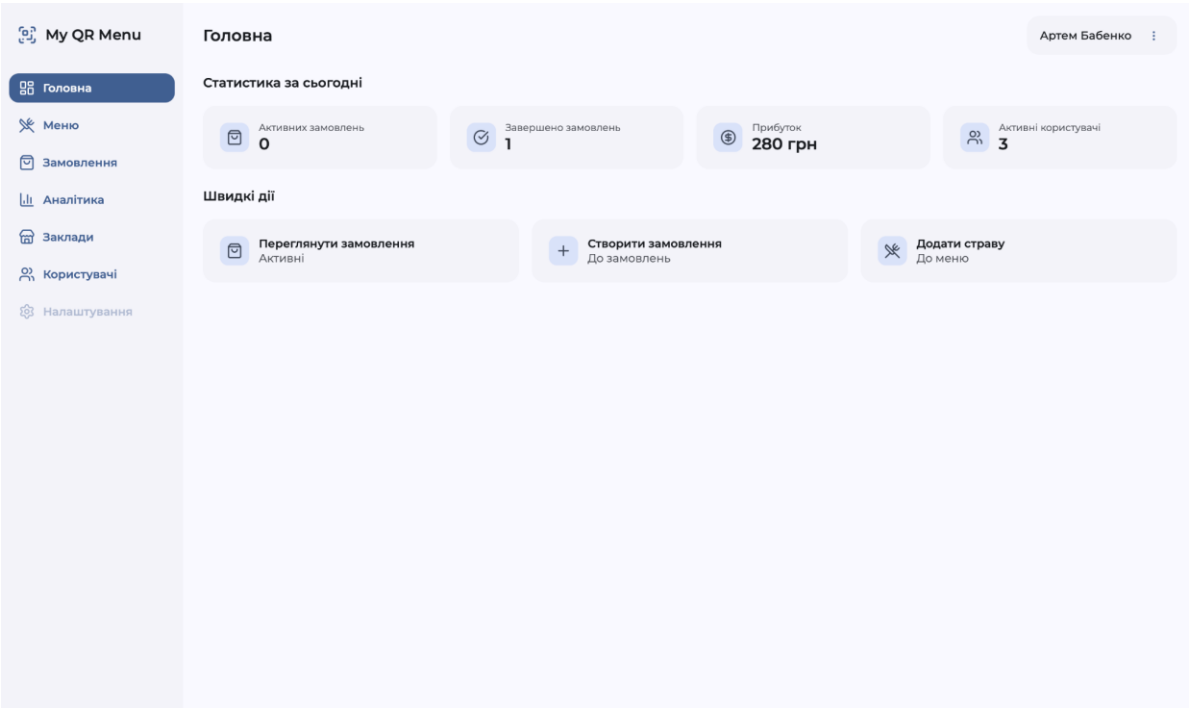


Рисунок А.1. Головна сторінка з швидкими діями

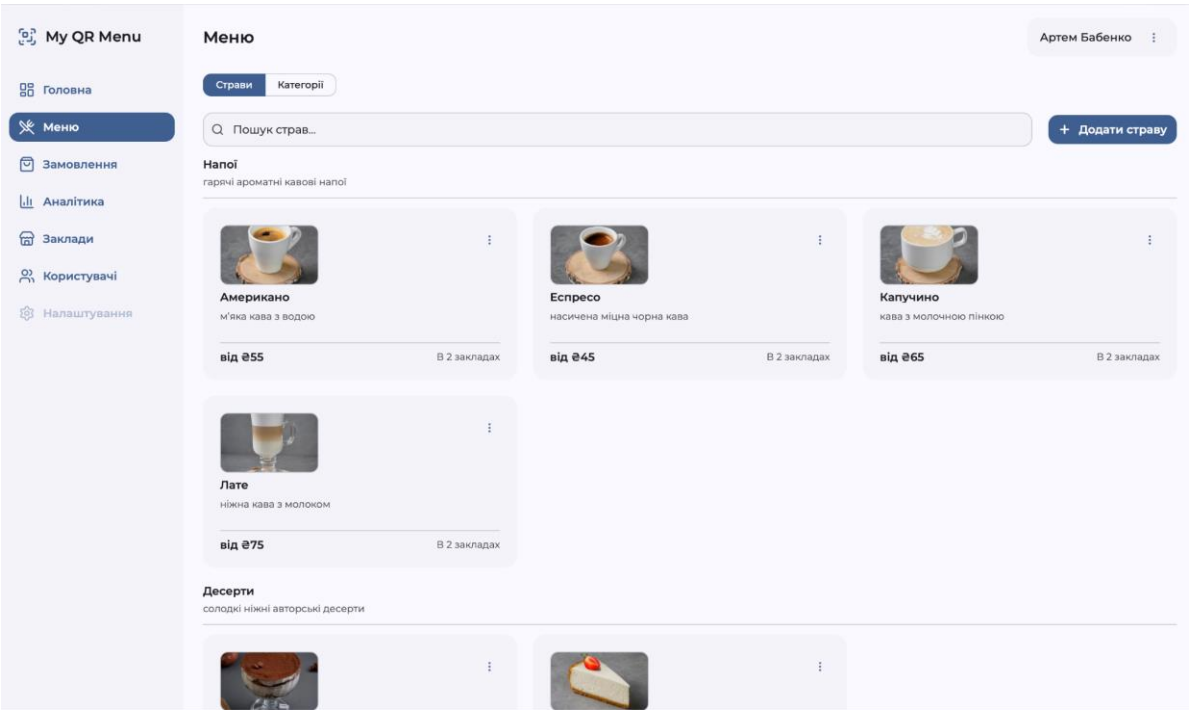


Рисунок А.2. Сторінка налаштування меню та категорій

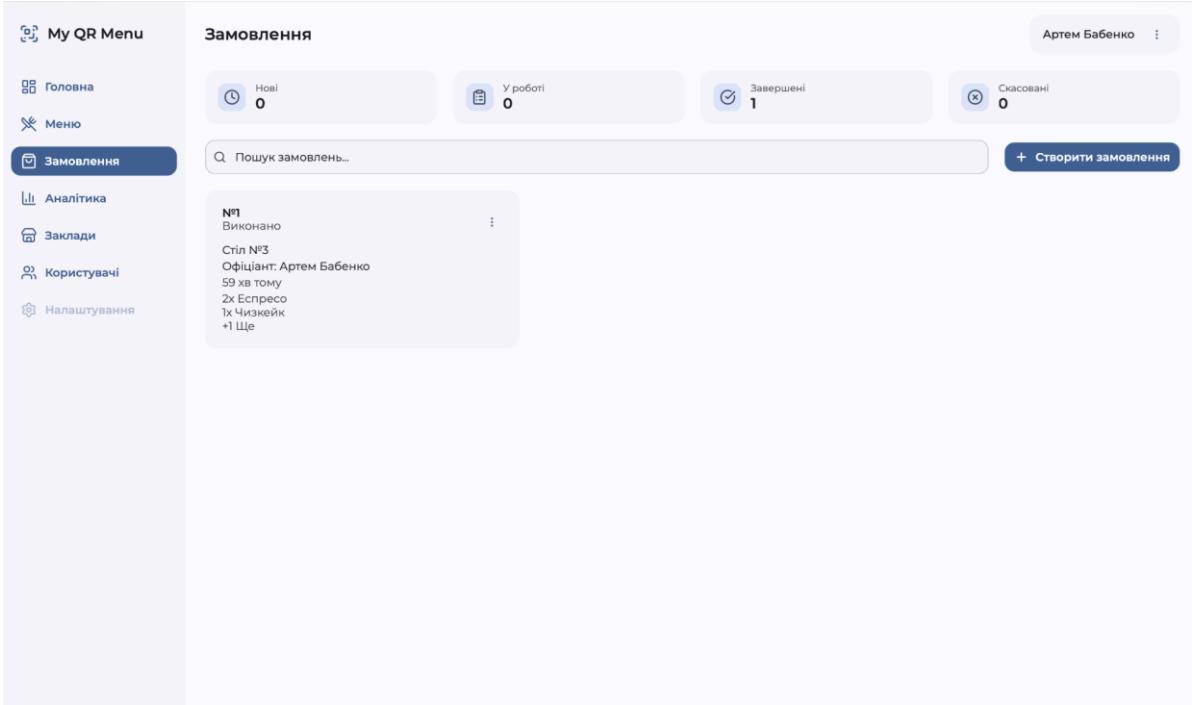


Рисунок А.3. Сторінка замовлень

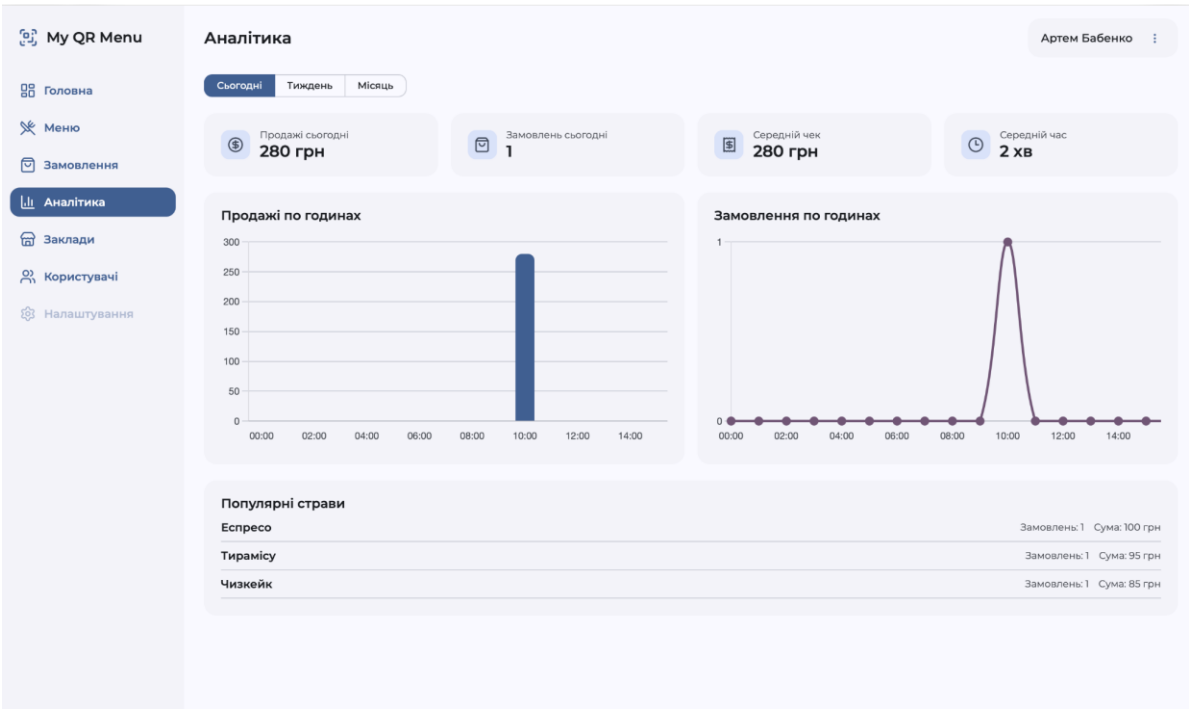


Рисунок А.4. Сторінка аналітики з діаграмами та показниками прибутку

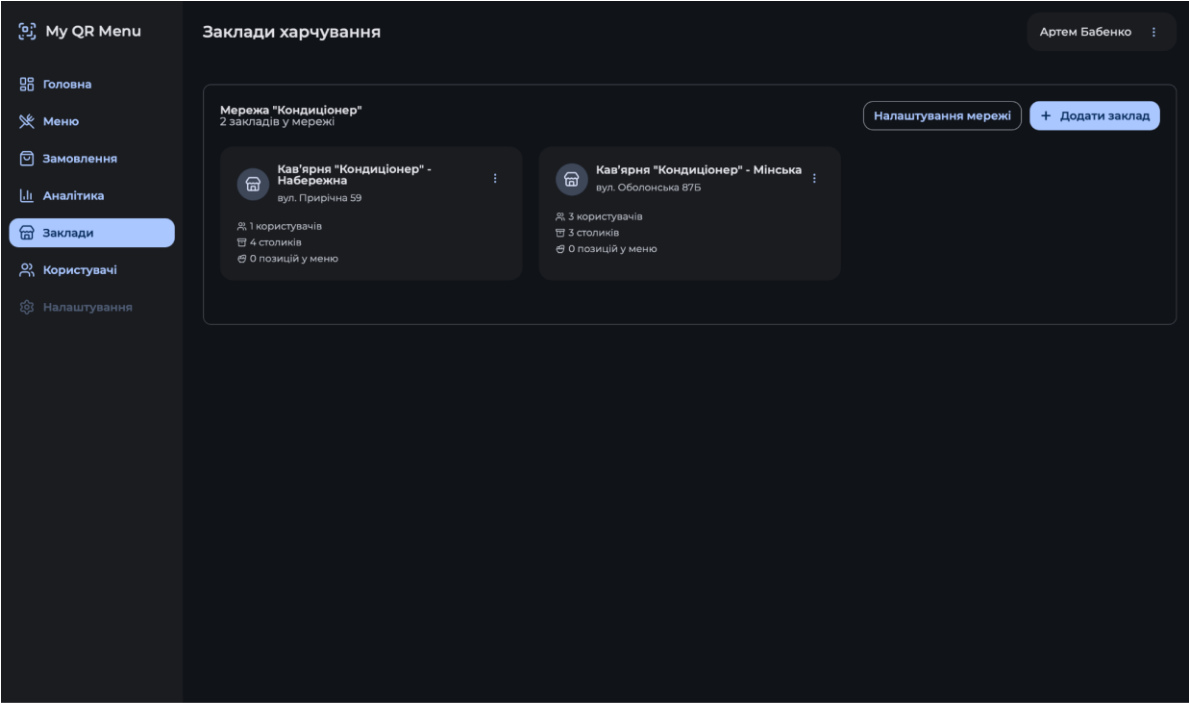


Рисунок А.5. Сторінка налаштування мереж та закладів харчування

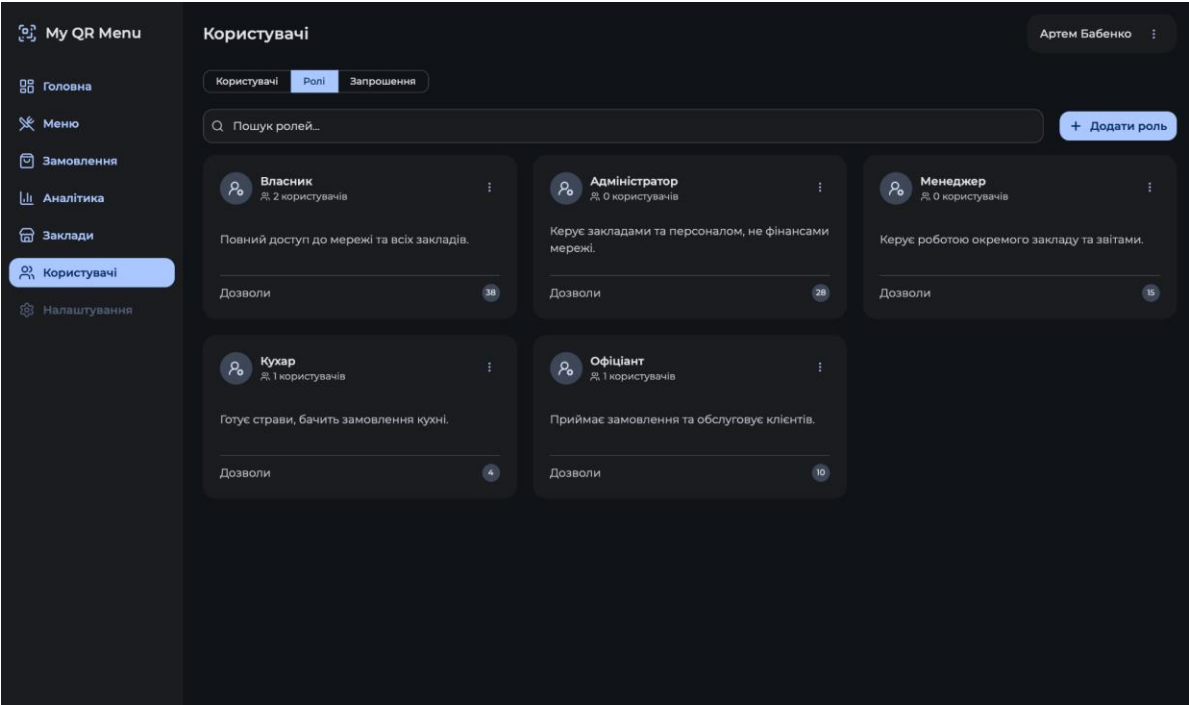


Рисунок А.6. Сторінка адміністрування користувачів і ролей

ДОДАТОК Б

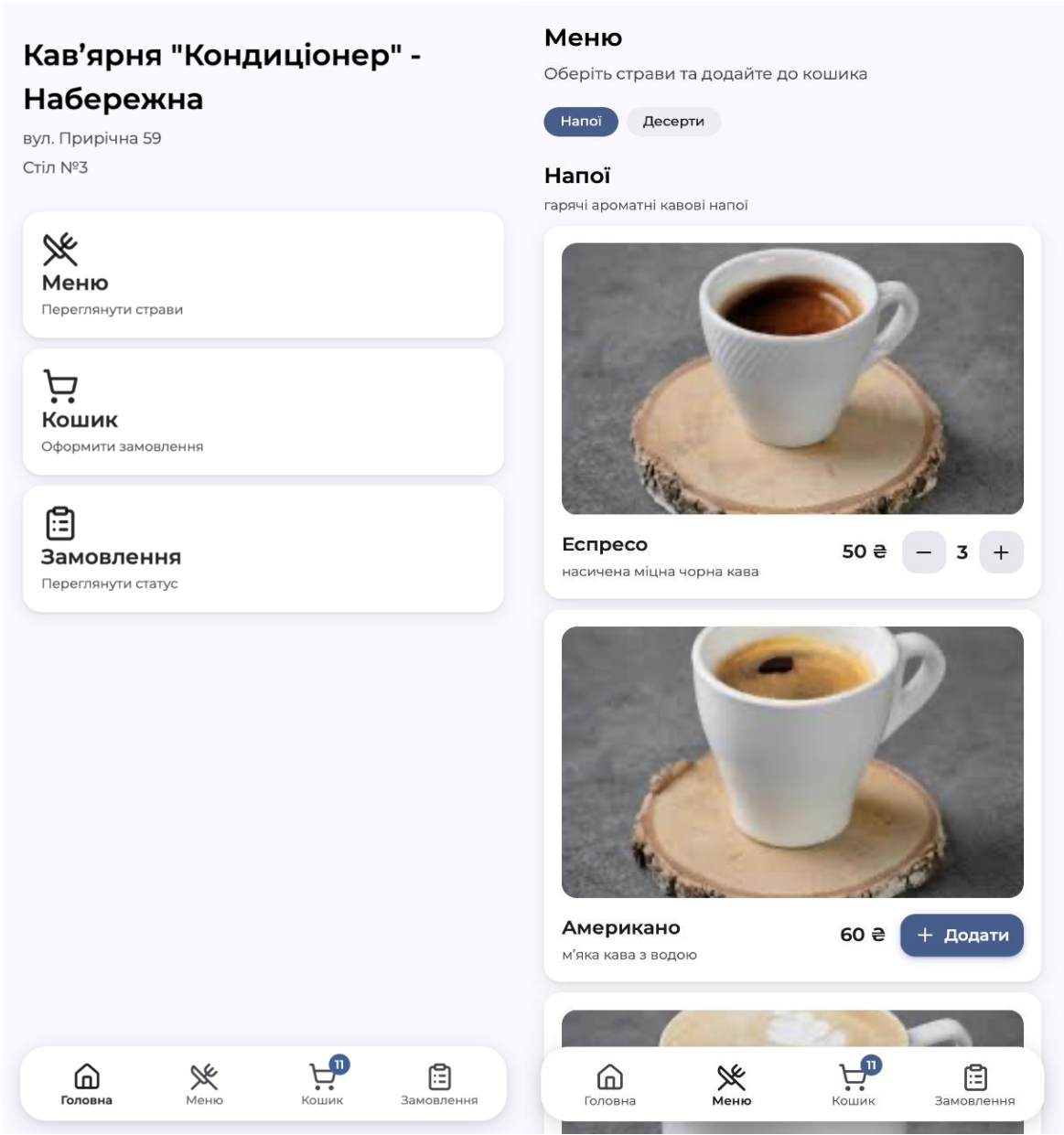


Рисунок Б.1. “Головна” сторінка та сторінка “Меню”

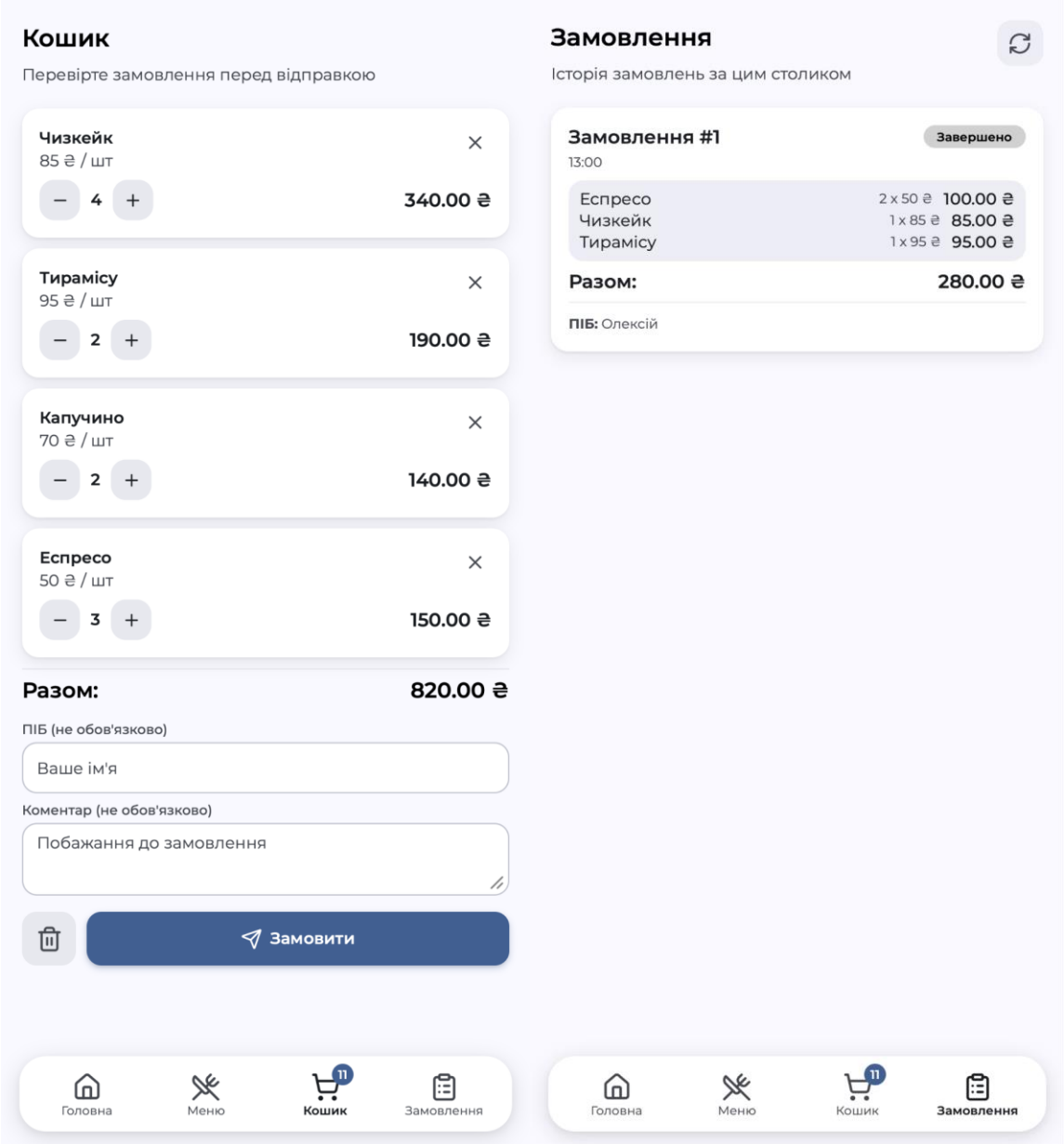


Рисунок Б.2. Сторінка “Кошик” та сторінка “Замовлення”