

МОДЕЛЮВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ СКЛАДОМ МЕДИЧНИХ ТОВАРІВ

Вінницький національний технічний університет

Анотація

Розглянуто підходи до моделювання та розробки програмного забезпечення для управління складом медичних товарів. Проаналізовано основні алгоритми оптимізації обліку, відстеження запасів та автоматизації постачань. Досліджено ефективність методів прогнозування попиту, інтеграції з системами постачальників та використання хмарних технологій. Визначено універсальні підходи, що забезпечують стабільність і точність роботи програмних рішень в умовах динамічних змін медичного ринку.

Ключові слова: управління складом, медичні товари, програмне забезпечення, алгоритмічна оптимізація, автоматизація.

Abstract

Approaches to modeling and software development for managing the inventory of medical products are considered. The main algorithms for optimizing accounting, inventory tracking, and supply automation are analyzed. The effectiveness of demand forecasting methods, integration with supplier systems, and the use of cloud technologies are studied. Universal approaches are identified that ensure the stability and accuracy of software solutions in the conditions of dynamic changes in the medical market.

Keywords: warehouse management, medical supplies, software, algorithmic optimization, automation.

Вступ

Розробка швидкого програмного забезпечення, яке може забезпечити своєчасну логістику, оптимізацію запасів і точний облік, є необхідною для автоматизації процесів управління складом медичних товарів. Інформаційні технології дозволяють створювати системи, що інтегруються з постачальниками, щоб відстежувати переміщення товарів у режимі реального часу та зменшити ймовірність нестачі або надлишку товару. Розробка алгоритмів, які адаптуються до змін у попиті, забезпечують безперервність постачання та відповідають нормативним вимогам, є необхідною завдяки зростаючим вимогам до медичних установ.

Баланс між швидкістю обробки запитів, ефективністю використання ресурсів і точністю даних є однією з найважливіших проблем у сфері складського управління. Прогнозування попиту, оптимізація маршрутів постачання та автоматизація обліку підвищують продуктивність складських операцій. Але такі методи потребують механізмів контролю якості даних, інтеграції з медичними системами та гнучких алгоритмів синхронізації.

Розробка архітектури, яка дозволяє адаптивне масштабування, динамічне оновлення даних і захист від збоїв, є важливою для забезпечення надійності роботи програмних рішень. Створення програмного забезпечення, що здатне до автономної адаптації та ефективного управління ресурсами, необхідно для зменшення впливу зовнішніх факторів, таких як зміни в регуляторі, варіативність постачань і необхідність швидкої реакції на надзвичайні ситуації.

Архітектурні підходи до оптимізації продуктивності системи управління складом медичних товарів

Проектування ефективної архітектури є ключовим фактором для забезпечення стабільної роботи програмного забезпечення, що керує складськими операціями. Використання принципів чистої архітектури дозволяє створити гнучку та масштабовану систему, яка чітко розділяє рівні відповідальності та спрощує підтримку коду. Завдяки такій структурі можна ізолювати бізнес-логіку від зовнішніх залежностей, що підвищує стійкість до змін у базі даних, зовнішніх сервісах та користувацьких інтерфейсах [1].

Одним із основних викликів у системах управління складом є балансування навантаження між операціями читання та запису. Використання патерну Command Query Responsibility Segregation дозволяє розділити операції отримання даних та модифікації стану системи, що значно знижує навантаження на основну базу даних та підвищує швидкість обробки запитів.

Для подальшого масштабування та розподілу навантаження застосовується подієво-орієнтована архітектура, що дозволяє реагувати на зміни стану системи в режимі реального часу. Впровадження асинхронних черг повідомлень, допомагає уникнути блокувань у високонавантажених процесах та забезпечує стійкість до пікових навантажень [2].

Продуктивність та відмовостійкість системи

Забезпечення високої продуктивності та відмовостійкості системи управління складом медичних товарів є ключовим завданням для ефективного функціонування всіх бізнес-процесів. Для досягнення цієї мети застосовуються різні підходи до масштабування, оптимізації обчислень та управління ресурсами.

Одним із критичних аспектів є масштабування системи, яке може здійснюватися як горизонтально, так і вертикально. Горизонтальне масштабування передбачає розподіл навантаження між кількома серверами, що дозволяє ефективно обробляти велику кількість одночасних запитів. Вертикальне масштабування, у свою чергу, полягає в покращенні продуктивності окремих серверів за рахунок збільшення їхньої обчислювальної потужності [3].

Для зменшення навантаження на базу даних впроваджується кешування даних, зокрема, використання Redis. Це дає змогу швидко отримувати інформацію про залишки медикаментів, історію постачань і статуси замовлень без необхідності виконання складних SQL-запитів. Крім того, застосовується реплікація бази даних, що забезпечує її відмовостійкість і дозволяє продовжувати роботу системи навіть у разі збою одного з вузлів.

Крім цього, важливим елементом є автоматичний моніторинг системи за допомогою таких інструментів, як Prometheus. Це дозволяє виявляти можливі збої в роботі сервісів та оперативно реагувати на них, мінімізуючи час простою системи. У разі критичних відмов передбачені механізми автоматичного перезапуску сервісів та аварійного відновлення.

Ще одним аспектом є оптимізація продуктивності бази даних за допомогою індексації, шардінгу та використання ефективних SQL-запитів. Це дозволяє зменшити час виконання операцій та підвищити швидкість обробки інформації [4].

Загалом, впровадження всіх зазначених методів забезпечує стабільну роботу системи навіть за умов високого навантаження, а також дозволяє уникнути критичних збоїв, що можуть призвести до втрати важливих даних або порушення складських операцій.

Інструментальні засоби розробки

Ці засоби допоможуть досягти ефективної розробки та впровадження програмного забезпечення для управління складом медичних товарів. Використовуючи сучасні інструменти моделювання, управління базами даних, серверні технології та засоби контролю версій, можна забезпечити надійність, продуктивність і масштабованість системи.

Інструменти для аналізу та моделювання Visual Paradigm є одним з найкращих CASE-засобів, який дозволяє автоматизувати майже весь цикл моделювання та проєктування. Він підтримує створення UML-діаграм, аналіз моделей та генерацію коду.

Microsoft Visio допоможе у моделюванні бізнес-процесів та їх оптимізації. Для ER-моделювання баз даних використовується SQL Server Management Studio, що полегшує розробку та адміністрування баз даних.

Системи управління базами даних Microsoft SQL Server є високопродуктивною реляційною системою керування базами даних, яка підтримує складні запити, реплікацію та масштабованість.

Серверні технології та API C# є сучасною мовою програмування, що використовується для розробки веб-додатків і бекенду. ASP.NET Core – це високопродуктивний фреймворк для створення веб- і хмарних додатків. Swagger – інструмент для документування та тестування REST API, що спрощує інтеграцію та розробку веб-сервісів.

Управління проєктами Jira використовується для гнучкого управління проєктами, дозволяючи планувати, відстежувати та аналізувати процеси розробки. GitHub є популярною платформою для контролю версій, яка підтримує спільну розробку та автоматизацію розгортання.

Висновок

Використання зазначених інструментів дозволить ефективно моделювати бізнес-процеси, керувати базами даних, розробляти API та забезпечувати контроль версійності коду. Це сприятиме успішному створенню надійної системи управління складом медичних товарів, що відповідає сучасним вимогам програмного забезпечення.

Поєднання чистої архітектури та подієво-орієнтованого підходу дозволяє створити надійну та масштабовану систему управління складом медичних товарів, що здатна ефективно обробляти велику кількість запитів, швидко реагувати на зміни та адаптуватися до зовнішніх умов.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Clean Architecture With .NET. Pearson Education, 2023.
2. Garofolo E. Practical Microservices: Build Event-Driven Architectures with Event Sourcing and CQRS. Pragmatic Programmers, LLC, The, 2019. 250 p.
3. Shore J., Warden S. Art of Agile Development. O'Reilly Media, Incorporated, 2021.
4. Iglés G. Relational Database Practices Book : Temporal Database: Relational Database and Transact Sql. Independently Published, 2021.

Глусенко Анастасія Сергіївна – студентка групи 2ПІ-24М, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: anastasiagluskoo7@gmail.com

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Hlusenka Anastasiia Sergiivna – student of the 2PI-24M group, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, Ukraine, email: anastasiagluskoo7@gmail.com

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua