

# **ЗАСТОСУВАННЯ ВІРТУАЛЬНОГО ПОМІЧНИКА В ІНТЕЛЕКТУАЛЬНІЙ СИСТЕМІ МАРКУВАННЯ ТА ОБЛІКУ МАТЕРІАЛЬНИХ РЕСУРСІВ**

Вінницький національний технічний університет

## **Анотація**

*У статті розглянуто приклад впровадження віртуального помічника в інтелектуальну систему маркування та обліку матеріальних ресурсів. Основна мета розробки – спростити процес управління інформацією та підвищити ефективність облікових операцій за допомогою автоматизації. Для інтеграції віртуального помічника використано Telegram Bot API, що забезпечує швидкий і зручний доступ до інформації через популярну платформу для обміну повідомленнями. Особливу увагу приділено розробці структури реляційної бази даних, яка слугує основою для функціонування системи. Таблиці бази даних створені на основі вхідних даних та оптимізовані для ефективної взаємодії з ботом, що дозволяє користувачам оперативної отримувати необхідну інформацію та оновлювати дані про ресурси. Система має потенціал для застосування в різних сферах, де важливо забезпечити точний контроль за матеріальними ресурсами.*

**Ключові слова:** віртуальний помічник, чат-бот, Telegram, баркод, інформація.

## **Вступ**

Віртуальні помічники стрімко розвиваються, надаючи користувачам усе більше можливостей. Завдяки прогресу в розпізнаванні мовлення та нейролінгвістичному програмуванню (методи, що допомагають впливати на думки та поведінку людей) їх здатність розуміти й виконувати запити значно зростає. З подальшим вдосконаленням технології розпізнавання голосу, роль віртуальних помічників у бізнес-процесах буде лише посилюватися [1].

Віртуальний помічник (AI-помічник або цифровий асистент) – це програмне забезпечення, яке сприймає голосові команди на природній мові та виконує завдання від імені користувача. Серед поширених типів таких помічників виділяють: чат-ботів зі штучним інтелектом, розмовних агентів та ШІ-віртуальних асистентів [2].

## **Результати дослідження**

Інтелектуальні віртуальні помічники здатні виконувати широкий спектр завдань на різних пристроях і платформах. Працюючи з користувачами, вони використовують певні алгоритми для обробки та надання інформації. Завдяки таким помічникам, користувачі можуть швидко отримувати доступ до потрібної інформації, керувати розумною технікою в домі та організовувати свої календарі через інтеграцію з різними сервісами та додатками, тощо.

Запуск ChatGPT, інтелектуального віртуального помічника та мовної моделі на базі штучного інтелекту від OpenAI, викликав обговорення щодо майбутнього віртуальних помічників. Хоча є багато сподівань та припущень про можливий вплив ChatGPT на цей ринок, поки залишається незрозумілим, до яких наслідків у подальшому розвитку галузі це призведе [2]. Окрім віртуальних помічників від OpenAI, на даний момент користуються популярністю віртуальні помічники у якості ботів в месенджері Telegram. Telegram.Bot — це найпопулярніший клієнт .NET для Telegram Bot API, що дозволяє розробникам створювати ботів для програми обміну повідомленнями Telegram.

Для того щоб реалізувати інтеграцію існуючої реляційної бази даних з ERP-системи [3] до Telegram бота необхідно виконати ряд пунктів [4]:

1. Створити Telegram бота (бот-канал);
2. Додати користувача до бот-канала;
3. Інтегрувати в Telegram бот таблиці інформаційної системи

Інтеграція виконується певним функціоналом, зокрема описуємо назву бота та прописуємо посилання на нього:

```
l_bot_name := datasource_dm.GetMethodName4Code(p_dsmttd_code => Barcode_BOT);  
  
l_link := 'https://telegram.me/' || l_bot_name || '?start';
```

### **Формування таблиць для зберігання інформації**

Для функціонування бота Telegram необхідно зберігати дані по: партіям, тарам, баркодам (основна таблиця яка зберігатиме записи), працівникам. Окрім цього мають бути таблиці, які міститимуть інформацію по запиту користувачів – аудит запитів.

На основі вхідних даних сформовано структуру таблиць реляційної бази даних, що формують систему роботи віртуального помічника:

1. ASSISTANT\_PART – таблиця в якій зберігаються дані про партії які використовуються для маркування у складському обліку.

Поля таблиці:

- А) PRT\_ID (Table primary key) – унікальний код ID партії;
- Б) PRT\_NUM - № партії;
- В) PRT\_CODE - код партії;
- Г) PRT\_COMMENT – примітка до партії;
- Д) PRT\_DATE – дата створення партії;

2. ASSISTANT\_CONTAINER – таблиця в якій зберігаються дані про тари які використовуються для маркування у складському обліку.

Поля таблиці:

- А) CONT\_ID (Table primary key) – унікальний код ID тари;
- Б) CONT\_NAME – назва тари;
- В) CONT\_CODE - код тари;
- Г) CONT\_UNT – одиниці збереження по тарі;
- Д) CONT\_COMMENT – примітка до тари;
- 3) CONT\_DATE – дата створення тари;

3. ASSISTANT\_BARCODE – таблиця в якій зберігаються дані про записи баркодів.

Поля таблиці:

А) BC\_ID (Table primary key) – унікальний код ID баркода;

Б) BC\_PART – унікальний код ID баркод до партії, зв'язок із таблицею ASSISTANT\_PART;

В) BC\_CONTAINER - унікальний код ID баркод до тари, зв'язок із таблицею ASSISTANT\_CONTAINER;

Г) BC\_DATE – дата створення баркода;

Д) BC\_WORKER – id працівника який створив цей баркод, зв'язок із таблицею ASSISTANT\_WORKER;

З) BC\_COMMENT – примітка до баркода;

4. ASSISTANT\_WORKER – таблиця в якій зберігаються дані про працівників.

Поля таблиці:

А) WRKR\_ID (Table primary key) – унікальний код ID працівника;

Б) WRKR\_NAME – ім'я працівника;

В) WRKR\_LAST\_NAME - прізвище працівника;

Г) WRKR\_POSITION – посада працівника;

Д) WRKR\_COMMENT – примітка до запису працівника;

З) WRKR\_PHONE\_NUM – номер телефону;

К) WRKR\_TAB\_NUM – табельний номер працівника;

О) WRKR\_DATE – дата зачислення на роботу;

5. ASSISTANT\_USER – таблиця в якій зберігаються дані про запити/повідомлення.

Поля таблиці:

А) USR\_ID (Table primary key) – унікальний код ID повідомлень;

Б) USR\_NAME – назва чату;

В) USR\_DATE – дата створення чату;

Г) USR\_WRKR – унікальний код ID працівника, зв'язок із таблицею ASSISTANT\_WORKER;

Д) USR\_MESSAGE – текст повідомлення/запиту;

З) USR\_MESSAGE\_DATE – дата повідомлення/запиту;

Як результат, на рисунку 1 зображено сформовану структуру таблиць реляційної бази даних та взаємодії між ними

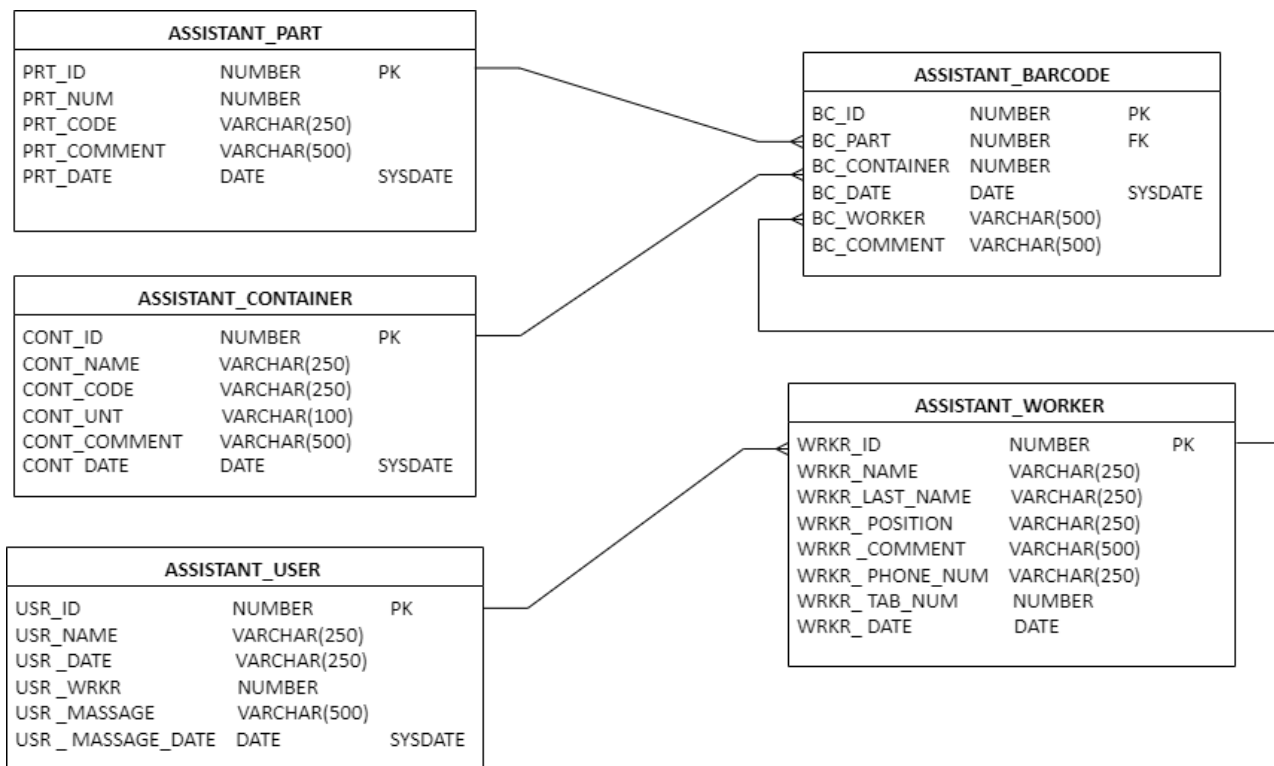


Рис. 1 – Структура таблиц реляційної бази даних,

### Функціональність бота

При першому запиті користувача бот перевіряє його унікальний Id код в таблиці “ASSISTANT\_WORKER”.

Якщо користувач в системі знайдений, бот зберігає його Id для аудиту. В іншому випадку бот повідомляє про відсутність прав доступу. Подібні віртуальні помічники використовуються тільки працівниками тих чи інших підприємств, однак необхідно ввести додаткові умови безпеки даних.

Функції запропонованого бота:

/get\_part <ID> — отримати інформацію про конкретну партію за її унікальним Id кодом;

/get\_container <ID> — отримати інформацію про тару за її унікальним Id кодом;

/get\_barcode <ID> — отримати інформацію по баркоду (наприклад, яка партія і тара прив'язана до цього баркоду).

/get\_worker <ID> — отримати інформацію про працівника за його унікальним Id кодом;

При кожному з цих запитів викликається відповідна PL/SQL процедура для отримання даних з бази. Наприклад, для команди /get\_part <ID> буде викликано процедуру GET\_PART\_ROW\_INFO.

Запити користувача зберігаються в таблиці “ASSISTANT\_USER” з інформацією про користувача, запитаний Id код та актуальний час запиту. Це дозволить вести історію активності і відслідковувати запити користувачів.

Приклад роботи бота:

Користувач відправляє запит: /get\_part 121238.

Бот перевіряє наявність користувача в таблиці працівників “ASSISTANT\_WORKER”.

Якщо користувач авторизований, бот викликає PL/SQL процедуру GET\_PART\_INFO, передає Id код (121238) і отримує інформацію про партію.

Відповідь з інформацією про партію відправляється користувачеві.

У таблицю “ASSISTANT\_USER” додається новий запис з даними про запит.

## Висновки

У статті розглянуто приклад впровадження віртуального помічника на базі Telegram Bot API в інтелектуальну систему маркування та обліку матеріальних ресурсів. Використання цього інструменту дозволяє спростити процес доступу до інформації, підвищити ефективність управління ресурсами та забезпечити зручність у користуванні системою. Результати дослідження підтверджують, що інтеграція віртуального помічника сприяє оптимізації облікових процесів, забезпечуючи швидкий доступ до даних.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ладуба М. Що таке нейролінгвістичне програмування та як воно працює / MC Today. [Електронний ресурс]. Режим доступу: <https://mc.today/uk/shho-take-nejrolingvistichne-programuvannya-ta-yak-vono-pratsyuje/>. Дата звернення: 07.10.2024.
2. В. Старжинський, О. Бісікало. «Застосування віртуальних помічників, їх переваги та недоліки, » в Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)», Вінниця, 2024. [Електронний ресурс]. Режим доступу: <https://d.conf.vntu.edu.ua/index.php/mn/mn2024/paper/view/20430>. Дата звернення: 11.03.2024. – 5 с.
3. Старжинський, В.; Бісікало, О. Розробка модуля “Штрих-коди” в інформаційній системі “ПлазмІС”. ЛІ НТКП ВНТУ. Факультет інтелектуальних інформаційних технологій та автоматизації, Вінниця. 2022. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/14602>. Дата звернення: 21.02.2022. – 3 с.
4. Telegram Integration in your APEX application. APEXPERT. [Електронний ресурс]. Режим доступу: <https://apexpert.in/2020/10/14/telegram-integration-in-your-apex-application/>. Дата звернення: 30.09.2024.

**Старжинський Валерій Юрійович** – аспірант групи 126-23а, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [3372292@gmail.com](mailto:3372292@gmail.com).

**Бісікало Олег Володимирович** – доктор технічних наук, професор, завідувач кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, e-mail: [obisikalo@vntu.edu.ua](mailto:obisikalo@vntu.edu.ua).

V. Starzhynskyi

O. Bisikalo

# APPLICATION OF A VIRTUAL ASSISTANT IN AN INTELLIGENT SYSTEM FOR MARKING AND ACCOUNTING OF MATERIAL RESOURCES

Vinnitsia National Technical University

## **Abstract**

*The article considers an example of implementing a virtual assistant into an intelligent system for marking and accounting of material resources. The main goal of the development is to simplify the information management process and increase the efficiency of accounting operations through automation. To integrate the virtual assistant, the Telegram Bot API was used, which provides quick and convenient access to information through a popular messaging platform. Particular attention was paid to the development of the relational database structure, which serves as the basis for the functioning of the system. The database tables are created based on input data and optimized for effective interaction with the bot, which allows users to quickly obtain the necessary information and update data about resources. The system has the potential for application in various areas where it is important to ensure accurate control over material resources.*

**Keywords:** virtual assistant, chatbot, Telegram, barcode, information.

**Valeriy Starzhinskii** – Department of intelligent information technologies and automation, Vinnitsia National Technical University, Vinnitsia, e-mail: [3372292@gmail.com](mailto:3372292@gmail.com).

**Bisikalo Oleh** – Doctor of Technical Sciences, Professor, Head of the Department of Automation and Intelligent Information Technologies, Vinnitsia National Technical University, Vinnitsia, e-mail: [obisikalo@vntu.edu.ua](mailto:obisikalo@vntu.edu.ua).