

УДК 004.77:72.054

DOI: 10.31673/2412-9070.2023.024853

О. М. ГАРАГАН, студент;

Б. Ю. ЖУРАКОВСЬКИЙ, доктор техн. наук, професор,

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ

СИСТЕМА СКЛАДСЬКОГО ПРИМІЩЕННЯ НА БАЗІ ІОТ

У статті розглянуто варіант поліпшення автоматизації складського приміщення завдяки впровадженню Інтернету речей. Систему спростковано з огляду на результати аналізу дослідження предметної сфери та аналізу вже наявних на ринку варіантів керування, що уможливило виокремлення ключових моментів для впровадження у власну систему керування. Створено прототип системи на основі IoT-давачів, даючи змогу ефективно збирати і передавати дані в досліджуваній архітектурі.

Запропонованим програмним забезпеченням можуть користуватись як компанії та підприємства, зосереджені на зберіганні та логістиці, так і поодинокі підприємці, котрі бажають автоматизувати складські процеси.

Ключові слова: IoT; складське приміщення; сервіс; керування; давач; база даних.

ВСТУП

Складське господарство є конкурентним фактором у будь-якому ланцюзі постачань і відіграє ключову роль у забезпеченні взаємодії між усіма партнерами. Для забезпечення ефективного функціонування ланцюга постачань потрібно мати ефективне керування складськими ресурсами та їх раціональне розподілення. Надійна система керування складськими приміщеннями може знизити витрати та підвищити задоволеність клієнтів. Однак із розвитком ринку та зростанням вимог до ефективності та швидкості обслуговування традиційні моделі систем керування [1] складом стають застарілими та неефективними. Тому компанії шукають інноваційні підходи та технології для оптимізації складських операцій.

Однією з таких інноваційних технологій є Інтернет речей. Він забезпечує зв'язок між кількома фізичними об'єктами та генерує велику кількість даних у реальному часі [2]. Ці дані можуть бути перетворені на корисну інформацію, що допомагає в керуванні та ухваленні рішень.

Аналіз дослідження. Застосування Інтернету речей відкриває нові можливості для оптимізації складських операцій. Ця концепція систем, розроблених через підімкнення великої кількості інтелектуальних пристроїв у мережу, допомагає створювати інноваційні складські приміщення та підвищувати рівень автоматизації.

Метою дослідження є розроблення системи керування складськими приміщеннями на основі Інтернету речей для оптимізації складських операцій та підвищення ефективності логістичних процесів.

ОСНОВНА ЧАСТИНА

Загальний огляд об'єкта дослідження

Складські витрати становлять до 5% [3] від собівартості продажів компанії, а за умов сучасної конкуренції на світовому ринку організації прагнуть підвищити рентабельність активів, тому оптимізація складських витрат стала першочерговим завданням для бізнесу.

Загалом зрозуміло, що використання систем керування складом на основі ручного керування для оновлення всіх повсякденних операцій є складною задачею. Автоматизувати процес можна завдяки IoT-компонентам [4], проте великі компанії, які мають значну кількість ліній замовлень і одиниць зберігання, можуть використовувати більш комплексний підхід та вдаватися до спеціалізованих систем керування — системи керування складом *Warehouse Management System (WMS)* [5].

WMS може бути спеціально розроблена або адаптована для відповідності конкретним потребам організації. Інтеграція підімкнених пристроїв і давачів у продуктах і матеріалах є революційним кроком у способі виробництва і доставляння товарів [6]. Давачі IoT можуть стежити за складом у реальному часі, надсилати повідомлення щодо потреби у поповненні запасів або автоматично розподіляти ресурси для виконання цього завдання [7].

Усі ці можливості, включно зі збором, обробленням та аналізом даних, стали значно доступнішими і широко застосовними в бізнес-секторі завдяки розвитку технології Інтернету речей. Сучасні IoT-рішення пропонують потужні аналітичні інструменти та платформи керування даними, що роблять використання цих даних ефективним та оптимальним.

Архітектура системи

Систему спроектовано за допомогою методу «верхнього вниз». Це підхід до проектування, який починається з формулювання загальних цілей системи на високому рівні, а потім розкладає їх на більш конкретні підсистеми або компоненти. Такий процес продовжується, дроблячи кожен підсистему на ще більш специфічні елементи, доти, доки не буде досягнуто достатнього рівня деталізації для виконання кожного компонента. Основні переваги методу «верхнього вниз» полягають у здатності ефективно керувати проектом, розбиваючи великий проєкт на менші, керовані частини. Це також сприяє більш ранньому виявленню та розв'язанню проблем в процесі проектування [8].

Отже, на основі ретельного оцінювання сучасних застосунків можна дійти висновку, що основні компоненти, потрібні для функціонування системи, такі: IoT-складові, центральний сервер, інтерфейс користувача.

Схему основних, а також додаткових компонентів та зв'язків між ними зображено на рис. 1.

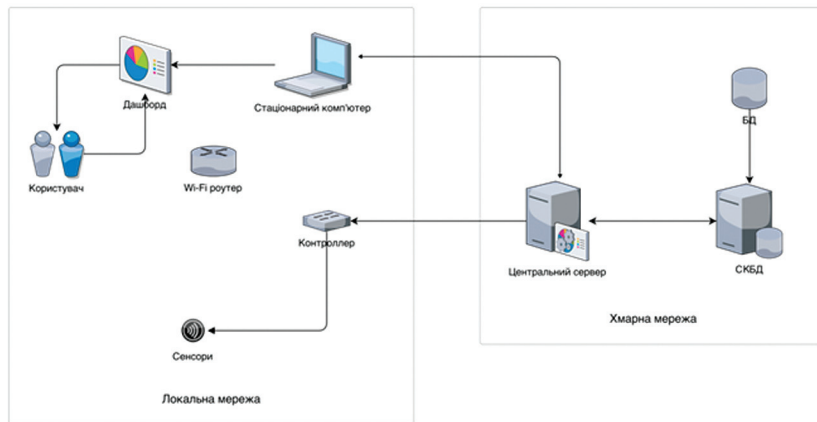


Рис. 1. Загальна схема архітектури системи

Центральний сервер відповідає за оброблення запитів від вебінтерфейсу і взаємодію зі СКБД. Він може виконувати різні дії, зокрема отримання, оновлення, видалення або вставлення даних у базу даних.

СКБД є програмним засобом, який використовується для створення, керування та маніпулювання базами даних. Він відповідає за зберігання, організацію та відтворення даних за запитом.

Давачі та сенсори застосовуються для збору даних, які потім відправляються до центрального сервера через контролер для оброблення.

Після отримання даних від пристроїв централізований сервіс аналізує ці дані, виконує необхідні обчислення або трансформації, а потім передає результати на панель аналітики.

Структура системи

Після опису основних складових архітектури системи варто розробити також детальну структуру системи. Основні елементи цієї структури містять компоненти, які відповідають за передавання та отримання даних. Кожен із цих компонентів має важливе значення в загальній роботі системи і є частиною більшого механізму взаємодії.

Важливим елементом є комунікація, або точніше, протоколи, застосовні для обміну інформацією між компонентами. Вибір відповідного протоколу може значно вплинути на ефективність та надійність системи, тому цей вибір потребує обережного розгляду.

Розуміння того, як компоненти взаємодіють та співпрацюють, є ключовим для ефективного проектування системи, що забезпечує її рівну та стабільну роботу (рис. 2).

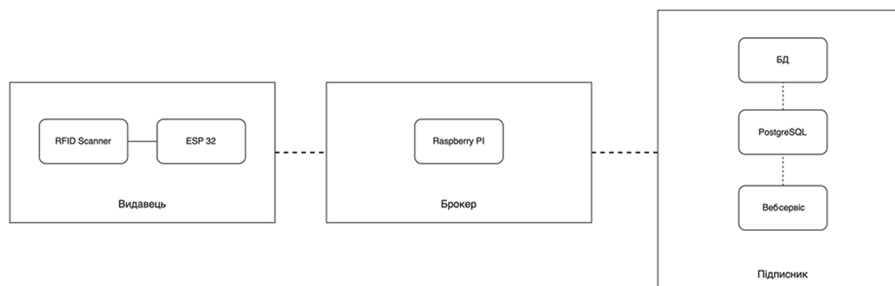


Рис. 2. Загальна схема структури системи

Збір даних

Керування складом має розширений спектр можливостей, який охоплює збір, оброблення і передавання інформації різного типу. Цю інформацію може бути взято з різних джерел, включно зі сканерами штрих-кодів та RFID-позначок, а також давачами, встановленими в різних точках виробничого процесу, які можуть збирати дані про різноманітні параметри (рис. 3).

Оскільки RFID-технологія добре себе зарекомендувала на ринку і є перевіреним варіантом збору інформації в подібних системах [9], було вирішено приділити увагу саме цьому методу. Дійсно, RFID, а також штрих-коди широко використовуються в сучасному складському обліку для ідентифікації та відстеження товарів. Загалом, вибір між штрих-кодами та RFID часто залежить від специфіки операцій, бюджету та потреб конкретного бізнесу. Проте розвиток технологій і масштабованість процесів все частіше змушує більшу кількість аналогових методів поступатися сучасним.

RFID RC522. RC522 — це популярний модуль RFID, який використовується в безконтактних системах зв'язку. Він працює на частоті 13,56 МГц і підтримує режим ISO/IEC 14443 A/MIFARE. RC522 може як зчитувати, так і записувати RFID-позначки [10]. Він взаємодіє з позначками, які відповідають стандарту ISO/IEC 14443 типу A. Модуль зв'язується з мікроконтролером або одноплатовим комп'ютером (наприклад, Arduino чи ESP32) за допомогою послідовного периферійного інтерфейсу SPI (Serial Peripheral Interface).

ESP32. ESP32 — це серія недорогих, малопотужних мікроконтролерів системи на кристалі з вбудованим Wi-Fi і дворежимним Bluetooth. ESP32 є популярним вибором завдяки своїм безпроводовим можливостям, високій швидкості процесора та широкому спектру апаратних периферійних пристроїв. Його можна використовувати в різних застосунках, включно з пристроями «розумного будинку», дистанційними давачами, домашньою автоматизацією та багатьма іншими рішеннями IoT [11]. Однак він також досить універсальний, щоб його застосовувати в різних інших застосунках вбудованих систем [12].

Архітектура програмного забезпечення. Архітектура програмного забезпечення охоплює набір рішень для проектування загальної структури та поведінки системи. Вона має вирішальне значення для визначення загального вигляду системи, а також того, як вона поводить себе в різних ситуаціях. Архітектура допомагає зацікавленим сторонам зрозуміти та оцінити, як система досягає функцій, зокрема безпеки, доступності і можливості модифікації [13].

Мова програмування JavaScript. JavaScript — це популярна мова програмування для вебрішень і її використання для програмування IoT-пристроїв є логічним продовженням цієї тенденції. Особливості JavaScript, зокрема Node.js, керування пам'яттю, подієво-орієнтоване програмування, простота впровадження, а також наявність великої кількості бібліотек і фреймворків, роблять цю мову програмування гарним вибором для IoT-розробників.

Платформа Node.js. Node.js особливо корисна у сфері Інтернету речей, даючи змогу ефективно обробляти множинні запити, що генеруються множинними давачами, маячками, передавачами та електродвигунами. За допомогою Node.js можна налагодити операції «запит – відповідь» і виконувати їх швидше, ніж за традиційними методами [14].

Додатково Node.js підтримує використання сокетів та протоколу MQTT, що забезпечує неперервне передавання даних в IoT-застосунках. Це важливо для високопродуктивних IoT-систем, де потрібно передавати великі обсяги даних без затримок.

Angular. Angular, створений та активно підтримуваний гігантом індустрії Google, є одним із найбільш популярних та потужних фреймворків для розроблення вебзастосунків [15]. Angular також відомий своїми потужними вбудованими інструментами та функціями. Це передбачає такі можливості, як двостороннє прив'язування даних, що забезпечує автоматичну синхронізацію між моделями даних та їх візуальним поданням.

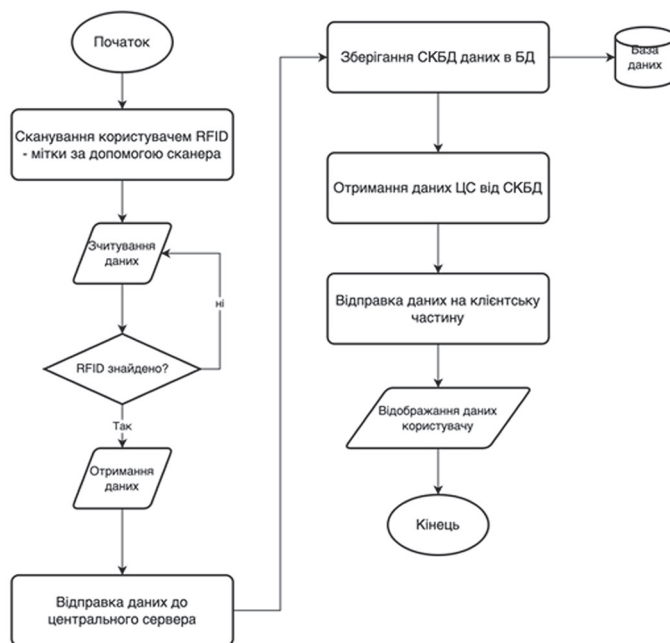


Рис. 3. Схема алгоритму передавання даних

PostgreSQL. Вимоги до архітектури Інтернету речей охоплюють ефективну, надійну і масштабовану систему баз даних, яка має здатність обробляти значний обсяг різнорідних потоків даних від різних пристроїв IoT. PostgreSQL, як реляційна база даних із відкритим вихідним кодом, відповідає цим вимогам завдяки своїм внутрішнім функціям, гнучкості та сумісності з вимогами IoT [16].

Розроблення серверної і клієнтської частини

Серверну частину було розроблено на Node.js, яка є платформою JavaScript, що працює на сервері. Node-postgres працює з будь-якою версією PostgreSQL й активно підтримується та оновлюється. Він також підтримує проміси, що робить його сумісним із сучасними асинхронними патернами у Node.js, такими як `async/await`.

Для оброблення HTTP-запитів від клієнтської частини був використаний Express.js, який є популярним фреймворком для Node.js. Для комунікації з IoT-пристроями застосовувався MQTT [9]. Клієнтську частину було розроблено на Angular, який використовує TypeScript, що є строго типізованою версією JavaScript, даючи змогу створювати більш надійний код.

Розроблення бази даних

База даних є важливим компонентом у будь-якій системі, вона слугує центральним репозиторієм для зберігання та керування інформацією як про систему в цілому, так і про окремі її елементи (рис. 4). Вона є організованим набором даних, що підтримують структурований доступ, і часто використовуються вбудовані мови програмування, зокрема DDL/DCL, для керування структурою та режимом доступу до даних.

Не менш важливу роль відіграє забезпечення безпеки збереженої інформації в базі даних. Це особливо актуально, оскільки база даних не тільки зберігає інформацію про роботу пристроїв та системи, а й обробляє велику кількість даних користувачів, включно з конфіденційною інформацією.

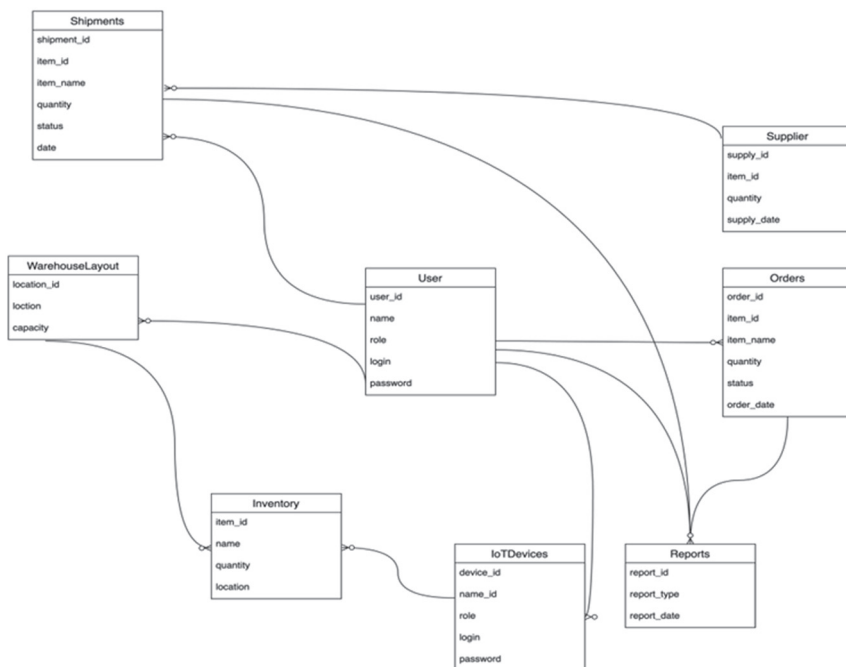


Рис. 4. ER-діаграма бази даних

Розроблення сторінок вебзастосунку

Реєстрація користувачів здійснюється через адміністратора системи. Це дає можливість контролювати, хто має доступ до системи, що особливо важливо для організацій, які працюють із конфіденційною інформацією, оскільки такий підхід допомагає унеможливити небажану реєстрацію третіми особами.

Загальний алгоритм реєстрації зображено на рис. 5.

Для згаданої раніше ролі адміністратора має бути власна панель, доступ до якої буде лише у користувача відповідної ролі.



Рис. 5. Схема алгоритму входу в систему

Адміністратор складської IoT-системи може мати такі можливості: моніторинг стану та керування пристроями IoT (рис. 6), аналіз даних, керування сповіщеннями, керування доступом.



Рис. 6. Схема взаємодії адміністратора з пристроями

Однією з головних функцій сервісу є моніторинг процесів і даних у режимі реального часу. Цю функцію забезпечує система логування, яка реєструє всі зміни, що відбуваються в робочій системі. Великий обсяг потрібної інформації про дані, що цікавлять користувача, можна зручно переглянути за вибраний період часу. Крім того, для кожної ролі передбачено візуалізацію даних із певним рівнем доступу.

Особливу увагу приділено відображенню даних, що стосуються типових складських процесів: отримання, інвентаризація, зберігання, відвантаження тощо. Ці показники є основою для моніторингу продуктивності та ефективності на складі, тому їх розміщено на головній сторінці дашборду. Крім того, на цій сторінці можна спостерігати за заповненістю складу та моніторити час виконання завдань працівниками на кожному етапі.

ВИСНОВКИ

У статті розглянуто систему керування складськими приміщеннями на основі Інтернету речей для оптимізації складських операцій та підвищення ефективності логістичних процесів.

Здійснено детальне проектування IoT-системи. Проведено глибокий аналіз структури системи, включно з компонентами, що відповідають за передавання та отримання даних. У межах розроблення системи використано низку технічних вирішень, зокрема використання Node.js та СКБД PostgreSQL для розроблення серверної частини, а також формування клієнтської частини. Розроблено інтерфейс сервісу разом із процесом реєстрації користувачів та їх авторизації в системі. Для забезпечення безпеки даних реєстрація користувачів має здійснюватися через адміністратора системи, що дає змогу контролювати доступ до системи.

У результаті створено систему, яка дає можливість ефективно керувати складом, забезпечує відстеження товарів та збір даних у реальному часі, а також дозволяє персоналу взаємодіяти із системою через інтуїтивно зрозумілий інтерфейс.

Список використаної літератури

1. Жураковский Б. Ю. Об'єктно-орієнтована модель системи керування мережею NGN // Вісник Держ. ун-ту інформаційно-комунікаційних технологій. 2012. Вип. 10, № 3. С. 81–84.
2. *Methods of the Objects Identification and Recognition Research in the Networks with the IoT Concept Support* / O. Shevchenko, A. Bondarchuk, O. Polonevych [et al.] // *Paper Proceedings of the Selected Papers on Publishing Papers with CEUR-WS co-located with Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021)*, 2923. 2021. P. 277–282. URL: <http://ceur-ws.org/Vol-2923/>
3. *Warehouse Rental Costs Expected to Increase by 5% in 2022 – CARGONOW*. CARGONOW – Connecting the Logistics Eco-System. URL: <https://cargonow.world/warehouse-rental-costs-expected-to-increase-by-5-in-2022/>
4. *Optimization Algorithms of Smart City Wireless Sensor Network Control* / M. Moshchenko, B. Zhurakovskiy, V. Poltorak [et al.] // *Paper Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems II. Vol. II co-located with International Conference on Problems of Infocommunications. Science and Technology (PICST 2021)*, Kyiv, Ukraine, October 26. 2021. Vol. 3188. P. 32–42. URL: <http://ceur-ws.org/Vol-3188/paper4.pdf>
5. *Smart Warehouses: How IoT Improves Warehouse Management* — Digiteum. Digiteum. URL: <https://www.digiteum.com/internet-of-things-for-smart-warehouses/>

6. **IoT in Warehouse Management: 6 Ways To Transform Operations.** Hopstack | *Fulfillment Automation and Visibility for Warehouse Management*. URL:
<https://www.hopstack.io/blog/iot-can-transform-warehouse-operations>
7. **Michaelis J.** *Warehouse Insights – The planning and optimization application for modern warehouses.* All Blog Posts / SAP Community. URL:
<https://blogs.sap.com/2020/04/16/warehouse-insights-the-planning-and-optimization-application-for-modern-warehouses/>
8. **Software System for Processing and Visualization of Big Data Arrays** / N. Fedorova, Y. Havrylko, A. Kovalchuk [et al.] // *ICCSEEA 2022: Advances in Computer Science for Engineering and Education*, P. 324–336. Part of the *Lecture Notes on Data Engineering and Communications Technologies* book series (LNDECT, Vol. 134).
9. **Жураковський Б. Ю., Зенів І. О.** *Технології інтернету речей: навч. посіб. [Електронний ресурс]* // КПІ ім. Ігоря Сікорського. 2021. 503 с. URL:
<https://ela.kpi.ua/handle/123456789/42078>
10. **Features of processing signals from stationary radiation sources in multi-position radio monitoring systems** / V. Druzhynin, S. Toliupa, O. Pliushch [et al.] // *CEUR Workshop Proceedings, 2020. Vol. 2746*. P. 46–65. URL:
<http://ceur-ws.org/Vol-2746/>.
11. **ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems.** *Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems*. URL:
<https://www.espressif.com/en/products/socs/esp32>
12. **Raspberry Pi Documentation.** *Raspberry Pi*. URL:
<https://www.raspberrypi.com/documentation/>
13. **Что такое MQTT?** – Описание протокола MQTT – AWS. *Amazon Web Services, Inc.* URL:
<https://aws.amazon.com/ru/what-is/mqtt/>
14. **Documentation** | *Node.js*. *Node.js*. URL:
<https://nodejs.org/en/docs>
15. **Angular.** *Angular*. URL:
<https://angular.io/docs>
16. **PostgreSQL: Documentation.** *PostgreSQL: The world's most advanced open source database*. URL:
<https://www.postgresql.org/docs/>

O. Garagan, B. Zhurakovskiy

WAREHOUSE SYSTEM BASED ON IoT

This paper considers the option of improving the automation of a warehouse by introducing the Internet of Things. The system was designed taking into account the results of the analysis of the subject area research and the analysis of control options already available on the market, making it possible to identify key points for implementation in our control system. This made it possible to create a prototype system based on IoT sensors, which allows for efficient data collection and transmission in the proposed architecture.

The created software can be used by companies and enterprises engaged in storage and logistics and individual entrepreneurs who want to automate warehouse processes.

Keywords: IoT; warehouse; service; management; sensor; database.

