КВ-11 Чебан Максим

Комп’ютерна система управління мережею торгових точок

Доповідь

**Слайд 1**

Титульний слайд

**Слайд 2**

У сучасних умовах активного розвитку торгівлі та цифрових технологій ефективне управління мережею торгових точок стає важливою складовою успішної діяльності підприємств. Зростаючий обсяг даних, необхідність швидкого прийняття рішень та контроль за процесами продажу вимагають впровадження сучасних комп’ютерних систем обліку та управління.

**Слайд 3**

Метою даного проєкту є створення комп’ютернрої системи для централізованого управління мережею торгових точок, що дозволяє вести облік товарів, продажів, категорій, ролей користувачів, а також здійснювати базовий аналіз діяльності.

Вимоги до комп’ютернрої системи, що розробляється:

* Віконний інтерфейс користувача на базі .NET Windows Forms;
* Робота з базою даних PostgreSQL;
* Можливість створення, редагування та перегляду товарів, категорій товарів, торгових точок, користувачів, ролей та продажів.
* Підтримка обліку залишків товару;
* Аутентифікація користувачів;
* Фільтрація, сортування та пошук даних;
* Створення базових звітів.

**Слайд 4**

Системи даного типу використовуються для автоматизації процесів обліку, продажів, управління персоналом і формування звітності, що особливо важливо для мережевого бізнесу. Найпопулярнішими рішеннями на ринку є: Poster POS, 1С:Підприємство, RetailCRM, iiko, Square POS, Lightspeed, Shopify POS. Незважаючи на широкий функціонал, більшість із них мають недоліки: високу вартість, складність налаштування, обмежену гнучкість та залежність від сторонніх платформ. У зв’язку з цим є потреба у створенні комп’ютерної системи управління мережею торгових точок, яка має бути простою у використанні, масштабованою, економічно ефективною та здатною адаптуватися до конкретних потреб малого й середнього бізнесу.

**Слайд 5**

Архітектура комп’ютерної системи управління мережею торгових точок має модульну, багаторівневу структуру, що забезпечує розділення відповідальностей, гнучкість у розробці та легкість масштабування.

Основні компоненти архітектури:

1. Клієнтський застосунок (Front-end)

2. Серверна частина (Back-end)

3. База даних (Data Layer)

**Слайд 6**

Для функціонування системи необхідна база даних для збереження інформації. Розроблена реляційна база даних відповідає нормальним формам і дозволяє ефективно зберігати, оновлювати та обробляти дані, пов’язані з торговельною діяльністю. У системі реалізовано вісім основних таблиць: Users (користувачі системи), Roles (ролі користувачів), Stores (торгові точки), Products (товари), Categories (категорії товарів), Sales (продажі), SaleProducts (деталізація товарних позицій у кожному продажу) та Inventory (залишки товарів у магазинах). Кожна таблиця має чітко визначені поля, типи даних і зовнішні ключі, що забезпечують зв’язки між сутностями

**Слайд 7**

Основною мовою програмування обрано C#, що забезпечує високу продуктивність і широкі можливості як для клієнтської, так і для серверної частини. У якості системи управління базами даних використовується PostgreSQL — потужна реляційна СУБД з відкритим кодом, яка добре масштабується, підтримує складні транзакції та має чудову інтеграцію з .NET-технологіями. Для створення клієнтського застосунку обрано Windows Forms (WinForms). Серверна частина реалізована на ASP.NET Core, сучасному фреймворку з підтримкою RESTful API, асинхронної обробки запитів, а також ORM-технології Entity Framework Core, що дозволяє працювати з базою даних через об’єкти.

**Слайд 8**

Структура проєкту організована відповідно до принципів чистої архітектури, що дозволяє розмежувати логіку за відповідними компонентами. Контролери (Controllers) відповідають за обробку HTTP-запитів. Контекст бази (DatabaseContext) забезпечує зв'язок з базою даних через ORM Entity Framework Core. Компонент Migrations використовується для керування змінами у структурі бази без втрати даних. У каталозі Models зберігаються класи, що відображають структуру таблиць бази. Бізнес-логіка зосереджена в класах Services, а їхні контракти визначені в ServicesContracts, що забезпечує гнучкість і легкість розширення функціоналу. Конфігураційний файл appsettings.json містить дані підключення до бази та інші параметри середовища, а основний файл запуску Program.cs відповідає за реєстрацію сервісів, підключення middleware-компонентів і ініціалізацію вебзастосунку.

**Слайд 9**

Клієнтський застосунок розроблено за допомогою Windows Forms (WinForms), що забезпечує створення інтуїтивно зрозумілого та функціонального інтерфейсу для роботи з даними. До основних компонентів інтерфейсу відносяться: Form, MenuStrip, ToolStrip, Button, Label, TextBox, ComboBox, DataGridView та інші.

Інтерфейс передбачає 7 основних пунктів меню, що активують відповідні форми для роботи з категоріями, товарами, магазинами, продажами, користувачами, ролями, а також для експорту звітів.

Обмін даними між клієнтською та серверною частинами системи здійснюється за допомогою HTTP-запитів до API, реалізованого на ASP.NET Core. Для цього застосовуються окремі класи-репозиторії (наприклад, CategoryRepository, SalesRepository), які інкапсулюють логіку взаємодії з API через HttpClient..

**Слайд 10**

Основні інтерфейси реалізовані за шаблоном CRUD-операцій. Для кожної сутності — категорій, продуктів, продажів, інвентаря, торгових точок, користувачів — передбачено відповідні форми для перегляду, створення, оновлення та видалення записів. Додатково реалізовано динамічне оновлення форм: при виборі пункту меню відображаються лише відповідні елементи інтерфейсу, а всі інші — приховуються. Це покращує зручність користування та забезпечує мінімалізм.

**Слайд 11**

Клієнтська частина програмного забезпечення реалізує механізм формування звітів у форматі html та .xlsx, що дозволяє користувачеві отримати зведену інформацію про проведені продажі у структурованому вигляді.

Таким чином, функціональність генерації звітів у вигляді Excel-документів дозволяє отримати гнучке, зрозуміле й придатне для архівації представлення даних, що значно розширює аналітичні можливості програмного забезпечення.

**Слайд 12**

У межах тестування клієнтської частини системи було перевірено основні функціональні можливості, реалізовані у Windows Forms-застосунку. Тестування охоплювало операції додавання, редагування, перегляду та видалення даних, а також взаємодію з інтерфейсом прикладного програмування (API).

У межах тестування серверної частини системи було перевірено роботу прикладного інтерфейсу програмування (API), що є ключовою ланкою у взаємодії клієнтської частини з базою даних. Тестування проводилось за допомогою вбудованого інструменту Swagger, який забезпечує зручну візуалізацію, перевірку та документування REST-запитів. Було здійснено перевірку всіх основних типів HTTP-запитів (GET, POST, PUT, DELETE).

**Слайд 13**

У ході дипломного проєкту розроблено комп’ютерну систему для централізованого управління мережею торгових точок, яка автоматизує ключові бізнес-процеси — від обліку товарів до звітності. Проведено аналіз існуючих рішень і виявлено їхні недоліки, що спонукало створити власний продукт, орієнтований на малий та середній бізнес в Україні. Проєкт базується на клієнт-серверній архітектурі з використанням C#, ASP.NET Core, WinForms і PostgreSQL. Реалізовано функціонал авторизації, ролей, звітності у форматах HTML і XLSX, а також забезпечено масштабованість і гнучкість системи. Проведено тестування клієнтської частини та API.

У перспективі планується розширення функціоналу системи за рахунок переходу на мікросервісну архітектуру, інтеграції з сучасними засобами безпеки та впровадження аналітики користувацької поведінки. Також розглядається інтеграція з хмарними сервісами для підвищення надійності, масштабованості та ефективності роботи з великими обсягами даних.

Таким чином, система є сучасним, гнучким інструментом, готовим до подальшого розвитку і адаптації під реальні потреби бізнесу.Дякую за увагу

**Слайд 14**

Дякую за увагу