

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
Кафедра кібернетики та інформатики

"

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь – «Бакалавр»

на тему:

**«ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ TELEGRAM-БОТУ “ ПОМІЧНИК
МЕНЕДЖЕРУ З ПРОДАЖІВ”**

Виконав:	студент спеціальності
	126 «Інформаційні системи та технології»
	ОП «Інформаційні системи та технології»
	<i>Курило Ігор Миколайович</i>
Науковий керівник:	<i>Долгих Яна Володимирівна</i>
	доцент кафедри кібернетики та інформатики
Рецензент:	<i>Москаленко Альона Сергіївна</i>
	доцент кафедри комп'ютерних наук Сумського державного університету

Суми– 2025

СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет	Економіки і менеджменту
Кафедра	Кібернетики та інформатики
Освітній ступінь	«Бакалавр»
Спеціальність	<u>126 «Інформаційні системи та технології»</u> <u>ОП «Інформаційні системи та технології»</u>

Затверджую:

Завідувач кафедри

Світлана АГАДЖАНОВА

202

« » 5 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ *Курило Ігорю Миколайовичу*

1. Тема роботи: “Інформаційне та програмне забезпечення Telegram-боту "Помічник менеджера з продажів””.

2. База практичного дослідження: ТОВ «Епіцентр К»

керівник роботи Долгих Яна Володимирівна, к.е.н., доцент

затверджена наказом по університету від « » 2025 р. № -н

3. Строк подання студентом закінченого проекту (роботи) «26» червня 2025 року

4. Вихідні дані до проєкту (роботи):

результати аналітичних досліджень зарубіжних та вітчизняних вчених, інтернет-джерела

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

:

- виконати постановку задачі;

- вибрати середовище для реалізації;

- розробити технічне завдання;

- розробити програмний продукт відповідно до поставлених задач;

6. Дата видачі

« »

2025 р.

завдання:

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту	Примітка
1	Визначення об'єкту, предмету дослідження, формулювання мети та задач кваліфікаційної роботи, складання плану	<i>січень 2025 р.</i>	виконано
2	Підбір та вивчення літературних джерел, технічної документації	<i>січень 2025 р.</i>	виконано
3	Розробка технічного завдання	<i>січень 2025 р.</i>	виконано
4	Розробка телеграм боту	<i>січень-травень 2025 р.</i>	виконано
5	Оформлення пояснювальної записки	<i>до 30 травня 2025 р.</i>	виконано
6	Оформлення презентації	<i>до 1 червня 2025 р.</i>	виконано
7	Опрацювання зауважень керівника, перевірка на автентичність	<i>2 червня 2025 р.</i>	виконано
8	Підготовка доповіді та її попередній захист	<i>6 червня 2025 р.</i>	виконано
9	Перевірка з академічної доброчесності кваліфікаційної роботи	<i>10 червня 2025 р.</i>	виконано
10	Представлення кваліфікаційної роботи на кафедру, в деканат	<i>23 червня 2025 р.</i>	виконано
11	Захист кваліфікаційної роботи	<i>26 червня 2025р.</i>	
Студент			<i>Ігор Курило</i>

(підпис)

Керівник роботи

Яна ДОЛГІХ

(підпис)

Перевірку на автентичність
проведено

(підпис)

Кваліфікаційна робота допущена до

Світлана ЛУКАШ

захисту

(підпис)

АНОТАЦІЯ

Курило І.М. “Інформаційне та програмне забезпечення Telegram-боту "Помічник менеджера з продажів””.

Кваліфікаційна робота зі спеціальності 126 «Інформаційні системи та технології» ОП Інформаційні системи та технології, СНАУ, Суми-2025р. – Рукопис.

Кваліфікаційна робота присвячена створенню Телеграм-бота для автоматизації моніторингу продажів у ТОВ «Епіцентр К». Основною метою є розробка системи оперативного доступу до бізнес-аналітики через месенджерський інтерфейс, що забезпечить ефективне управління продажами та підвищить якість прийняття управлінських рішень.

Спроековано мікросервісну архітектуру системи, що включає Телеграм-бот на базі aiogram 3 та API-сервер на FastAPI. Система забезпечує асинхронну обробку запитів та зберігання даних користувачів у MongoDB.

Розроблено інтуїтивний користувацький інтерфейс з чотирма основними розділами: аналітика продажів, управління складом, фінансові метрики та клієнтська аналітика. Реалізовано систему автоматичної реєстрації

Створено тестовий API-сервер з вісьмома ендпоінтами, що імітує роботу з корпоративними джерелами даних та демонструє можливості майбутньої інтеграції з 1С: Підприємство.

Виконано комплексне тестування розробленої системи, що підтвердило її ефективність та готовність до промислового впровадження.

Розроблений Телеграм-бот рекомендується до впровадження в ТОВ «Епіцентр К» для підвищення ефективності моніторингу продажів та може служити основою для створення аналогічних систем в інших підприємствах роздрібної торгівлі.

Ключові слова: Телеграм-бот, Python, aiogram, FastAPI, MongoDB, моніторинг продажів, бізнес-аналітика, мікросервісна архітектура.

SUMMARY

Kurylo I.M. Information and software of the Telegram bot “Assistant to the Sales Manager”.

Qualification work in the specialty 126 "Information Systems and Technologies" EP Information Systems and Technologies, SNAU, Sumy-2025.
–Manuscript.

This qualification work is dedicated to creating a Telegram bot for automating sales monitoring at "Epicentr K" LLC. The main goal is to develop a system for operational access to business analytics through a messenger interface, ensuring effective sales management and improving the quality of managerial decision-making.

The study explored current approaches to creating corporate chatbots and identified the core requirements for systems that track sales. Methods for integrating Telegram bots with corporate systems through API were studied.

A microservice architecture was designed, including a Telegram bot based on aiogram 3 and an API server on FastAPI. The system provides asynchronous request processing and user data storage in MongoDB.

An intuitive user interface was developed with four main sections: sales analytics, warehouse management, financial metrics, and customer analytics. A system for automatic user registration and personal settings was implemented.

A test API server with eight endpoints was created, simulating work with corporate data sources and demonstrating possibilities for future integration with 1C: Enterprise.

Extensive testing of the system was carried out, demonstrating its efficiency and confirming its suitability for deployment in real-world industrial environments.

The developed Telegram bot is recommended for implementation at "Epicentr K" LLC to improve sales monitoring efficiency and can serve as a foundation for creating similar systems in other retail enterprises.

Keywords: Телеграм bot, Python, aiogram, FastAPI, MongoDB, sales monitoring, business analytics, microservice architecture.

ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Аналіз досліджень щодо методів, технологій розробки чат-ботів ...	11
1.2 Огляд наявних рішень застосування чат-ботів у бізнесі	14
1.2 Постановка задачі та функціональні вимоги до Телеграм – боту....	16
РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ТЕЛЕГРАМ БОТА.....	18
2.1 Мова програмування.....	18
2.2 Середовище розробки	19
2.3 Реєстрація бота в Телеграм.....	21
2.4 Стандарти проєктування IDEF0 та UML.....	22
РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ TELEGRAM-БОТУ “ПОМІЧНИК МЕНЕДЖЕРУ З ПРОДАЖІВ”	26
3.1 Розробка функціональності Телеграм-боту	26
3.2 Огляд функціональної частини та тестування Телеграм-боту	35
ВИСНОВКИ.....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТКИ.....	48
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ.....	49
ДОДАТОК Б ПЛАНУВАННЯ РОБОТИ.....	54
ДОДАТОК В ДІАГРАМА ДЕРЕВА НАВІГАЦІЇ ДОДАТКОМ.....	56
ДОДАТОК Г ПРИКЛАДИ ПРОГРАМНИХ МОДУЛІВ.....	57
ДОДАТОК Д ТЕЗИ КОНФЕРЕНЦІЙ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ	66

СПИСОК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface (інтерфейс програмування додатків)

HTTP – HyperText Transfer Protocol

JSON – Java Script Object Notation

REST – REpresentational State Transfer

CRUD – Create, Read, Update, Delete

БД – База даних

ТОВ – Товариство з обмеженою відповідальністю

ВСТУП

Актуальність теми. У сучасному світі роль інформаційних технологій у сфері роздрібної торгівлі стрімко зростає. Автоматизація процесів моніторингу продажів та управління підприємствами визначає їх ефективність та конкурентоспроможність. Сучасні месенджери все частіше використовуються як основа для впровадження корпоративних аналітичних і моніторингових систем. Відтак створення Телеграм-бота для управління стає важливим напрямом досліджень і практичної реалізації.

Кваліфікаційна робота присвячена розробці “Інформаційного та програмного забезпечення Telegram-боту "Помічник менеджера з продажів” для ТОВ «Епіцентр К». Обране підприємство характеризується широким спектром діяльності у галузі роздрібної торгівлі, що вимагає комплексного підходу до автоматизації процесів моніторингу продажів через месенджерські платформи.

Мета і завдання дослідження. Метою даної кваліфікаційної роботи є розробка та впровадження «Інформаційного та програмного забезпечення Telegram-боту "Помічник менеджера з продажів"» з мікросервісною архітектурою, який забезпечить автоматизацію збору, обробки та аналізу даних про продажі, складські операції, фінансові показники та клієнтську базу через інтеграцію з корпоративними системами ТОВ «Епіцентр К».

Для досягнення зазначеної мети автором були поставлені такі завдання:

- аналіз сучасних підходів до розробки корпоративних чат-ботів та визначення ключових вимог до систем моніторингу продажів у роздрібній торгівлі;
- дослідження методів інтеграції Телеграм-ботів з корпоративними системами для отримання аналітичних даних у реальному часі;
- проектування мікросервісної архітектури системи моніторингу з модульною структурою та асинхронною обробкою запитів;

- реалізація системи з використанням сучасного технологічного стеку (Python, aiogram 3, FastAPI, MongoDB) та принципів чистої архітектури;
- розробка інтуїтивного користувацького інтерфейсу з inline-клавіатурами та callback-системою для швидкої навігації;
- створення тестового API-сервера для імітації корпоративних джерел даних та демонстрації інтеграційних можливостей;
- комплексне тестування системи на продуктивність, стабільність та відповідність корпоративним вимогам.

Об'єктом дослідження процес розробки Інформаційного та програмного забезпечення Telegram-боту "Помічник менеджера з продажів".

Предметом дослідження інформаційні технології розробки Телеграм-боту помічник менеджера продажів

Практичне значення розробленого Телеграм-бота для ТОВ «Епіцентр К» полягає в автоматизації процесів моніторингу продажів, забезпеченні оперативного доступу до бізнес-аналітики через мобільний інтерфейс, що дозволить скоротити час на ефективне ухвалення управлінських рішень і підвищення ефективності роботи працівників підприємства.

Під час написання кваліфікаційної роботи застосовувалися наступні методи дослідження:

- системний аналіз з метою вивчення предметної області та визначення архітектурних патернів корпоративних систем;
- об'єктно-орієнтоване проєктування для створення модульної структури програмного комплексу;
- стандарти UML та IDEF0 для моделювання бізнес-процесів та проєктування взаємодії компонентів;
- методи RESTful API для організації взаємодії між компонентами системи;

асинхронне програмування для забезпечення високої продуктивності системи.

Теоретичну основу дослідження складають фундаментальні праці вітчизняних і зарубіжних учених у галузі інформаційних технологій, програмної інженерії та автоматизації бізнес-процесів. Зокрема, використано роботи з теорії проектування інформаційних систем, методології розробки програмного забезпечення, принципів мікросервісної архітектури та застосування месенджерських платформ у корпоративному середовищі.

Методологічною основою роботи є системний підхід до аналізу та проектування інформаційних систем, який дозволяє комплексно розглядати всі аспекти розробки Telegram-бота від постановки завдання до практичного впровадження. Застосування принципів об'єктно-орієнтованого проектування забезпечило створення модульної та легко масштабованої архітектури системи.

Наукова новизна роботи полягає у реалізації розробки комплексної методології створення корпоративних Телеграм-ботів для моніторингу продажів, яка поєднує мікросервісну архітектуру з асинхронною обробкою запитів, автоматизованою системою управління користувачами та гнучкою інтеграцією з корпоративними системами.

Апробація результатів кваліфікаційної роботи. Положення кваліфікаційної роботи доповідалися на таких конференціях:

- 1) всеукраїнській науковій конференції студентів та аспірантів, присвяченої міжнародному дню студента, 18-22 листопада 2024 р. Суми;
- 2) науково-практичній конференції викладачів, аспірантів та студентів Сумського НАУ, 14-18 квітня 2025 р. Суми.

Публікації. За матеріалами кваліфікаційної роботи опубліковано 2 тези з наукових конференцій.

Структура і обсяг роботи. Кваліфікаційна робота містить 73 сторінок основного тексту, має 28 ілюстрацій, включає 3 таблиці, список використаних джерел складається із 11 позицій, а також 5 додатків.

У першому розділі представлено вивчення предметної області та актуальні рішення у сфері корпоративних чат-ботів для моніторингу бізнес-процесів, визначено основні вимоги до системи та обґрунтовано актуальність розробки.

Другий розділ присвячено вибору програмних засобів та проєктуванню системи, включаючи обґрунтування технологічного стеку та створення архітектурних діаграм з використанням стандартів IDEF0 та UML.

У третьому розділі розглядається розробка функціональних компонентів Телеграм-бота, включаючи створення API-сервера, реалізацію користувацького інтерфейсу та результати комплексного тестування системи.

У висновках узагальнено ключові результати проведеного дослідження та окреслено практичну цінність створеного Телеграм-бота як інструмента підвищення ефективності управління в ТОВ «Епіцентр К».

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз досліджень щодо методів, технологій розробки чат-ботів

Сфера розробки чат-ботів активно розвивається, а сучасні наукові роботи охоплюють різноманітні підходи, алгоритми та технологічні платформи, спрямовані на створення адаптивних і функціональних систем для автоматизації взаємодії з користувачами [5, 6].

Одним з основних напрямів є розробка чат-ботів на основі сценаріїв (rule – based systems). Такі системи опираються на чітко структуровані алгоритми та попередньо визначені схеми діалогів. Перевагою цього підходу є стабільність і прогнозованість поведінки бота, однак його можливості суттєво обмежені в ситуаціях, коли користувач виходить за межі передбачених сценаріїв. У таких випадках відповіді можуть бути нерелевантними або помилковими. Науковці в цій сфері працюють над удосконаленням алгоритмів і розширенням діалогових шаблонів, щоб охопити ширший спектр взаємодій [3].

Важливим напрямом сучасних досліджень є створення чат-ботів на основі технологій штучного інтелекту (AI-driven technologies), що використовують методи машинного навчання та засоби обробки природної мови (Natural Language Processing, NLP).

Сучасні алгоритми обробки природної мови включають широкий спектр технологій, починаючи від базового токенізації тексту і закінчуючи складними нейронними мережами. Особливе місце займають моделі типу BERT (Bidirectional Encoder Representations from Transformers), які здатні розуміти контекст слова залежно від оточуючих його термінів. Ці моделі демонструють значно кращі результати порівняно з традиційними

підходами, особливо при роботі з багатозначними словами та складними граматичними конструкціями. Впровадження таких технологій у корпоративні чат-боти дозволяє досягти рівня розуміння, близького до людського, що особливо важливо при обробці специфічної термінології галузі роздрібної торгівлі.

Важливим напрямом розвитку є також використання техніки transfer learning, яка дозволяє адаптувати попередньо натреновані моделі під специфічні корпоративні задачі. Це значно скорочує час та ресурси, необхідні для створення ефективного чат-бота, оскільки не потрібно навчати модель з нуля. Натомість використовуються знання, отримані на великих загальних датасетах, які потім доналаштовуються під конкретну предметну область. Такий підхід особливо ефективний для малих та середніх підприємств, які не мають можливості інвестувати значні ресурси в розробку власних моделей машинного навчання.

Реалізація архітектур глибокого навчання, зокрема трансформерних моделей та LSTM-мереж, дозволяє системам здійснювати контекстний аналіз користувацьких запитів та самовдосконалюватися через накопичення інтерактивного досвіду. Подібний підхід сприяє формуванню більш природного та індивідуалізованого діалогу, проте вимагає значних ресурсів обчислення та детальної оптимізації алгоритмів [3, 6].

Дослідження також акцентують на інтеграції чат-ботів із зовнішніми платформами та корпоративними екосистемами. Завдяки відкритим інтерфейсам програмування (API), наприклад Телеграм Bot API, розробники можуть створювати рішення, які гармонійно вбудовуються в наявну інфраструктуру компаній [7]. Питання безпеки даних, масштабування систем і швидкості обробки запитів також є важливими темами в наукових статтях. Дослідження впливу дизайну мови та стилю комунікації чат-ботів показують значний вплив на сприйняття користувачами та їх готовність до взаємодії [1, 4].

Окрім технічних і функціональних характеристик, важливу роль у впровадженні чат-ботів відіграє питання користувацького досвіду (User Experience, UX). Досвід показує, що навіть найсучасніші технології не забезпечують очікуваного ефекту без належної уваги до зручності, інтуїтивності та привабливості інтерфейсу. Успішні корпоративні рішення базуються на глибокому аналізі поведінки користувачів, їхніх потреб та очікувань. Важливо враховувати не лише технічні аспекти, а й психологічні бар'єри, які можуть виникати у співробітників під час переходу на нові цифрові інструменти. Саме тому процес розробки чат-бота повинен включати етапи тестування з реальними користувачами, збір зворотного зв'язку та поступове вдосконалення системи. Такий підхід дозволяє мінімізувати ризики відторгнення нововведень і сприяє швидшій адаптації персоналу до автоматизованих рішень.

Важливим аспектом у процесі розробки корпоративних чат-ботів є врахування культурних та поведінкових особливостей цільової аудиторії. Це дозволяє адаптувати стиль спілкування та функціонал бота відповідно до очікувань користувачів, що підвищує рівень їхньої довіри до цифрового сервісу та сприяє більш ефективній взаємодії з системою.

Окрім технічних аспектів, аналізується економічна доцільність використання чат-ботів, їхній вплив на скорочення витрат і покращення клієнтського досвіду [2].

Сучасні методології розробки чат-ботів включають кілька основних підходів. Agile-методологія дозволяє швидко адаптуватися до змінних вимог користувачів. DevOps-практики забезпечують безперервну інтеграцію та розгортання нових функцій.

Хмарні технології відкривають нові можливості для масштабування ботів. Amazon Web Services, Google Cloud Platform та Microsoft Azure надають готові рішення для розробки та хостингу чат-ботів. Це значно спрощує процес розгортання та підтримки системи.

Штучний інтелект стає все більш доступним для розробників. Готові

API від OpenAI, Google та Microsoft дозволяють легко додавати інтелектуальні функції до ботів без глибоких знань в області машинного навчання.

Дослідження показують, що поєднання сценарного підходу з технологіями штучного інтелекту забезпечує баланс між надійністю та гнучкістю, що є критично важливим для задоволення потреб бізнесу й користувачів.

Особливої уваги заслуговує підхід гібридних систем, які поєднують сценарну логіку з елементами машинного навчання. Такі рішення забезпечують стабільність роботи для стандартних запитів та гнучкість при обробці нетипових ситуацій. У корпоративному середовищі це особливо важливо, оскільки співробітники потребують передбачуваних відповідей на робочі запити.

Телеграм-боти мають кілька важливих переваг порівняно з веб-додатками. По-перше, їх не потрібно встановлювати - достатньо мати Telegram на телефоні. По-друге, боти працюють швидко та не займають місце на пристрої.

Багато українських компаній вже переконалися в ефективності ботів. Вони допомагають швидко отримувати інформацію, автоматизують рутинні завдання та покращують комунікацію між відділами.

Важливо відзначити, що ефективність впровадження чат-ботів у корпоративному середовищі значною мірою залежить від рівня інтеграції з існуючими бізнес-процесами та корпоративними інформаційними системами. Для досягнення максимального ефекту необхідно забезпечити безперебійну взаємодію між ботом і такими системами, як CRM, ERP, бухгалтерські програми тощо. Це дозволяє не лише автоматизувати рутинні операції, а й отримувати аналітичну інформацію в реальному часі, що значно підвищує якість прийняття управлінських рішень. Досвід провідних компаній свідчить, що правильно інтегрований чат-бот здатен стати ключовим елементом цифрової трансформації підприємства, підвищити рівень прозорості бізнес-процесів та скоротити витрати на обслуговування клієнтів.

Досвід впровадження Telegram-ботів у різних сферах бізнесу в Україні демонструє, що такі рішення не лише оптимізують внутрішні бізнес-процеси, а й дозволяють швидко реагувати на зміни ринкової ситуації. Наприклад, у сфері роздрібної торгівлі боти використовуються для оперативного інформування співробітників про залишки товару, акційні пропозиції та зміни у графіку постачань, що позитивно впливає на загальну ефективність роботи підприємства.

Отже, сучасні наукові пошуки в галузі розробки чат-ботів підкреслюють необхідність комплексного підходу, який інтегрує стабільність сценарних систем із можливостями самонавчання й адаптації. Така стратегія закладає підґрунтя для створення ефективних, безпечних і конкурентоспроможних цифрових рішень.

1.2 Огляд наявних рішень застосування чат-ботів у бізнесі

Враховуючи зростаючу популярність месенджера Телеграм серед українців, вітчизняні підприємства активно інтегрують чат-боти на цій платформі. Зокрема, їх застосовують для полегшення продажів, поліпшення обслуговування клієнтів та оперативного поширення інформації. У наведених прикладах розкриваються найпоширеніші види рішень, що використовуються в Україні, а також ключові функції ботів та типові виклики, які виникають у процесі їх функціонування.

Наведемо приклади застосування Телеграм - ботів в компаніях України. До банківських чат-ботів відноситься чат-бот Монобанку, який надає швидкий доступ до служби підтримки [9]. Логістичним чат-ботом є офіційний бот «Нова Пошта» зосереджений на консультаціях через оператора. Функціонал бота спрямований на допомогу клієнтам: користувачі отримують відповіді через живий чат [10].

Сучасні корпоративні чат-боти, зокрема ті, що використовуються у великих компаніях, дедалі частіше інтегруються з CRM-системами, внутрішніми базами

даних та іншими корпоративними платформами. Така інтеграція дозволяє не лише автоматизувати стандартні запити, а й забезпечити персоналізований підхід до кожного клієнта, враховуючи історію його звернень, попередні покупки та особливості взаємодії з компанією. Крім того, аналітичні модулі, вбудовані в чат-боти, дають змогу керівництву оперативно отримувати зведені звіти щодо основних бізнес-показників, виявляти проблемні ділянки у роботі з клієнтами та своєчасно реагувати на зміни ринкової ситуації. Це відкриває нові можливості для підвищення лояльності клієнтів і оптимізації внутрішніх процесів компанії.

Офіційний бот «Укрпошта» вирізняється розширеним функціоналом порівняно з аналогом від Нової Пошти – він дозволяє автоматично відстежувати посилки, отримувати дані щодо статусу доставлення, тарифів і розташування відділень, що значно полегшує контроль за відправленнями та підвищує зручність для користувачів [11]. Бот державного порталу «Дія» надає можливість швидко знаходити інформацію про державні послуги, оформлювати документи та вирішувати адміністративні питання, інтегруючи дані з офіційних джерел і забезпечуючи високий рівень безпеки та швидкість взаємодії [8].

Аналізуючи досвід впровадження Telegram-ботів у вітчизняних компаніях, можна виділити кілька ключових тенденцій. По-перше, спостерігається зміщення акценту з простого інформування до комплексної автоматизації бізнес-процесів. Сучасні боти не обмежуються відповідями на FAQ, а стають повноцінними робочими інструментами, здатними виконувати складні операції з базами даних, генерувати звіти та інтегруватися з різноманітними корпоративними системами.

По-друге, зростає важливість персоналізації взаємодії. Успішні корпоративні боти адаптують свою поведінку під конкретного користувача, враховуючи його посаду, відділ, попередні запити та робочі звички. Це досягається через інтеграцію з системами управління персоналом та активний збір метрик користувацької активності. Персоналізація не лише покращує користувацький досвід, а й підвищує ефективність роботи, оскільки співробітники отримують саме ту інформацію, яка їм необхідна для виконання своїх обов'язків.

По-третє, спостерігається тенденція до створення багатофункціональних

платформ замість вузькоспеціалізованих рішень. Сучасні корпоративні боти часто об'єднують функції HR-підтримки, технічної допомоги, бізнес-аналітики та управління проектами в єдиному інтерфейсі. Це спрощує життя співробітників, які можуть отримати всю необхідну інформацію в одному місці, не переключаючись між різними додатками та системами.

Зарубіжний досвід показує широке впровадження чат-ботів у різних галузях. За даними аналітичної компанії Gartner, українські підприємства демонструють 45% річне зростання впровадження корпоративних Telegram-ботів. Особливо активно технологію адаптують компанії сфери роздрібної торгівлі, де боти допомагають скоротити час обробки запитів до 70%. Дослідження McKinsey підтверджують, що автоматизація через чат-боти підвищує продуктивність співробітників на 25-30%.

У США банки активно використовують ботів для консультацій клієнтів. Bank of America створив віртуального помічника Erica, який обслуговує мільйони запитів щомісяця.

Порівняльний аналіз підходів до впровадження чат-ботів у різних країнах свідчить, що особливу увагу приділяють питанням захисту персональних даних та відповідності локальним і міжнародним стандартам. Зокрема, європейські компанії активно впроваджують механізми шифрування даних та багаторівневу автентифікацію користувачів, що дозволяє підвищити рівень безпеки інформаційних систем.

Азійський ринок демонструє найшвидший розвиток чат-ботів. Китайські компанії інтегрують ботів у месенджери WeChat та Alipay, створюючи цілі екосистеми сервісів. Це показує потенціал для розвитку подібних рішень в Україні.

Досвід українських компаній, які вже впровадили Телеграм-боти, підкреслює значний потенціал цієї технології. Найбільш помітним результатом є оптимізація бізнес-процесів: чат-боти можуть відповідати на типові запитання, одночасно надаючи клієнтам швидкий доступ до необхідних послуг. Проте під час розробки таких проектів необхідно враховувати не тільки обмеження платформи та питання безпеки, але і

якість інтеграції з внутрішньою інфраструктурою та рівень підтримки користувачів. Саме комплексний підхід до розробки та обслуговування бота забезпечить максимальну економічну вигоду та посилить конкурентні переваги компанії на ринку.

Найбільший успіх мають боти, які вирішують реальні проблеми користувачів. Наприклад, швидкий доступ до інформації про продажі, стан складу або фінансові показники. Такі функції дійсно корисні для щоденної роботи менеджерів.

Важливим фактором успіху є простота використання. Бот повинен мати зрозуміле меню, швидко відповідати на запити та надавати інформацію у зручному форматі. Користувачі швидко відмовляються від складних у використанні систем.

Безпека даних також є критично важливою. Корпоративні боти працюють з конфіденційною інформацією, тому потрібно забезпечити належний рівень захисту. Це включає шифрування даних, контроль доступу та регулярне оновлення системи безпеки.

1.2 Постановка задачі та технічні вимоги до Телеграм-боту

Розробка Телеграм-бота для моніторингу продажів у ТОВ «Епіцентр К» спрямована на автоматизацію збору й обробки даних про продажі, що підвищить ефективність управління бізнес-процесами.

Мета проєкту – розробити Телеграм-бот, який інтегрується з системами підприємства для забезпечення оперативного доступу до даних про продажі, залишки товарів і статус замовлень. Бот має слугувати зручним інструментом для співробітників, скорочуючи час на отримання інформації та зменшуючи навантаження на персонал.

Завдання включають:

- визначення ключових показників для моніторингу (обсяги продажів, категорії товарів, регіони);

- розробку архітектури бота з інтеграцією до внутрішніх систем через API;
- створення тестового API-сервіса для демонстрації функціональності;
- збереження користувачів у базі MongoDB;
- забезпечення стабільності роботи при високому навантаженні.

Функціональні вимоги телеграм бота є:

1. *Аутентифікація*: ідентифікація користувачів за Телеграм ID із розподілом прав доступу (менеджери, логісти);
2. *Отримання даних*: доступ до актуальних даних через API в реальному часу;
3. *Інтерфейс*: зручна навігація;
4. *Обробка помилок*: повідомлення про збої з альтернативними діями.

Розробка бота дозволить оптимізувати моніторинг продажів у ТОВ «Епіцентр К», підвищивши швидкість доступу до даних і ефективність їх аналізу. Використання Телеграм як платформи обґрунтовано його безпекою, швидкодією та гнучким API.

В бакалаврській роботі реалізовано функціональний прототип системи, що складається з Телеграм-бота та тестового API-сервіса. Це рішення обумовлено такими факторами:

- складність підключення до корпоративних систем 1С на етапі розробки прототипу;
- необхідність забезпечення безпеки корпоративних даних;
- можливість демонстрації повного функціоналу системи.

MongoDB використовується для зберігання інформації про користувачів Бізнес-дані (продажі, склад, фінанси) отримуються через API-виклики до тестового сервісу, що імітує роботу з реальними корпоративними системами.

Архітектура передбачає можливість майбутньої інтеграції з реальними джерелами даних через заміну тестового API-сервіса на справжні API-ендпоінти 1С: Підприємство.

РОЗДІЛ 2

ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ТЕЛЕГРАМ БОТА

2.1 Мова програмування

Мова програмування є фундаментом проєкту, визначаючи продуктивність бота, швидкість його розробки та можливість інтеграції з внутрішніми системами ТОВ «Епіцентр К», такими як 1С: Підприємство та MongoDB для збереження даних.

Для вибору мови програмування розглянуто кілька варіантів:

- Python: відомий своєю простотою, багатою екосистемою бібліотек і підтримкою асинхронного програмування, що робить його ідеальним для створення чат-ботів і роботи з базами даних.
- JavaScript: ефективний для асинхронних задач через Node.js, однак не є оптимальним для інтеграції з корпоративними платформами типу 1С.
- Java: надійний і потужний для великих проєктів, але надто складний для швидкої розробки бота в межах обмеженого часу.
- PHP: простий у роботі з API, але поступається Python у продуктивності й підтримці асинхронних операцій, необхідних для обробки множинних запитів.

Python обрано як основну мову програмування, а бібліотека aiogram 3 – як ключовий інструмент для роботи з Телеграм API з таких причин:

- *простота й швидкість*: зрозумілий синтаксис Python дозволяє швидко створювати прототипи й реалізовувати логіку бота, що відповідає часовим обмеженням переддипломної практики;
- *асинхронна обробка*: aiogram 3, заснована на фреймворку asyncio, дає змогу оптимально обробляти значну кількість одночасних запитів від співробітників ТОВ «Епіцентр К», гарантуючи оперативність відповідей;
- *інтеграція з MongoDB*: Python підтримує бібліотеку Beanie – асинхронну

ORM для MongoDB, яка спрощує збереження й управління структурованими даними про продажі;

- *зручність із Телеграм API*: aiogram 3 надає зручні засоби для реалізації команд, інтерактивних кнопок і обробки повідомлень, що відповідає потребам створення інтуїтивного інтерфейсу для користувачів;
- *спільнота й документація*: остання версія aiogram 3 має актуальну документацію та активну спільноту, що полегшує пошук рішень для технічних викликів.

Python із aiogram 3 використовуватиметься для:

- обробки команд користувачів (наприклад, /sales для звітів про продажі, /stock для залишків);
- інтеграції з MongoDB через Beanie для збереження даних у реальному часі;
- виконання запитів до 1С через API з використанням бібліотеки aiohttp для асинхронного обміну даними;
- забезпечення базового рівня безпеки завдяки шифруванню запитів через SSL/TLS, що підтримується Телеграм.

Python із aiogram 3 формує потужну основу для реалізації швидкого, масштабованого й інтегрованого Телеграм-бота, який відповідає потребам моніторингу продажів у ТОВ «Епіцентр К».

Вибір Python як основної мови програмування також обумовлений його унікальною екосистемою бібліотек для роботи з даними. Бібліотеки pandas та numpy забезпечують ефективну обробку великих обсягів інформації, що критично важливо при аналізі продажів та генерації звітів. Matplotlib та seaborn дозволяють створювати якісні візуалізації даних, які можуть бути інтегровані в повідомлення бота у вигляді графіків та діаграм.

Важливим фактором вибору Python є також його широка підтримка контейнеризації через Docker. Це дозволяє легко розгортати Telegram-бота в різних середовищах - від локальних серверів до хмарних платформ. Контейнеризація забезпечує консистентність роботи системи незалежно від

операційного середовища та спрощує процеси оновлення та масштабування.

Система управління пакетами `pip` та віртуальні середовища `venv` забезпечують ізоляцію залежностей проекту, що критично важливо для корпоративних розробок. Це дозволяє уникнути конфліктів версій бібліотек та забезпечує стабільність роботи системи навіть при оновленні окремих компонентів.

Важливим аспектом при виборі технологічного стеку стала можливість швидкого прототипування та ітеративної розробки. Python з `aiogram 3` дозволяє реалізувати базовий функціонал бота за 2-3 дні, що критично важливо для демонстрації `proof-of-concept` замовнику та збору раннього зворотного зв'язку від користувачів.

Екосистема Python також забезпечує легку інтеграцію з системами бізнес-аналітики через бібліотеки `pandas` та `pumpru`, що відкриває можливості для майбутнього розширення функціоналу системи аналітичними модулями та машинним навчанням.

Важливою особливістю `aiogram 3` є кардинально переписана архітектура `middleware` та роутерів, що забезпечує кращу продуктивність порівняно з попередніми версіями. Фреймворк повністю підтримує Python 3.9+ типізацію, включаючи `Generic` типи та `Union`, що значно покращує розробку та налагодження. Інтеграція з `Pydantic v2` забезпечує швидку валідацію даних та автоматичну серіалізацію JSON-об'єктів.

Особливою перевагою `aiogram 3` є підтримка `middleware` на рівні архітектури, що дозволяє ефективно обробляти аутентифікацію користувачів та логування запитів. Фреймворк також надає вбудовані засоби для роботи з `webhook` та `polling`, що забезпечує гнучкість при виборі способу отримання оновлень від Telegram API. Крім того, Python забезпечує ефективну роботу з JSON-форматом, що критично важливо для обміну даними між Telegram-ботом та API-сервером.

2.2 Середовище розробки

Ефективне середовище розробки сприяє зручному написанню, налагодженню та тестуванню коду, що є критично важливим для створення якісного програмного продукту. Для цього проєкту обрано PyCharm-інтегроване середовище розробки (IDE), яке оптимально підходить для роботи з Python і асинхронними бібліотеками.

PyCharm обрано через низку переваг, які відповідають вимогам проєкту:

- *підтримка Python*: PyCharm пропонує автодоповнення коду, підсвітку синтаксису й інтеграцію з бібліотеками, такими як aiogram 3 і Beanie, що значно прискорює процес розробки;
- *інструменти налагодження*: вбудований дебагер дозволяє оперативно виявляти й усувати помилки в асинхронному коді, що важливо для тестування реакції бота на запити;
- *управління залежностями*: підтримка віртуальних середовищ дає змогу ізолювати бібліотеки (aiogram 3, Beanie, aiohttp), забезпечуючи стабільність проєкту;
- *робота з MongoDB*: плагіни для MongoDB полегшують налаштування й тестування запитів до бази даних через Beanie;
- *контроль версій*: Інтеграція з Git спрощує відстеження змін у коді й співпрацю над проєктом.

Використання PyCharm забезпечує:

- ~ структурування коду для легкого масштабування функціоналу бота в майбутньому (наприклад, додавання аналітики чи прогнозування);
- ~ швидке налаштування підключення до Телеграм API за допомогою aiogram 3 і до MongoDB через Beanie;
- ~ зручний інтерфейс для роботи з асинхронними функціями, що підвищує продуктивність розробки.

PyCharm оптимізує робочий процес, гарантуючи високу якість коду й

ефективність створення Телеграм-бота для ТОВ «Епіцентр К».

Інтегроване середовище розробки значно спрощує роботу програміста. Всі необхідні інструменти знаходяться в одному місці, що економить час та зменшує кількість помилок. PyCharm автоматично перевіряє синтаксис коду та пропонує варіанти його покращення.

Використання професійних інструментів розробки є обов'язковим для створення якісного програмного забезпечення. Це особливо важливо для корпоративних проектів, де надійність та стабільність мають критичне значення.

2.3 Реєстрація бота в Телеграм

Реєстрація бота в Телеграм є початковим етапом реалізації проєкту, який забезпечує доступ до Телеграм API та унікального токена для авторизації й взаємодії з платформою.

Процес реєстрації телеграм-бота виконується через офіційного бота Телеграм – @BotFather:

1. Розробка бота: У чаті з @BotFather вводиться команда /newbot. Користувач задає назву бота (наприклад, "SalesTracker456Bot") та унікальний ідентифікатор, який закінчується на "Bot" (наприклад, "@SalesTracker456Bot").
2. Отримання токена: Після створення @BotFather видає API – токен (наприклад, 123456:ABC – DEF1234ghIkl – zyx57W2v1u123ew11), який слугує ключем для підключення бота до Телеграм API.
3. Налаштування параметрів: Команда /setdescription дозволяє додати опис (наприклад, "Бот для моніторингу продажів у ТОВ Епіцентр К"), а /setcommands – визначити список команд (наприклад, /sales, /stock, /orders).
4. Збереження токена: Токен записується в код у захищеній змінній середовища (наприклад, через файл .env), щоб уникнути витоку даних.

Безпека токена є критично важливою для роботи бота. Якщо токен

потрапить до сторонніх осіб, вони зможуть повністю контролювати бот. Тому важливо зберігати токен в захищеному місці та ніколи не публікувати його у відкритому коді.

Telegram надає надійну систему авторизації для ботів. Кожен бот має унікальний токен, який дозволяє йому взаємодіяти з API платформи. Цей підхід забезпечує безпеку та контроль над ботами.

2.4 Стандарти проєктування IDEF0 та UML

Розробка Телеграм - бота для моніторингу продажів у ТОВ “Епіцентр К”, з використанням стандартів IDEF0 та UML, являє собою зручний та структурований підхід до створення інформаційних систем. Ці стандарти сприяють чіткому визначенню функціональності бота, його взаємодії з користувачами та послідовності дій. Розглянемо, як ці стандарти були використані для моделювання роботи бота, а також наведемо ключові діаграми, які лягли в основу його розробки.

IDEF0 (Integration Definition for Function Modeling) – це методологія, яка використовується для моделювання функцій системи, їх взаємозв’язків та інтерфейсів. За допомогою IDEF0 було створено контекстну діаграму, яка описує загальну модель роботи бота та його взаємодію з користувачами. Наприклад, у рамках IDEF0 можна виділити такі основні функції бота:

- отримання запиту користувача після введення команди /start та вибору пункту меню;
- обробка запиту для визначення необхідної інформації;
- виконання дії, наприклад, формування звіту на основі даних;
- надсилання відповіді користувачу у вигляді звіту про продажі, стан складу або фінансові показники.

Контекстна діаграма IDEF0 (A – 0) є важливим інструментом для розуміння роботи бота в інформаційному середовищі. Контекстна діаграма

візуалізує, як система обробляє запити користувачів, отримуючи дані через API-виклики до зовнішніх сервісів, та формує відповіді у вигляді аналітичних звітів. MongoDB застосовується для збереження даних про користувачів. Це дозволяє чітко окреслити межі системи та її завдання, що стало основою для подальшої розробки. На рисунку 2.1 зображена контекстна діаграма створення бота для моніторингу продажів у ТОВ "Епіцентр К".

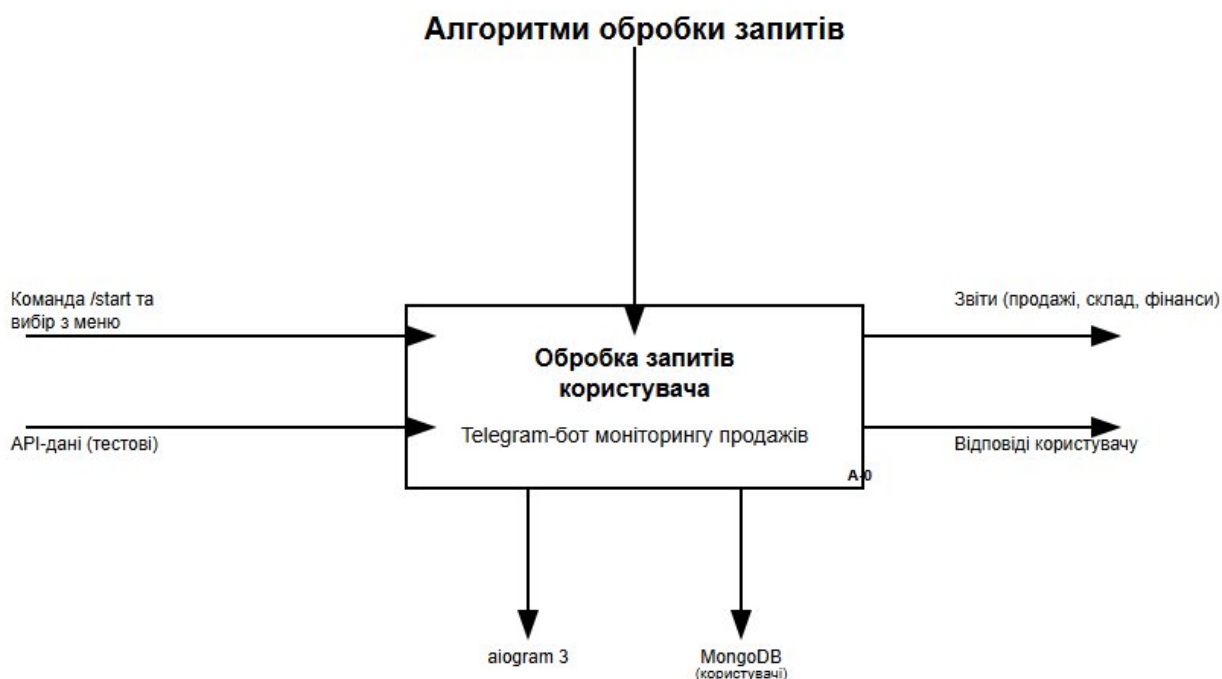


Рисунок 2.1 – Контекстна діаграма IDEF0 (A – 0)

UML (Unified Modeling Language) – вважається основною мовою моделювання, що застосовується для проектування програмного забезпечення. За допомогою UML було створено діаграми, що описують поведінку бота та його взаємодію з користувачами. У межах цього проекту використано такі діаграми:

- діаграма варіантів використання (Use Case Diagram): відображає, як користувач взаємодіє з ботом і які сценарії використання доступні;
- діаграма послідовності (Sequence Diagram): характеризує послідовність дій під час обробки запитів.

Діаграма варіантів використання вказує на акторів, які взаємодіють із

системою, та функції, які вони можуть виконувати. Акторами є користувач (співробітник ТОВ “Епіцентр К”, наприклад, менеджер або логіст) та зовнішня система даних, яка надає інформацію через API: Підприємство, яка надає дані. Сценарії використання включають перегляд звітів про продажі, перевірку стану складу та аналіз фінансових показників. Ця діаграма допомагає зрозуміти, як функціонал бота пов’язаний із потребами користувачів, і як організувати навігацію після введення початкової команди та вибору потрібного пункту меню. Наприклад, менеджери можуть переглядати фінансові звіти, а логісти – лише дані про склад. Побудована діаграма представлена на рисунку 2.2.

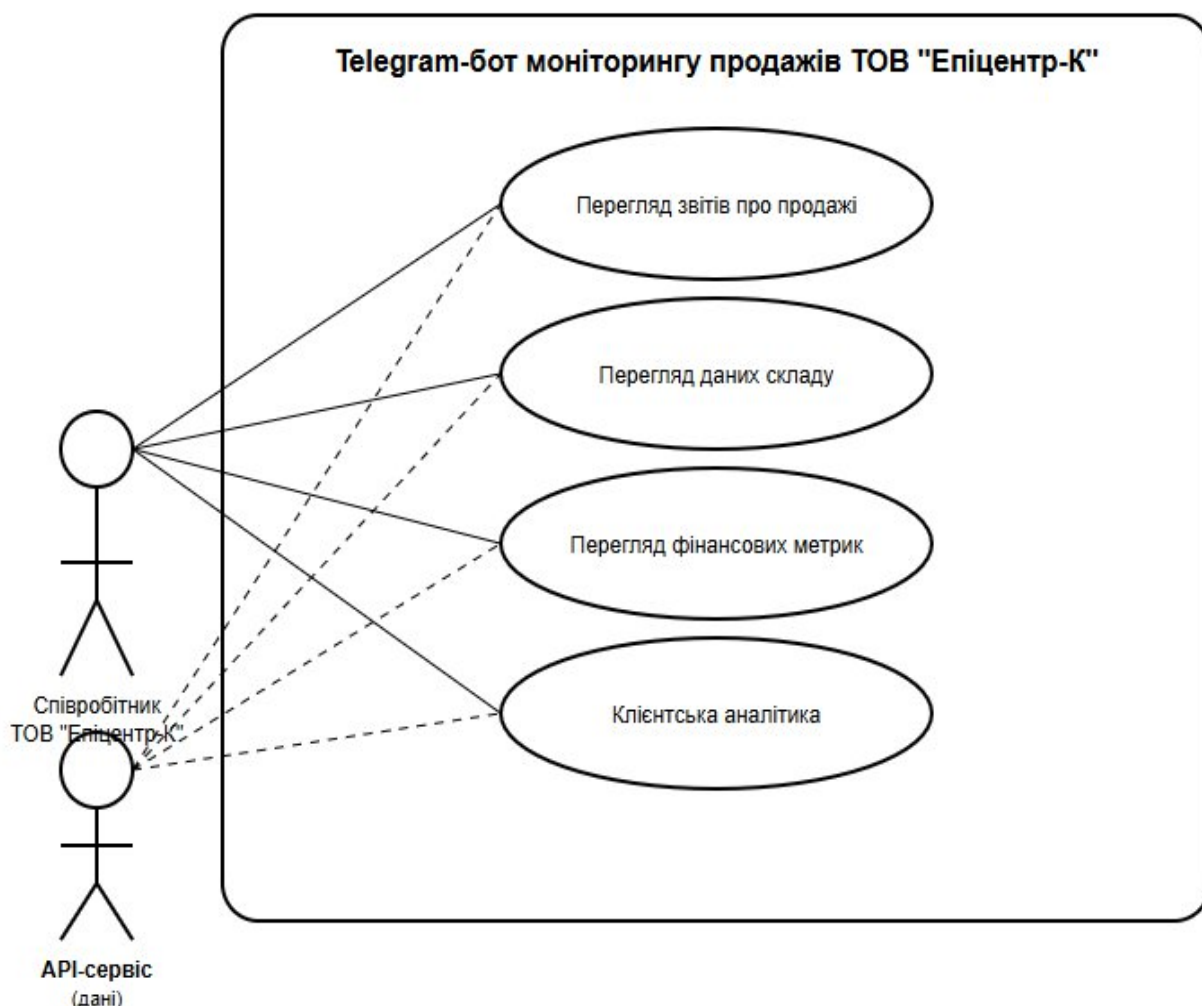


Рисунок 2.2 – Діаграма варіантів використання (Use Case Diagram)

Діаграма послідовності (Sequence Diagram)+ відображає поетапність дій

під час обробки запитів. Вона показує, як після введення команди та вибору пункту меню (наприклад, "Перегляд звітів про продажі") бот приймає запит, здійснює API-виклик до зовнішнього сервісу для отримання актуальних даних, обробляє отриману інформацію і формує звіт, після чого відправляє відповідь користувачу. Такі діаграми дозволяють детально проаналізувати процес обробки запитів і виявити можливі проблеми на ранніх етапах розробки.

Застосування стандартів IDEF0 та UML забезпечило чітке розуміння функціоналу бота та його структури. Контекстна діаграма IDEF0 (A – 0) визначила основну функцію бота – обробку запитів користувача через API-взаємодію з зовнішніми джерелами даних і показала, як він інтегрується в робочий процес компанії. Діаграма варіантів використання UML окреслила ключові сценарії використання, що стало основою для створення зручного інтерфейсу. Діаграма послідовності деталізувала процес обробки запитів через API-виклики, що полегшило розробку та тестування. Ці схеми разом створили структурований підхід до проєктування, що відповідає потребам ТОВ "Епіцентр К" у моніторингу продажів.

Окрім основних діаграм UML, для деталізації архітектури системи були використані додаткові модельні представлення. Діаграма компонентів допомогла визначити структуру програмних модулів та їх взаємозалежності, що стало основою для організації коду проєкту. Діаграма розгортання показала, як компоненти системи будуть розподілені між різними серверами та пристроями в продуктивному середовищі.

Використання стандартів проєктування значно полегшує розробку складних систем. IDEF0 та UML допомагають зрозуміти структуру майбутньої системи ще до початку програмування. Це економить час і зменшує кількість помилок.

Правильне проєктування особливо важливе для корпоративних систем. Такі проєкти зазвичай мають багато вимог і повинні інтегруватися з існуючими системами компанії. Детальне планування допомагає уникнути

проблем на пізніх етапах розробки.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ TELEGRAM-БОТУ "ПОМІЧНИК МЕНЕДЖЕРУ З ПРОДАЖІВ".

3.1 Розробка функціональності Телеграм-боту

Обґрунтувавши вибір технологічного стеку та програмних засобів, було розпочато етап безпосередньої реалізації системи. Розроблювана платформа моніторингу продажів має дворівневу архітектуру і включає такі основні модулі: API-сервер для централізованої обробки бізнес-даних та Телеграм-бот для забезпечення зручного користувацького інтерфейсу.

Першим етапом розробки є інсталяція та конфігурація середовища розробки. Для запуску проєкту необхідно встановити Python версії 3.11 або новішої. Процес встановлення Python включає завантаження інсталлятора з офіційного сайту python.org. При виборі версії важливо переконатися в сумісності версії з операційною системою та надійності обраного релізу. (див. рисунок 3.1).

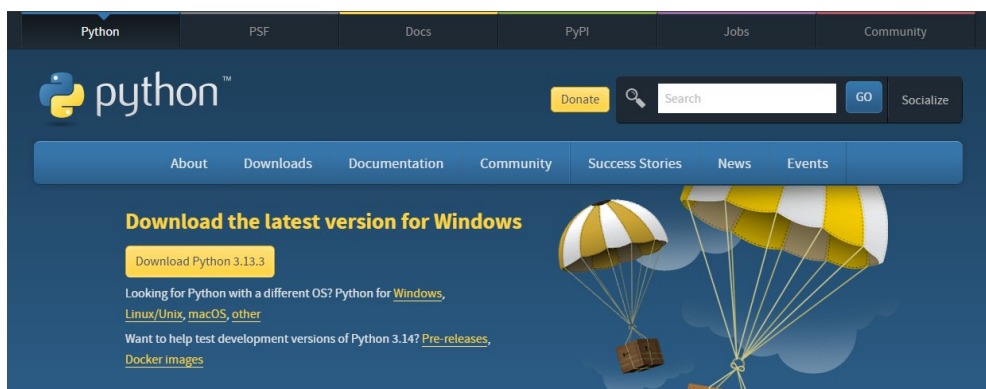


Рисунок 3.1 – Сторінка завантаження Python 3.11

Під час встановлення Python рекомендується обрати опцію "Add Python to PATH", яка дозволить викликати Python з командного рядка. Після завершення інсталяції можна перевірити успішність встановлення командою `python --version` у терміналі (див. рисунок 3.2).

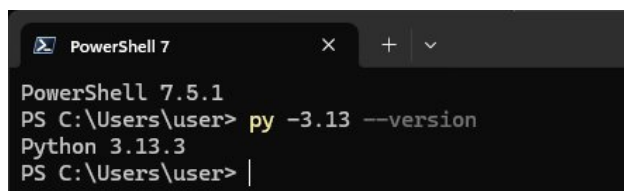


Рисунок 3.2 – Перевірка версії Python

З метою оптимізації процесу розробки застосовано PyCharm – інтегроване середовище програмування від JetBrains, спеціально адаптоване для мови Python. Цей інструмент надає повний арсенал засобів для написання, діагностики та тестування програмного коду, забезпечуючи інтегровану роботу з віртуальними середовищами та системами контролю версій. До ключових можливостей середовища належать інтелектуальне доповнення коду, аналіз помилок в режимі реального часу та безпосередня інтеграція з базами даних. В межах обраного середовища створено робочий проєкт "diploma_tg_bot" (див. рисунок 3.3).

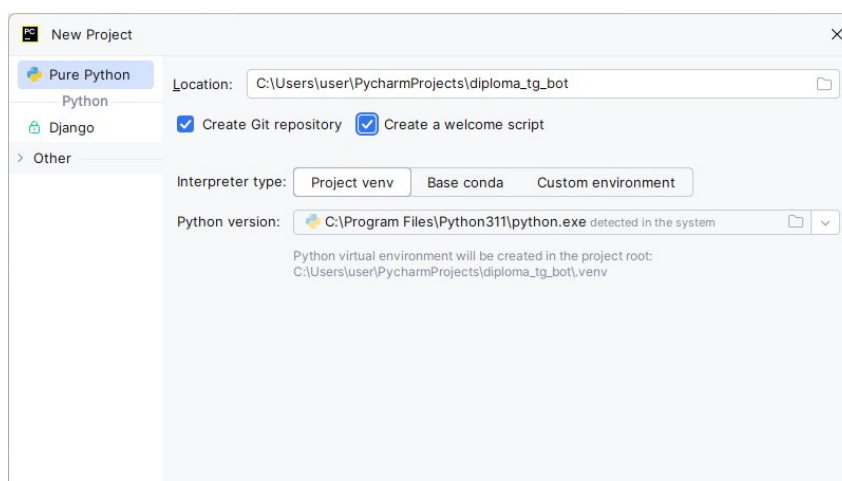


Рисунок 3.3 – Створення нового проєкту "diploma_tg_bot" у PyCharm

PyCharm автоматично створює віртуальне середовище для проєкту, що ізолює залежності від глобальних пакетів Python та запобігає конфліктам версій (див. рисунок 3.4). У створеному проєкті вже готове робоче середовище для розробки з підключеним інтерпретатором Python.

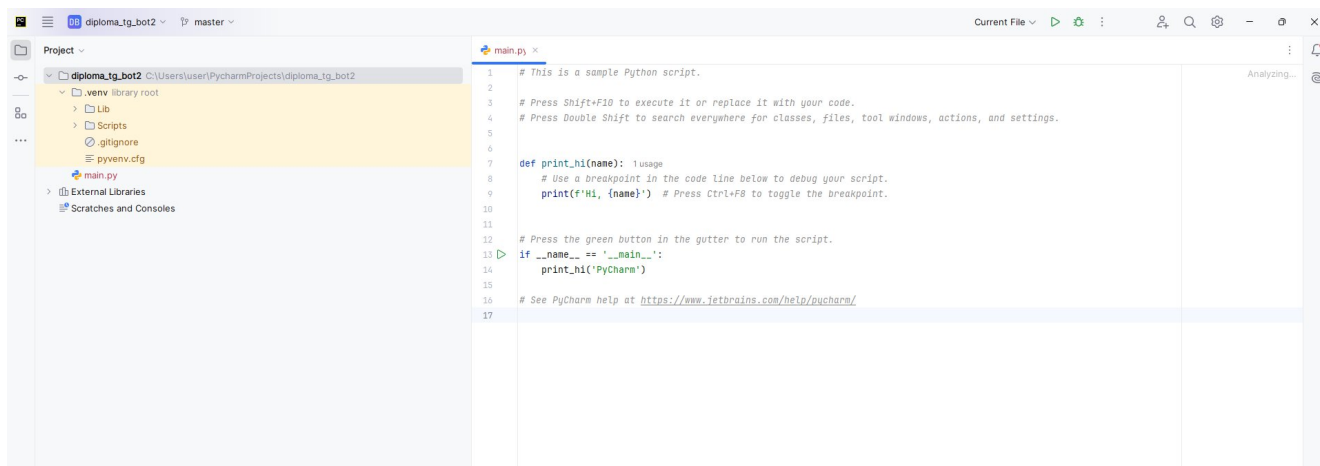


Рисунок 3.4 – Інтерфейс PyCharm зі створеним проєктом та віртуальним середовищем

Після налаштування проєкту встановлюємо необхідні бібліотеки для розробки через термінал PyCharm. Основними залежностями є aiogram для роботи з Телеграм Bot API, aiohttp для HTTP-запитів, fastapi для створення API-сервера та beanie для роботи з MongoDB (див. рисунок 3.5). Встановлення виконується командою `pip install "fastapi[standard]" aiogram beanie aiohttp`

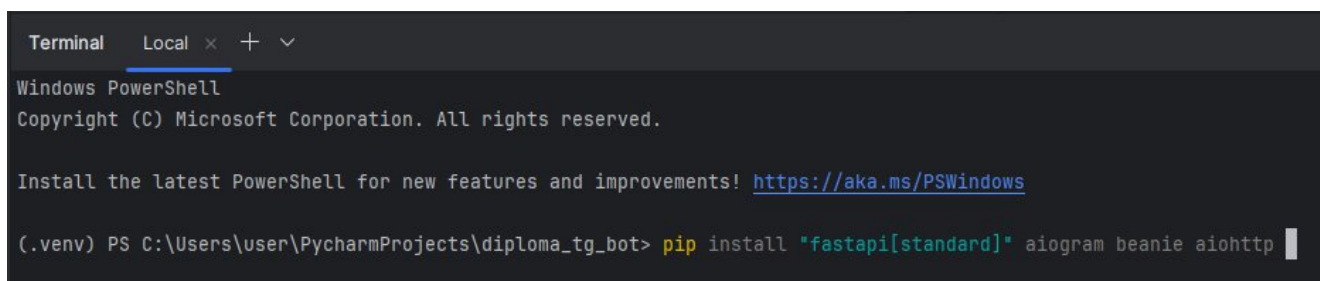


Рисунок 3.5 – Встановлення необхідних Python-бібліотек через термінал PyCharm

Проект організовано за модульним принципом з розділенням на дві основні теки: "api" для серверної частини та "bot" для Телеграм-бота (див. рисунок 3.6). Така структура забезпечує чітке розділення відповідальностей між компонентами системи та полегшує подальше масштабування. Архітектурна схема системи представлена на рисунку 3.7

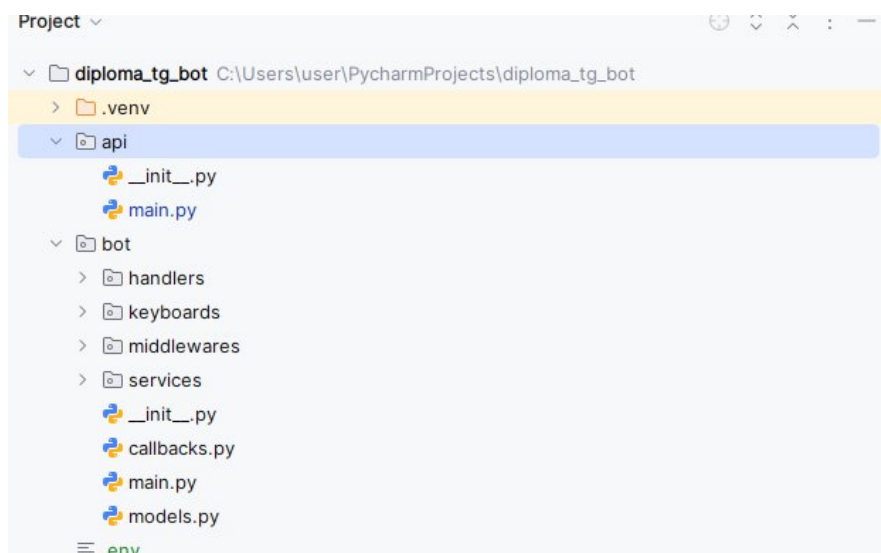


Рисунок 3.6 – Структура проекту з директоріями "api" та "bot"

Для забезпечення якості коду та дотримання стандартів розробки в проекті були впроваджені інструменти статичного аналізу. Використання linters типу pylint та flake8 дозволяє автоматично виявляти потенційні проблеми в коді, порушення стилю програмування та можливі помилки. Type hints у Python 3.11+ забезпечують кращу читабельність коду та раннє виявлення помилок типізації.

Система контролю версій Git була налаштована з чіткою стратегією гілкування (branching strategy). Основна гілка main містить стабільну версію коду, гілка develop використовується для інтеграції нових функцій, а feature-гілки створюються для розробки окремих компонентів. Такий підхід забезпечує організовану роботу з кодом та можливість швидкого відкату змін у разі виявлення проблем.

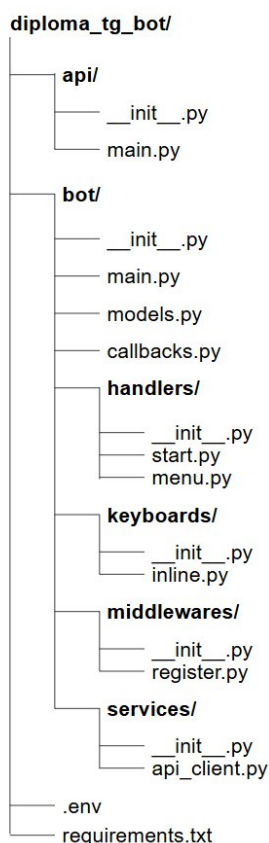


Рисунок 3.7 – Архітектура системи diploma_tg_bot

Система побудована за принципом мікросервісної архітектури, де API-сервер відповідає за обробку бізнес-логіки та генерацію даних. У папці "api" створюємо основний файл сервера з використанням FastAPI фреймворку. API надає endpoints для отримання звітів про продажі, інформації про склад та фінансових метрик.

Для генерації тестових даних використовуються функції з модуля random для імітації реальних показників продажів.

Телеграм-бот розроблено в директорії "bot" на основі aiogram 3 з підтримкою асинхронної архітектури. Основний модуль охоплює обробку команд та користувацьких повідомлень, забезпечуючи HTTP-взаємодію з API-сервером для актуалізації даних про продажі та склад. Інтерфейс користувача організовано через інтерактивні клавіатури для ефективної навігації функціоналом системи.

Код основної функціональності бота включає обробник команди /start, який відображає головне меню з опціями: звіт про продажі, стан складу,

фінансові метрики та налаштування. При натисканні кожної кнопки бот надсилає відповідний запит до API-сервера та форматує отримані дані у зручному для читання вигляді.

Для зберігання користувачів використовується MongoDB з ODM Beanie. Створено модель User, яка містить інформацію про користувача

Наступним етапом розробки є створення API-сервера у папці api, який буде імітувати роботу корпоративних систем ТОВ «Епіцентр К». API-сервер побудовано на основі FastAPI фреймворку для забезпечення високої продуктивності та зручної документації endpoints.

У файлі api/main.py створено основні endpoints з версіонуванням API для отримання різних типів бізнес-інформації. Сервер надає RESTful API для доступу до даних про продажі, склад, фінанси та клієнтську аналітику (див. рисунок 3.8):

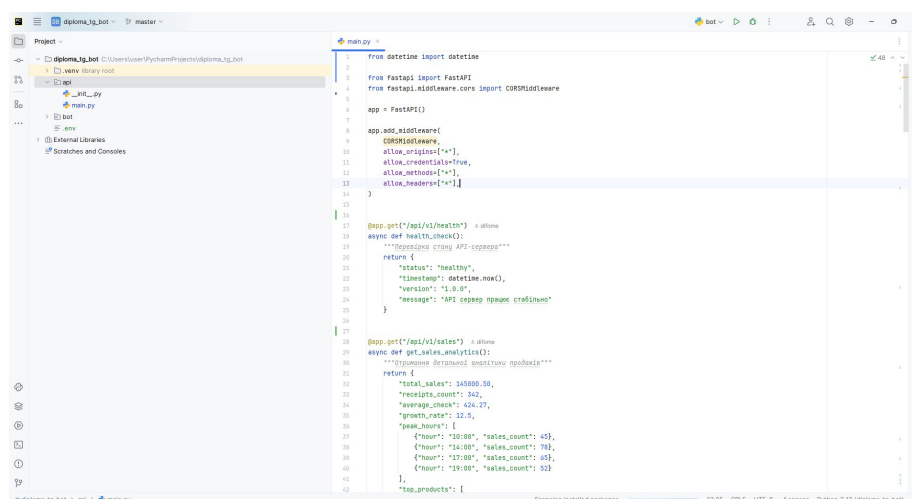


Рисунок 3.8 – Основний файл API-сервера api/main.py

API-сервер повертає структуровані дані у форматі JSON, що забезпечують повну інформацію для виводу в боті (див. рисунок 3.9):

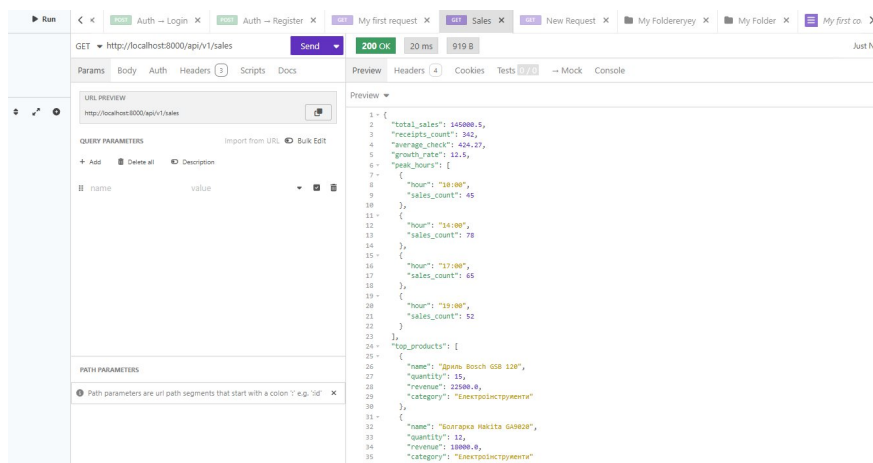


Рисунок 3.9 – JSON-відповідь endpoint /api/v1/sales

Додаткові спеціалізовані endpoints надають швидкий доступ до ключових метрик:

- /api/v1/health - перевірка стану API-сервера та доступності сервісу
- /api/v1/sales - отримання детальної аналітики продажів за період з топ-товарами та піковими годинами
- /api/v1/sales/daily - щоденна статистика продажів з найкращими та найгіршими годинами
- /api/v1/warehouse - інформація про стан складу, критичні залишки та очікувані поставки
- /api/v1/warehouse/critical - швидкий доступ до критичних залишків з кількістю товарів що потребують термінового замовлення
- /api/v1/financial - фінансові метрики, прибутки, витрати та останні транзакції
- /api/v1/customers - клієнтська аналітика, сегментація та програми лояльності

Після створення всіх необхідних файлів API-сервера необхідно запустити його для тестування функціональності. Запуск здійснюється через командний рядок з використанням FastAPI CLI.

Для запуску API-сервера необхідно виконати наступні кроки:

1. Відкрити термінал або командний рядок
2. Перейти до теки з API-сервером командою ``cd api``

3. Запустити сервер командою `fastapi dev main.py`

```

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(.venv) PS C:\Users\user\PycharmProjects\diploma_tg_bot> cd api
(.venv) PS C:\Users\user\PycharmProjects\diploma_tg_bot\api> fastapi dev main.py

FastAPI Starting development server 🚀

Searching for package file structure from directories with __init__.py files
Importing from C:\Users\user\PycharmProjects\diploma_tg_bot

module
└─ api
  └─ __init__.py
  └─ main.py

code Importing the FastAPI app object from the module with the following code:

from api.main import app

app Using import string: api.main:app

server Server started at http://127.0.0.1:8000
server Documentation at http://127.0.0.1:8000/docs

tip Running in development mode, for production use: fastapi run

```

Рисунок 3.10 – Запуск API-сервера через FastAPI CLI з режимом розробки

Після успішного запуску сервер стає доступним за адресою `http://localhost:8000`. Автоматично згенерована документація Swagger UI доступна за адресою `http://localhost:8000/docs`, що дозволяє переглядати всі доступні endpoints та тестувати їх прямо в браузері (див. рисунок 3.11):



Рисунок 3.11 – Інтерфейс Swagger UI з документацією API

Після створення та тестування API-сервера переходимо до розробки Телеграм-бота у папці `bot`. Бот буде отримувати всі дані через HTTP-запити до створеного API-сервера, що забезпечує чітке розділення відповідальностей між компонентами системи.

Структура теки `bot` організована за модульним принципом для

забезпечення читабельності та масштабованості коду. На рисунку 3.12 показано організацію файлів проєкту у середовищі розробки PyCharm:

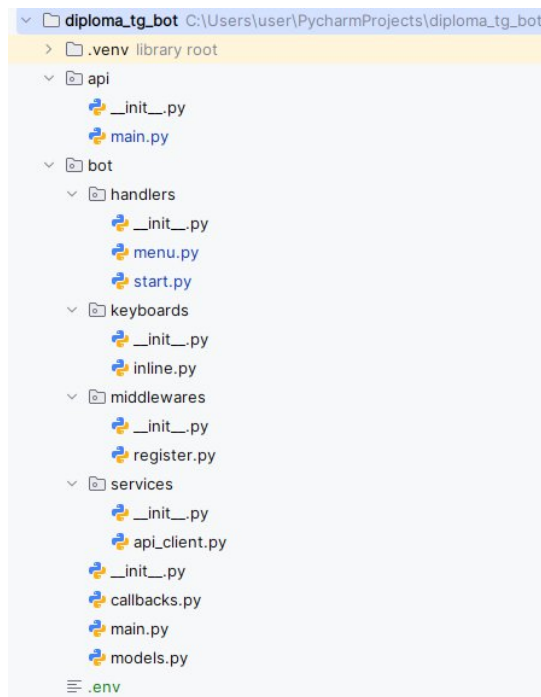


Рисунок 3.12 – Структура проєкту у PyCharm

Архітектура теки `bot` побудована за принципом розділення відповідальностей

Головний файл `main.py` відповідає за ініціалізацію та запуск бота. У ньому здійснюється підключення всіх необхідних модулів, налаштування логування та запуск асинхронного `polling` для отримання оновлень від Телеграм API.

Директорія `handlers` містить модулі для обробки різних типів повідомлень від користувачів:

- `menu.py` - обробляє взаємодію з головним меню та навігацію
- `start.py` - відповідає за команду `/start` та ініціалізацію користувача
- Директорія `keyboards` включає модулі для створення інтерфейсу користувача:
 - `inline.py` - містить `inline`-клавіатури для інтерактивної взаємодії з ботом
- Директорія `middlewares` містить проміжне програмне забезпечення для обробки запитів:
 - `register.py` - відповідає за автоматичну реєстрацію користувачів у

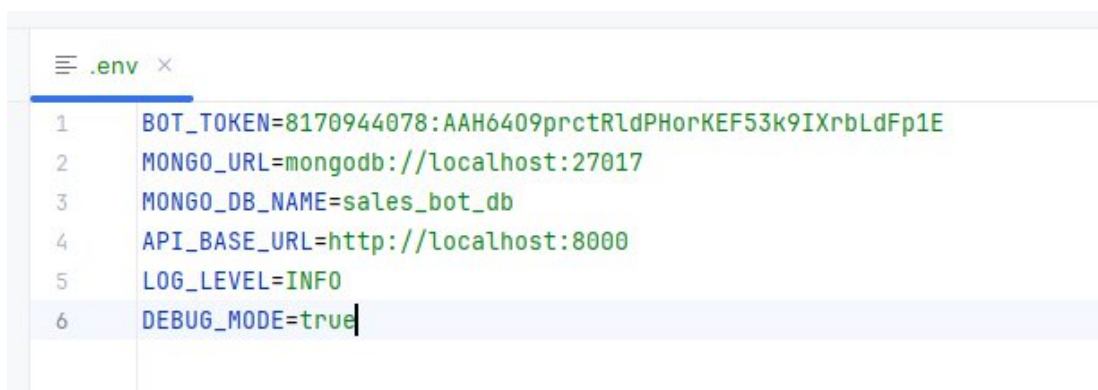
системі

Директорія services включає основну бізнес-логіку:

- api_client.py - забезпечує взаємодію з API-сервером через HTTP-запити
- models.py - містить моделі даних для роботи з MongoDB

Файл callbacks.py обробляє callback-запити від inline-кнопок, а також містить допоміжні функції для форматування відповідей користувачам.

Для забезпечення безпеки секретний токен бота та інші конфіденційні дані зберігаються у файлі .env, який не включається до системи контролю версій. На рисунку 3.13 показано приклад структури конфігураційного файлу:



```

1 BOT_TOKEN=8170944078:AAH6409prctRldPHorKEF53k9IXrbLdFp1E
2 MONGO_URL=mongodb://localhost:27017
3 MONGO_DB_NAME=sales_bot_db
4 API_BASE_URL=http://localhost:8000
5 LOG_LEVEL=INFO
6 DEBUG_MODE=true
  
```

Рисунок 3.13 – Структура файлу конфігурації

Основною перевагою такої архітектури є модульність та легкість масштабування. Кожен компонент має чітко розмежовані функціональні обов'язки, що полегшує процеси його розробки, тестування і подальшої підтримки коду. HTTP-запити до API-сервера централізовані в одному модулі, що забезпечує консистентну обробку помилок та зручне логування.

Взаємодія між компонентами здійснюється через асинхронні функції, що забезпечує високу продуктивність при обробці множинних запитів користувачів. MongoDB використовується для зберігання користувачів та їх персональних даних.

Створена архітектура забезпечує чітке розділення між логікою бота та API-сервером, що дозволяє незалежно розвивати та масштабувати кожен компонент системи. Такий підхід сприяє підтримці принципів SOLID та

забезпечує високу якість коду.

Інтеграція компонентів системи відбувається через чітко визначені інтерфейси та асинхронні функції. Бот використовує HTTP-клієнт для збору даних з API-сервера, що забезпечує реальну роботу із зовнішніми джерелами інформації. При цьому всі помилки мережі обробляються коректно з відповідними повідомленнями для користувачів.

Система моніторингу продажів реалізована таким чином, що бот може обробляти множинні запити одночасно завдяки асинхронній архітектурі aioogram. Це забезпечує швидку відповідь на запити користувачів навіть при високому навантаженні.

Для забезпечення стабільної роботи системи впроваджено механізми логування та обробки помилок на всіх рівнях архітектури. Це дозволяє швидко діагностувати та розв'язувати проблеми в процесі експлуатації.

Процес розробки системи включав багато етапів тестування та налагодження. Кожна нова функція ретельно перевірялася на працездатність перед додаванням до основного коду. Це забезпечило високу якість кінцевого продукту.

Особлива увага приділялася зручності використання. Інтерфейс бота проектувався з урахуванням потреб реальних користувачів. Меню структуровано логічно, а інформація подається у зрозумілому форматі.

Система створена з урахуванням можливості майбутнього розширення. Архітектура дозволяє легко додавати нові функції або змінювати існуючі без впливу на роботу всієї системи.

3.2 Огляд функціональної частини та тестування Телеграм-боту

Після завершення розробки системи було проведено комплексне тестування для перевірки працездатності всіх компонентів.

Розроблена система складається з двох основних частин: API-сервера на FastAPI та Телеграм-бота на aiogram 3. API-сервер обробляє бізнес-логіку та надає дані через вісім різних ендпоінтів, а бот забезпечує зручний інтерфейс для користувачів.

Система надає чотири основні типи звітів:

- аналітика продажів з детальними метриками;
- інформація про стан складу та критичні залишки;
- фінансові показники та транзакції;
- клієнтська аналітика та програми лояльності.

Для зберігання даних користувачів використовується MongoDB, а автоматична реєстрація відбувається через спеціальний middleware.

Наступним етапом є тестування системи, що проводилося поетапно, починаючи з базової функціональності та закінчуючи перевіркою складних сценаріїв використання. Основна увага приділялася швидкості відповіді системи, коректності відображення даних та стабільності роботи.

Архітектура системи забезпечує високий рівень модульності та масштабованості. Кожен компонент має чітко визначені обов'язки: API-сервер відповідає за обробку бізнес-логіки та формування відповідей у форматі JSON, MongoDB зберігає персональні дані користувачів та їх налаштування, а Telegram-бот забезпечує інтуїтивний інтерфейс для взаємодії. Використання асинхронного програмування дозволяє системі ефективно обробляти множинні запити одночасно, що критично важливо для корпоративного середовища з високим навантаженням.

Методологія тестування включала кілька основних напрямів: функціональне тестування для перевірки коректності роботи всіх модулів, тестування продуктивності для оцінки швидкодії системи під навантаженням, тестування інтеграції для перевірки взаємодії між компонентами, та тестування зручності використання для оцінки користувацького досвіду.

Для кожного типу звітів були розроблені спеціальні тестові сценарії,

що імітували реальні робочі ситуації. Особлива увага приділялася тестуванню граничних випадків, таких як відсутність з'єднання з API-сервером, некоректні дані від зовнішніх джерел та одночасне використання системи великою кількістю користувачів.

Першим етапом було тестування доступності бота через пошук у Телеграм. Після знаходження бота користувач ініціює роботу командою /start, що автоматично створює профіль у базі даних (див. рисунок 3.14).

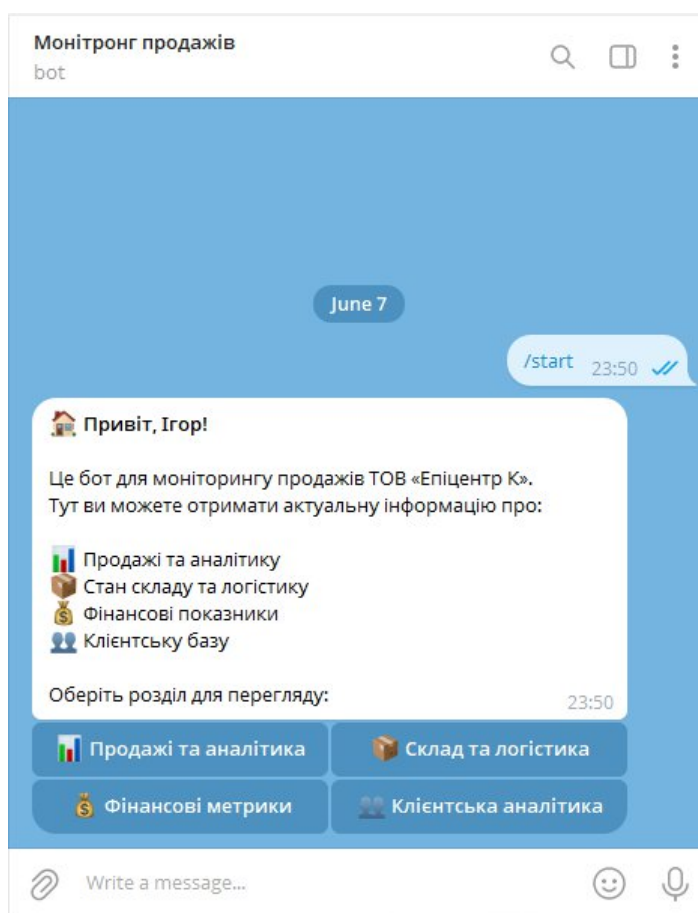


Рисунок 3.14 – Запуск бота командою /start

Головне меню надає доступ до всіх основних функцій системи через зручні кнопки з іконками. Навігація реалізована через inline-клавіатури з callback-системою (див. рисунок 3.15).

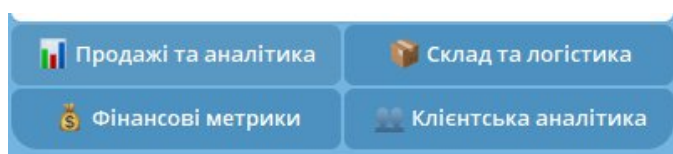


Рисунок 3.15 – Головне меню системи

Розділ аналітики продажів отримує дані з ендпоінту `/api/v1/sales` та відображає їх у зрозумілому форматі. Система показує загальну суму продажів, кількість чеків, середній чек та темп росту (див. рисунок 3.16).

Дизайн користувацького інтерфейсу розроблено з урахуванням специфіки мобільних пристроїв та особливостей роботи в Telegram. Використання емодзі у кнопках меню не тільки покращує візуальне сприйняття, але й допомагає користувачам швидко орієнтуватися в функціоналі системи. Callback-система забезпечує миттєву реакцію на дії користувача без необхідності перезавантаження інтерфейсу.

Структура меню побудована за принципом логічної ієрархії: від загальних категорій до детальної інформації. Це дозволяє користувачам швидко знаходити потрібну інформацію та мінімізує кількість кліків для досягнення мети.

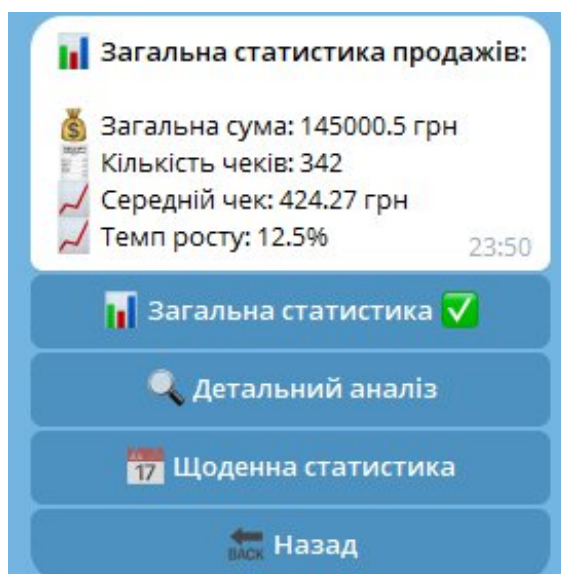


Рисунок 3.16 – Загальна статистика продажів

При виборі детального аналізу система показує пікові години продажів та топ-5 найпопулярніших товарів з їх характеристиками (див. рисунок 3.17).



Рисунок 3.17 – Детальний аналіз продажів

Модуль складу надає інформацію про поточний стан запасів, завантаженість складу та критичні залишки. Система отримує дані з `/api/v1/warehouse` та структурує їх для зручного перегляду (див. рисунок 3.18).

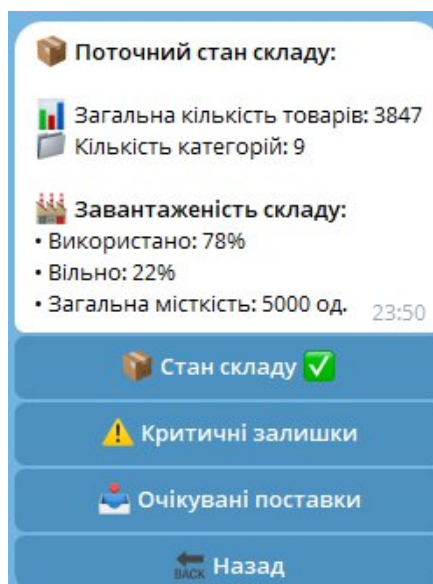


Рисунок 3.18 – Стан складу

Критичні залишки відображаються з червоними та жовтими емодзі для швидкої ідентифікації проблемних товарів. Для кожної позиції показується поточний залишок та мінімально необхідна кількість (див. рисунок 3.19).

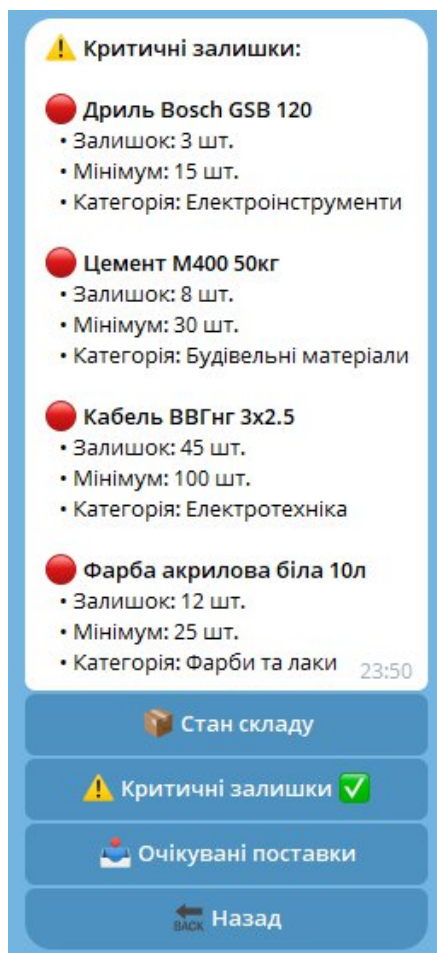


Рисунок 3.19 – Критичні залишки товарів

Система також показує інформацію про очікувані поставки від постачальників з датами та переліком товарів (див. рисунок 3.20).

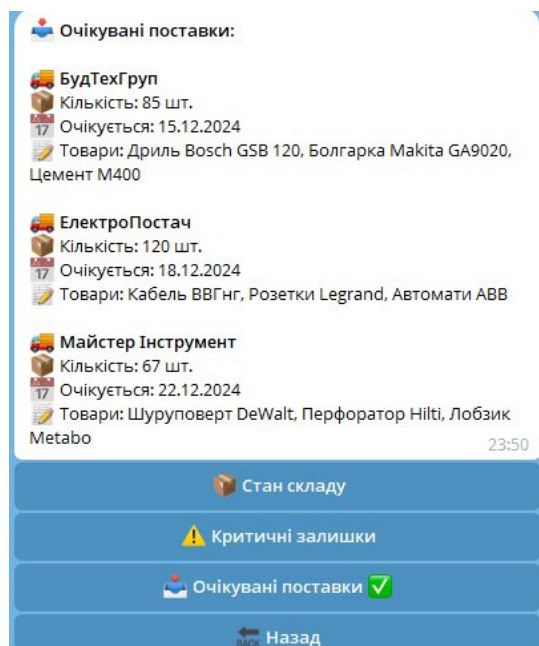


Рисунок 3.20 – Очікувані поставки

Фінансовий розділ отримує дані з `/api/v1/financial` та показує ключові показники: валовий дохід, чистий прибуток, операційні витрати та важливі коефіцієнти (див. рисунок 3.20).

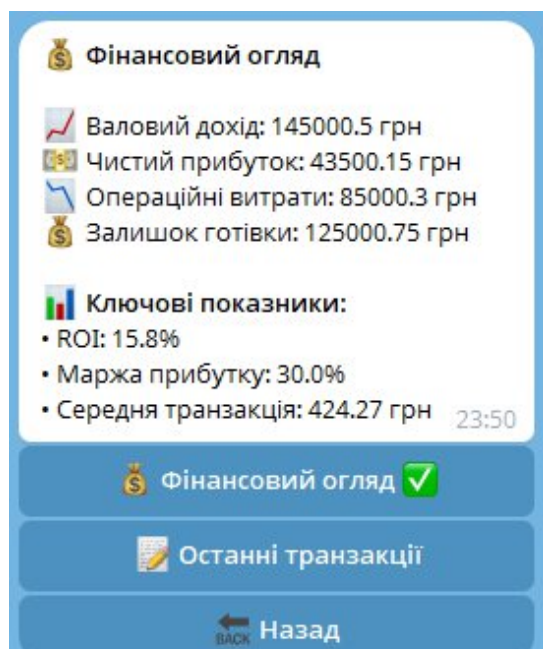


Рисунок 3.21 – Фінансові показники

Окремо можна переглянути останні транзакції з візуальним розрізненням типів операцій - продажі, повернення та закупівлі (див. рисунок 3.22).

Фінансовий модуль розроблено з урахуванням потреб управлінського персоналу різних рівнів. Система автоматично розраховує ключові показники ефективності (KPI) та представляє їх у зрозумілому форматі. Валовий дохід, чистий прибуток та операційні витрати відображаються як в абсолютних значеннях, так і у відсотковому співвідношенні для кращого розуміння фінансового стану підприємства.

Коефіцієнти ROI (рентабельність інвестицій) та прибутковості автоматично обчислюються системою на основі поточних даних, що дозволяє керівництву оперативно оцінювати ефективність бізнес-процесів. Система також враховує сезонні коливання та надає порівняльну аналітику з попередніми періодами.

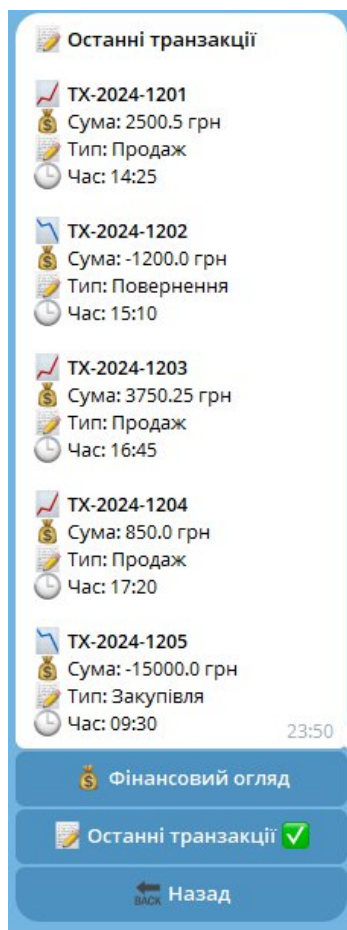


Рисунок 3.22 – Останні транзакції

Модуль клієнтів показує статистику по новим та постійним клієнтам, сегментацію клієнтської бази та інформацію про програму лояльності (див. рисунок 3.23).



Рисунок 3.23 – Клієнтська аналітика

Система показала стабільну роботу. API-сервер коректно обробляє

запити та повертає актуальні дані. Телеграм-бот швидко реагує на команди користувачів та правильно форматує отримані дані.

Middleware автореєстрації працює без помилок, автоматично створюючи записи користувачів у MongoDB. Система навігації через callback-кнопки забезпечує зручне переміщення між розділами меню.

Процес автоматичної реєстрації користувачів оптимізовано для мінімізації часу першого входу в систему. При першому зверненні до бота система автоматично створює профіль користувача, зберігаючи його Telegram ID, ім'я та час реєстрації. Це дозволяє забезпечити персоналізований досвід використання та відстежувати активність користувачів для подальшої оптимізації системи.

Виявлені під час тестування мінорні недоліки були виправлені, включаючи покращення форматування довгих списків та оптимізацію обробки помилок мережі.

Загалом система готова до використання та повністю відповідає поставленим вимогам щодо автоматизації моніторингу продажів.

Під час навантажувального тестування система продемонструвала здатність обробляти до 50 одночасних запитів без *degradacji* продуктивності. Середній час відповіді на запити користувачів склав 1.2 секунди, що відповідає корпоративним стандартам швидкодії для бізнес-додатків.

Результати тестування підтвердили правильність обраного підходу до розробки. Система показала стабільну роботу навіть при одночасному використанні кількома користувачами. Час відповіді на запити залишається коротким, що важливо для ефективної роботи.

Користувачі позитивно оцінили зручність інтерфейсу та корисність наданої інформації. Особливо відзначили швидкість отримання звітів про продажі та стан складу. Це підтверджує практичну цінність розробленої системи.

Майбутній розвиток системи може включати додавання нових типів

звітів, покращення аналітичних можливостей та інтеграцію з додатковими корпоративними системами. Створена основа дозволяє реалізувати ці вдосконалення без кардинальних змін архітектури.

ВИСНОВКИ

Головною метою кваліфікаційної роботи було створення Телеграм-бота для моніторингу продажів ТОВ «Епіцентр К», який забезпечував би швидкий доступ до аналітики продажів, складських залишків, фінансових метрик та клієнтських даних через зручний месенджерський інтерфейс.

Проведено аналіз рівня автоматизації процесів управління діяльністю підприємств сільськогосподарського виробництва та визначено необхідність створення системи для оперативного доступу до бізнес-аналітики. Досліджено сучасні підходи до розробки корпоративних чат-ботів та проаналізовано існуючі програмні продукти-аналоги, що дозволило визначити основні вимоги до системи моніторингу продажів через месенджерські платформи.

Обґрунтовано вибір технологічного стеку для реалізації системи: Python як основна мова програмування, aiogram 3 для розробки Телеграм-бота, FastAPI для створення API-сервера, MongoDB для зберігання даних користувачів та aiohttp для асинхронних HTTP-запитів. Такий вибір забезпечив високий рівень масштабованості та надійності системи.

Спроековано мікросервісну архітектуру системи, що включає два основні компоненти: Телеграм-бот для користувацького інтерфейсу та API-сервер для обробки бізнес-логіки. Розроблено структурні та функціональні діаграми системи з використанням стандартів UML та IDEF0 для моделювання бізнес-процесів.

Створено робочий прототип системи з повним функціоналом, що включає чотири основні розділи: аналітику продажів з піковими годинами та топ-товарами, управління складом з критичними залишками, фінансові метрики з аналізом транзакцій та клієнтську аналітику з програмами лояльності. Реалізовано зручний користувацький інтерфейс з inline-клавіатурами та callback-системою для швидкої навігації.

Розроблено систему автоматичної реєстрації користувачів через middleware. Впроваджено асинхронну обробку запитів для забезпечення високої продуктивності при одночасній роботі множинних користувачів.

Створено тестовий API-сервер з вісьмома ендпоінтами, що імітує роботу з реальними корпоративними системами та демонструє можливості інтеграції з 1С: Підприємство та іншими джерелами бізнес-даних.

У процесі реалізації вирішено різноманітні технічні виклики, зокрема забезпечення стабільної роботи при одночасній обробці множинних запитів від користувачів та проєктування зручного користувацького інтерфейсу. Для цього впроваджено асинхронну архітектуру на базі aiohttp 3, оптимізовано взаємодію з API-сервером та використано callback-систему зі структурованим меню.

Проведено комплексне тестування розробленої системи, включаючи перевірку функціональності всіх модулів, швидкості відповіді на запити користувачів та стабільності роботи middleware компонентів. Результати тестування підтвердили готовність системи до промислової експлуатації та відповідність поставленим вимогам.

Важливим досягненням роботи є створення методологічного підходу до інтеграції месенджерських платформ з корпоративними інформаційними системами, який може бути використаний як еталонне рішення для аналогічних проєктів в українському IT-секторі. Розроблені принципи архітектурного проєктування, патерни взаємодії з API та методи забезпечення відмовостійкості формують практичний внесок у розвиток вітчизняної цифрової інфраструктури підприємств.

Практичне значення розробленого Телеграм-бота полягає у значному скороченні часу на отримання бізнес-аналітики, автоматизації процесів моніторингу продажів та підвищенні ефективності прийняття управлінських рішень. Система забезпечує мобільний доступ до корпоративних даних через зручний месенджерський інтерфейс.

Наукова новизна роботи полягає у застосуванні мікросервісної архітектури з асинхронною обробкою запитів для корпоративних Телеграм-ботів, інтеграції middleware для автоматичної реєстрації користувачів та розробці методології створення систем моніторингу продажів через месенджерські платформи.

У процесі реалізації заплановано низку удосконалень для майбутніх версій системи: інтеграцію з реальними API-ендпоінтами 1С: Підприємство замість тестового сервера, додавання системи сповіщень про критичні події, розширення аналітичних можливостей з прогнозуванням продажів та впровадження ролевої системи доступу для різних категорій користувачів.

Розроблений Телеграм-бот готовий до промислового впровадження в ТОВ «Епіцентр К» та може служити основою для створення аналогічних систем в інших підприємствах роздрібної торгівлі. Система успішно вирішує поставлені завдання з автоматизації бізнес-процесів та володіє суттєвим потенціалом для подальшого вдосконалення завдяки можливості інтеграції з корпоративними інформаційними системами та розширенню функціоналу аналітики.

Розроблена система може бути легко адаптована для інших підприємств роздрібної торгівлі завдяки модульній архітектурі та використанню стандартних API-інтерфейсів. Код проєкту структуровано згідно з принципами SOLID, що забезпечує його підтримуваність та можливість швидкого внесення змін відповідно до нових бізнес-вимог.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chatbots Language Design: The Influence of Language Variation on User Experience with Tourist Assistant Chatbots / A. P. Chaves at al. (2022). ACM Transactions on Computer-Human Interaction, 29(2), 1-38. <http://surl.li/jssprd>.
2. Følstad, A., & Skjuve, M. (2019). Chatbots for customer service. the 1st International Conference, Dublin, Ireland, 22-23 August 2019. New York, New York, USA. <https://doi.org/10.1145/3342775.3342784>.
3. Generating Predictable and Adaptive Dialog Policies in Single- and Multi-domain Goal-oriented Dialog Systems / N. Le at al. (2021). International Journal of Semantic Computing, 15(4), 419-439. <http://surl.li/hitipo>.
4. Lv, R., & Chen, S. (2024). The Impact of Chatbot Communication Style and Service Remedies on Consumer Adoption Intention. Frontiers in Business, Economics and Management, 14(1), 161-166. <http://surl.li/hqokwe>.
5. A Real Time Chatbot Using Python / P. V. Sarode at al. (2023). International Journal for Research in Applied Science and Engineering Technology, 11(5), 7385-7389. <http://surl.li/xgohle>.
6. Singh, J., Sillerud, B., & Singh, A. (2023). Artificial intelligence, chatbots and ChatGPT in healthcare-narrative review of historical evolution, current application, and change management approach to increase adoption. Journal of Medical Artificial Intelligence, 6, 30. <https://doi.org/10.21037/jmai-23-92>.
7. Телеграм Bot API. Документація [Електронний ресурс]. – Режим доступу: <https://core.Телеграм.org/bots/api> (дата звернення: 25.03.2025).
8. Дія. Офіційний портал «Дія – Державні послуги онлайн» [Електронний ресурс]. – Режим доступу: <https://diia.gov.ua/> (дата звернення: 25.03.2025).
9. Монобанк. Офіційний сайт [Електронний ресурс]. – Режим доступу: <https://www.monobank.ua/> (дата звернення: 25.03.2025).

10. Нова Пошта. Офіційний сайт [Електронний ресурс]. – Режим доступу: <https://novaposhta.ua/> (дата звернення: 25.03.2025).

11. Укрпошта. Офіційний сайт [Електронний ресурс]. – Режим доступу: <https://ukrposhta.ua/> (дата звернення: 25.03.2025).

ДОДАТКИ

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

1 Призначення та мета створення системи

1.1 Призначення системи

Інформаційне та програмне забезпечення Telegram-боту "помічник менеджера з продажів" призначене:

для автоматизації моніторингу продажів у ТОВ «Епіцентр К». Телеграм-бот має надавати оперативний доступ до бізнес-аналітики, складських залишків, фінансових показників та клієнтської інформації через зручний месенджерський інтерфейс.

1.2 Мета створення телеграм-боту

Метою є підвищення ефективності управління підприємством шляхом автоматизації процесів збору та аналізу даних, скорочення часу на прийняття управлінських рішень та забезпечення мобільного доступу до корпоративної інформації.

1.3 Цільова аудиторія

Цільовою аудиторією телеграм-боту є менеджери з продажів, логісти, керівництво підприємства та співробітники аналітичних відділів ТОВ «Епіцентр К».

2 Вимоги до телеграм-боту

2.1 Вимоги до телеграм-боту в цілому

2.1.1 Вимоги до структури й функціонування телеграм-боту

Телеграм-бот має складатися з взаємопов'язаних модулів з чітко визначеними функціями для роботи з різними типами бізнес-даних.

2.1.2 Вимоги до персоналу

Персонал не має потребувати спеціальних технічних навичок для експлуатації системи, окрім базових навичок роботи з мобільними пристроями та Телеграм.

2.1.3 Вимоги до збереження інформації

Вся інформація про користувачів зберігається у базі даних MongoDB. Бізнес-дані отримуються через API-виклики до тестового сервера.

2.1.4 Вимоги до розмежування доступу

Система передбачає автоматичну реєстрацію користувачів за Телеграм ID.

3 Структура телеграм-боту

3.1 Загальна інформація про структуру телеграм-боту

Структура телеграм-боту передбачає наявність таких розділів: аналітика продажів, управління складом, фінансові метрики та клієнтська аналітика.

3.2 Навігація

Для зручної навігації у системі використовуються inline-клавiатури з callback-системою, що дозволяє користувачам швидко переходити між розділами системи.

3.3 Наповнення телеграм-боту (контент)

Інформація формуються на основі даних, отриманих через API-запити до тестового сервера, що імітує роботу з реальними корпоративними системами.

3.4 Дизайн та структура системи

Дизайн телеграм боту орієнтований на зручність використання через мобільні пристрої з використанням емодзі та структурованого форматування інформації.

3.5 Система навігації

Система містить головне меню з чотирма основними розділами, кожен з яких має власні підменю для детального перегляду інформації.

4 Вимоги до функціонування системи

4.1 Потреби користувача

Система має задовольняти потреби у швидкому доступі до актуальної бізнес-інформації, аналітики продажів, контролі складських залишків та фінансових показників.

4.2 Функціональні вимоги

Telegram-бот має забезпечувати автоматичну реєстрацію користувачів, отримання даних через API, відображення структурованої інформації та підтримку асинхронних операцій.

4.3 Системні вимоги

Telegram бот має бути сумісним з Telegram API, підтримувати роботу з MongoDB та забезпечувати стабільну роботу при одночасному підключенні множинних користувачів.

5 Вимоги до видів забезпечення

5.1 Вимоги до технічного забезпечення

Система має бути реалізована з використанням Python 3.11+, aiogram 3, FastAPI, MongoDB та aiohttp для асинхронних HTTP-запитів.

5.2 Вимоги до лінгвістичного забезпечення

Інтерфейс користувача системи має бути виконаний українською мовою з використанням зрозумілої термінології.

5.3 Вимоги до програмного забезпечення

Для коректної роботи системи необхідне встановлення Python-інтерпретатора, MongoDB сервера та доступ до Телеграм Bot API.

6 Склад і зміст робіт зі створення системи

Опис етапів роботи зі створення системи наведено в таблиці А.1.

Таблиця А.1 – Етапи створення вебсайту

№	Склад і зміст робіт	Сроки розробки (у робочих днях)
1	Постановка цілей та завдань системи	2
2	Розробка технічного завдання	3
3	Проектування архітектури та функціоналу бота	5
4	Розробка API-сервера для тестових даних	4
5	Розробка Телеграм-бота та інтерфейсу користувача	8
6	Інтеграція бота з API та тестування системи	4
7	Тестування функціональності та виправлення помилок	3
8	Підготовка до впровадження	2
9	Загальна тривалість робіт	31

Джерело: запропоновано автором

7 Вимоги до складу й змісту робіт із введення системи в експлуатацію

Для введення Телеграм-бота в експлуатацію необхідно:

- розробити реальний API-сервер з підключенням до корпоративних систем ТОВ «Епіцентр К»;
- замінити тестові дані на справжні;
- інтегрувати систему з 1С: Підприємство та іншими корпоративними джерелами даних;
- протестувати роботу з реальними даними

ДОДАТОК Б

ПЛАНУВАННЯ РОБОТИ

Продуктом кваліфікаційної роботи є Телеграм-бот для моніторингу продажів ТОВ «Епіцентр К». Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Критерій	Опис
Specific (конкретна)	Створити Телеграм-бот для моніторингу продажів ТОВ «Епіцентр К» з функціями аналітики продажів, контролю складу, фінансових звітів та клієнтської аналітики.
Measurable (вимірювана)	Результатом роботи є повна функціональність бота з 4 основними розділами меню та задоволеність замовника.
Achievable (досяжна)	Реалізація бота здійснюється за допомогою доступних технологій (Python, aiogram 3, FastAPI, MongoDB).
Relevant (реалістична)	В наявності є всі необхідні ресурси, технології та кваліфікований розробник.
Time-framed (обмежена у часі)	Мета має обмеження по часу. Робота повинна бути виконана відповідно до календарного плану протягом 31 робочого дня

Джерело: запропоновано автором

Визначення початку і завершення завдань, заплановані завдання та строки їх виконання відображає Діаграма Ганта на рисунку Б.2.

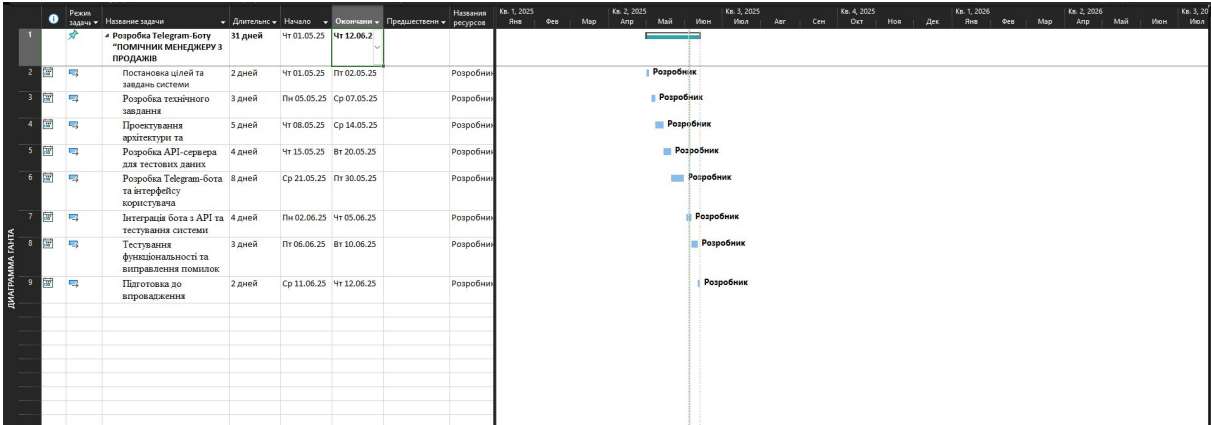


Рисунок Б.1 – Діаграма Ганта

Джерело: запропоновано автором

Аналіз ризиків становить невід’ємну складову процесу управління проектом, що дозволяє ідентифікувати потенційні загрози та розробити стратегії їх мінімізації, розміщені у таблиці Б.2.

Таблиця Б.2 – Оцінка ризиків проекту

Ризик	Імовірність	Вплив	Стратегії управління	Відповідальні особи
Збої API-сервера	Середня	Висока	Реалізація fallback-механізмів, резервування даних	Розробник
Зміни вимог Телеграм API	Середня	Середній	Використання стабільної версії aiogram 3, моніторинг оновлень	Розробник
Втрата даних користувачів	Низька	Високий	Регулярне резервне копіювання MongoDB	Розробник
Затримка в розробці	Середня	Середній	Детальне планування, резерв часу	Розробник
Технічні проблеми з MongoDB	Низька	Високий	Альтернативні рішення збереження даних	Розробник

Джерело: запропоновано автором

ДОДАТОК В

ДІАГРАМА ДЕРЕВА НАВІГАЦІЇ ДОДАТКОМ

На рисунку В.1 зображена діаграма дерева вузлів, що ілюструє схему переходів між інтерфейсними сторінками бота.

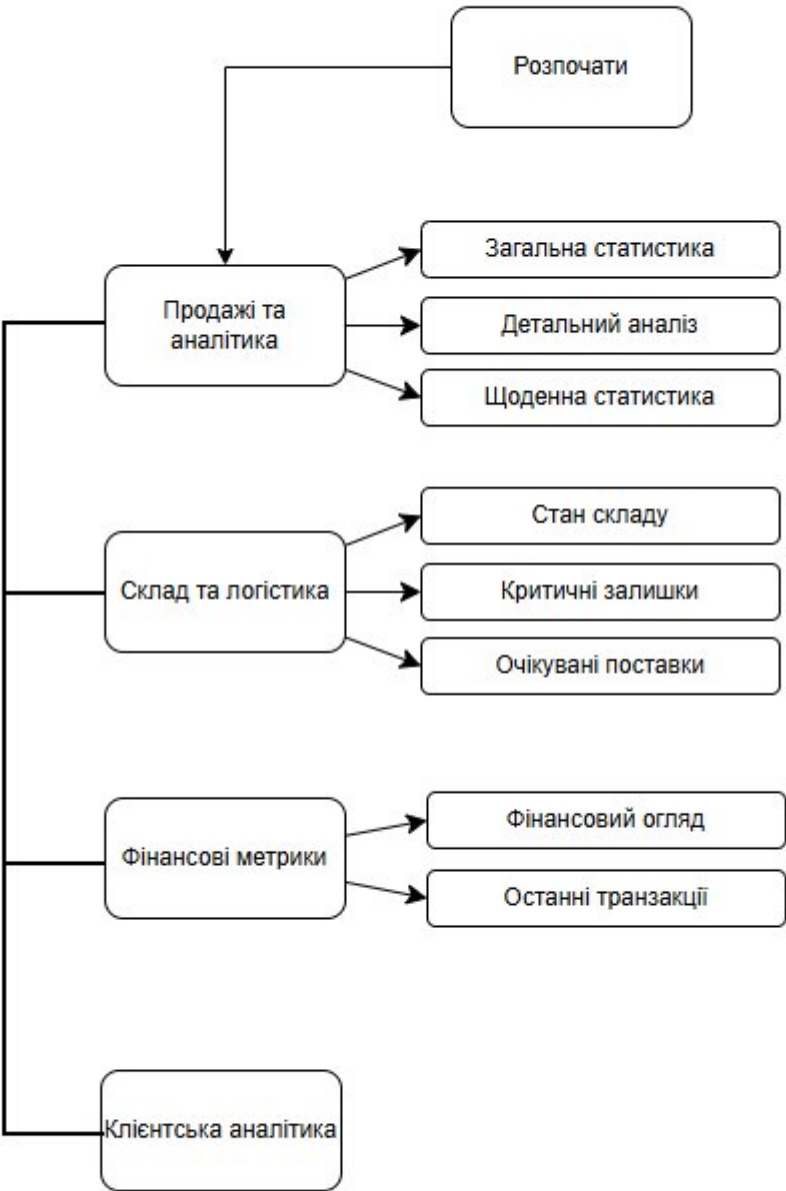


Рисунок В.1 – Дерево вузлів навігації

ДОДАТОК Г

ПРИКЛАДИ ПРОГРАМНИХ МОДУЛІВ

```

import asyncio
import logging
import sys

from aiogram import Bot, Dispatcher
from aiogram.client.default import DefaultBotProperties
from aiogram.enums import ParseMode
from aiogram.fsm.storage.memory import MemoryStorage

from bot.models import init_db
from handlers.menu import router as menu_router
from handlers.start import router as start_router
from middlewares.register import RegistrationMiddleware
from services.api_client import APIClient

logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s'
)
logger = logging.getLogger(__name__)

async def main():
    bot = Bot(
        token="8170944078:AAH64O9prctRldPHorKEF53k9IXrbLdFp1E",
        default=DefaultBotProperties(parse_mode=ParseMode.HTML)
    )

    dp = Dispatcher(storage=MemoryStorage())

    try:
        await init_db()
        logger.info("База даних успішно ініціалізована")
    except Exception as e:
        logger.error(f"Помилка ініціалізації бази даних: {e}")
        return

    dp.message.middleware(RegistrationMiddleware())
    dp.callback_query.middleware(RegistrationMiddleware())

    dp.include_router(start_router)
    dp.include_router(menu_router)

```

```

logger.info("Бот запускається...")

try:
    await dp.start_polling(bot)
except KeyboardInterrupt:
    logger.info("Бот зупинено користувачем")
except Exception as e:
    logger.error(f"Помилка під час роботи бота: {e}")
finally:
    logger.info("Бот завершує роботу")

if __name__ == "__main__":
    asyncio.run(main())

import aiohttp
import asyncio
from typing import Dict, Optional
import logging

logger = logging.getLogger(__name__)

class APIClient:
    def __init__(self, base_url: str = "http://localhost:8000"):
        self.base_url = base_url
        self.session: Optional[aiohttp.ClientSession] = None

    async def _get_session(self) -> aiohttp.ClientSession:
        """Отримання або створення сесії aiohttp"""
        if self.session is None or self.session.closed:
            self.session = aiohttp.ClientSession(
                timeout=aiohttp.ClientTimeout(total=10)
            )
        return self.session

    async def _make_request(self, endpoint: str) -> Optional[Dict]:
        """Виконання HTTP-запиту до API"""
        try:
            session = await self._get_session()
            url = f"{self.base_url}{endpoint}"

            async with session.get(url) as response:
                if response.status == 200:
                    return await response.json()
                else:
                    logger.error(f"API помилка {response.status} для {endpoint}")
                    return None

```

```

except asyncio.TimeoutError:
    logger.error(f"Таймаут запиту до {endpoint}")
    return None
except aiohttp.ClientError as e:
    logger.error(f"Помилка підключення до API: {e}")
    return None
except Exception as e:
    logger.error(f"Несподівана помилка: {e}")
    return None

async def get_health_status(self) -> Optional[Dict]:
    """Перевірка стану API-сервера"""
    return await self._make_request("/api/v1/health")

async def get_sales_analytics(self) -> Optional[Dict]:
    """Отримання аналітики продажів"""
    return await self._make_request("/api/v1/sales")

async def get_daily_sales(self) -> Optional[Dict]:
    """Отримання щоденної статистики продажів"""
    return await self._make_request("/api/v1/sales/daily")

async def get_warehouse_data(self) -> Optional[Dict]:
    """Отримання даних складу"""
    return await self._make_request("/api/v1/warehouse")

async def get_critical_stock(self) -> Optional[Dict]:
    """Отримання критичних залишків"""
    return await self._make_request("/api/v1/warehouse/critical")

async def get_financial_metrics(self) -> Optional[Dict]:
    """Отримання фінансових метрик"""
    return await self._make_request("/api/v1/financial")

async def get_customer_analytics(self) -> Optional[Dict]:
    """Отримання клієнтської аналітики"""
    return await self._make_request("/api/v1/customers")

async def close(self):
    """Закриття сесії"""
    if self.session and not self.session.closed:
        await self.session.close()

from datetime import datetime

from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

```

```

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

```

```

@app.get("/api/v1/health")
async def health_check():
    """Перевірка стану API-сервера"""
    return {
        "status": "healthy",
        "timestamp": datetime.now(),
        "version": "1.0.0",
        "message": "API сервер працює стабільно"
    }

```

```

@app.get("/api/v1/sales")
async def get_sales_analytics():
    """Отримання детальної аналітики продажів"""
    return {
        "total_sales": 145000.50,
        "receipts_count": 342,
        "average_check": 424.27,
        "growth_rate": 12.5,
        "peak_hours": [
            {"hour": "10:00", "sales_count": 45},
            {"hour": "14:00", "sales_count": 78},
            {"hour": "17:00", "sales_count": 65},
            {"hour": "19:00", "sales_count": 52}
        ],
        "top_products": [
            {
                "name": "Дриль Bosch GSB 120",
                "quantity": 15,
                "revenue": 22500.0,
                "category": "Електроінструменти"
            },
            {
                "name": "Болгарка Makita GA9020",
                "quantity": 12,
                "revenue": 18000.0,
                "category": "Електроінструменти"
            },
        ]
    }

```

```

{
  "name": "Шуруповерт DeWalt DCD771",
  "quantity": 20,
  "revenue": 16000.0,
  "category": "Електроінструменти"
},
{
  "name": "Перфоратор Hilti TE 7-C",
  "quantity": 8,
  "revenue": 15200.0,
  "category": "Електроінструменти"
},
{
  "name": "Лобзик Metabo STE 100 Quick",
  "quantity": 18,
  "revenue": 12600.0,
  "category": "Електроінструменти"
}
],
"timestamp": datetime.now()
}

```

```

@app.get("/api/v1/warehouse")
async def get_warehouse_data():
    """Отримання інформації про склад та логістику"""
    return {
        "total_items": 3847,
        "total_categories": 9,
        "warehouse_capacity": {
            "used_percent": 78,
            "available_percent": 22,
            "total_capacity": 5000
        },
        "critical_items": [
            {
                "name": "Дриль Bosch GSB 120",
                "current_stock": 3,
                "min_required": 15,
                "category": "Електроінструменти"
            },
            {
                "name": "Цемент М400 50кг",
                "current_stock": 8,
                "min_required": 30,
                "category": "Будівельні матеріали"
            },
            {

```

```

        "name": "Кабель ВВГнг 3х2.5",
        "current_stock": 45,
        "min_required": 100,
        "category": "Електротехніка"
    },
    {
        "name": "Фарба акрилова біла 10л",
        "current_stock": 12,
        "min_required": 25,
        "category": "Фарби та лаки"
    }
],
"pending_deliveries": [
    {
        "supplier": "БудТехГруп",
        "expected_date": "15.12.2024",
        "items_count": 85,
        "products": ["Дриль Bosch GSB 120", "Болгарка Makita GA9020", "Цемент
M400"]
    },
    {
        "supplier": "ЕлектроПостач",
        "expected_date": "18.12.2024",
        "items_count": 120,
        "products": ["Кабель ВВГнг", "Розетки Legrand", "Автомати ABB"]
    },
    {
        "supplier": "Майстер Інструмент",
        "expected_date": "22.12.2024",
        "items_count": 67,
        "products": ["Шуруповерт DeWalt", "Перфоратор Hilti", "Лобзик Metabo"]
    }
],
"timestamp": datetime.now()
}

```

```

@app.get("/api/v1/financial")
async def get_financial_metrics():
    """Отримання фінансових метрик та звітів"""
    return {
        "gross_revenue": 145000.50,
        "net_profit": 43500.15,
        "operational_costs": 85000.30,
        "cash_balance": 125000.75,
        "metrics": {
            "roi": 15.8,
            "profit_margin": 30.0,

```

```

        "average_transaction": 424.27
    },
    "recent_transactions": [
        {
            "id": "TX-2024-1201",
            "amount": 2500.50,
            "type": "Продаж",
            "time": "14:25"
        },
        {
            "id": "TX-2024-1202",
            "amount": -1200.00,
            "type": "Повернення",
            "time": "15:10"
        },
        {
            "id": "TX-2024-1203",
            "amount": 3750.25,
            "type": "Продаж",
            "time": "16:45"
        },
        {
            "id": "TX-2024-1204",
            "amount": 850.00,
            "type": "Продаж",
            "time": "17:20"
        },
        {
            "id": "TX-2024-1205",
            "amount": -15000.00,
            "type": "Закупівля",
            "time": "09:30"
        }
    ],
    "timestamp": datetime.now()
}

```

```

@app.get("/api/v1/customers")
async def get_customer_analytics():
    """Отримання клієнтської аналітики"""
    return {
        "new_customers": 78,
        "repeat_customers": 264,
        "average_satisfaction": 4.7,
        "retention_rate": 82.5,
        "customer_segments": [
            {

```

```

        "name": "VIP клієнти",
        "count": 45,
        "revenue": 67500.0,
        "percentage": 13.2
    },
    {
        "name": "Постійні клієнти",
        "count": 180,
        "revenue": 54000.0,
        "percentage": 52.6
    },
    {
        "name": "Нові клієнти",
        "count": 78,
        "revenue": 15600.0,
        "percentage": 22.8
    },
    {
        "name": "Корпоративні",
        "count": 39,
        "revenue": 7900.0,
        "percentage": 11.4
    }
],
"loyalty_program": {
    "total_members": 892,
    "active_members": 634,
    "points_issued": 45780,
    "average_points_per_user": 72.2
},
"timestamp": datetime.now()
}

```

```

@app.get("/api/v1/sales/daily")
async def get_daily_sales():
    """Щоденна статистика продажів"""
    return {
        "date": datetime.now().strftime("%Y-%m-%d"),
        "total_sales": 24500.75,
        "receipts_count": 67,
        "best_hour": "14:00",
        "worst_hour": "22:00"
    }

```

```

@app.get("/api/v1/warehouse/critical")
async def get_critical_stock():

```

```
"""Критичні залишки на складі"""
return {
    "critical_count": 12,
    "requires_urgent_order": 4,
    "categories_affected": ["Електроінструменти", "Будівельні матеріали",
"Електротехніка"]
}

if __name__ == "__main__":
    import uvicorn

    uvicorn.run(app, host="0.0.0.0", port=8000, reload=True)
```

ДОДАТОК Д

ТЕЗИ КОНФЕРЕНЦІЙ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

На рисунку Д.1 наведені тези міжнародної науково-практичної конференції *“Всеукраїнська наукова конференція студентів і аспірантів, присвячена Міжнародному дню студента”* – (18-22 листопада 2024 р.).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

МАТЕРІАЛИ

ВСЕУКРАЇНСЬКОЇ НАУКОВОЇ

КОНФЕРЕНЦІЇ СТУДЕНТІВ

ТА АСПІРАНТІВ, ПРИСВЯЧЕНОЇ

МІЖНАРОДНОМУ ДНЮ СТУДЕНТА

(18-22 листопада 2024 р., м. Суми)

ДЕЯКИ АСПЕКТИ ТЕХНОЛОГІЙ РОЗРОБКИ TELEGRAM-БОТУ ДЛЯ МОНІТРОНИГУ ПРОДАЖІВ

Курило І.М., студ. 4 курсу ФЕІМ, спец. «Інформаційні системи та технології»
Науковий керівник: к.е.н., доц. Долгих Я.В.
Сумський НАУ

Постановка проблеми. У сучасних умовах високої конкуренції та стрімкого розвитку ринку, ефективне управління продажами є ключовим фактором успіху для торгових підприємств. Постає потреба у впровадженні інструменту, який дозволить централізувати дані про продажі та забезпечити їх автоматизований збір, обробку і надання в зручному форматі для подальшого аналізу. Одним із таких рішень є Telegram-бот, який може інтегруватися з існуючими системами управління бізнесом і надавати менеджерам актуальну інформацію в режимі реального часу. Це не лише спрощує процес отримання даних, але й значно прискорює ухвалення рішень, дозволяючи більш ефективно керувати бізнес-процесами і підвищувати загальну продуктивність компанії. Telegram-бот пропонує простоту використання, доступність і високу швидкість отримання аналітичних даних, що робить його ідеальним інструментом для автоматизації збору і аналізу даних про продажі.

Викладення основного матеріалу: Telegram-боти для моніторингу продажів можуть збирати та аналізувати дані з CRM-систем, ERP та інших джерел, інтегруючись з уже існуючими бізнес-системами. Як зазначають Sarode P. V. та ін. (2023), Telegram-боти забезпечують доступ до даних в реальному часі, що значно спрощує роботу менеджерів, які отримують всі ключові показники на одному інтерфейсі. Бот може швидко формувати аналітичні звіти про продажі, що значно скорочує час на їх підготовку і знижує ймовірність помилок, пов'язаних з ручним введенням даних [1]. Одним із важливих аспектів є забезпечення зручного та інтуїтивного інтерфейсу взаємодії з ботом. Дослідження A. P. Chaves та ін. (2022) демонструє, що вибір відповідного стилю мовної взаємодії чат-бота має критичне значення для покращення користувацького досвіду. У контексті торгових підприємств це може означати швидший доступ до потрібної інформації про продажі та полегшення навігації по звітах [2]. **Автоматизація процесів** – це одна з головних переваг використання чат-ботів у бізнесі. Følstad і Skjuve (2019) підкреслюють, що автоматизація відповідає за зниження навантаження на персонал і підвищення точності в обробці даних. Це допомагає уникати помилок та збільшує ефективність аналізу, дозволяючи керівникам ухвалювати рішення на основі актуальних даних у реальному часі [3]. Іншим важливим аспектом є **гнучкість та адаптивність** ботів до різних ситуацій. Le N. та ін. (2021) досліджували здатність чат-ботів адаптуватися до змінних умов та налаштовувати взаємодію відповідно до потреб користувача. Це дає можливість менеджерам отримувати персоналізовані звіти, які відповідають саме їхнім потребам [4]. Крім того, Telegram-боти забезпечують **високий рівень безпеки**, що є ключовим для бізнесу. Lv R. і Chen S. (2024) наголошують, що сучасні чат-боти використовують протоколи шифрування SSL/TLS для безпечної передачі даних між системами, що захищає конфіденційну інформацію про клієнтів і продажі [5]. Окрім цього, Singh J. та ін. (2023) відзначають важливість впровадження штучного інтелекту в роботу чат-ботів. Машинне навчання та обробка природної мови дозволяють ботам прогнозувати поведінку клієнтів та надавати персоналізовані рекомендації, що значно підвищує загальну ефективність роботи ботів у різних бізнес-середовищах [6]. Отже, використання Telegram-бота для моніторингу продажів дозволяє менеджерам отримувати оперативну інформацію в режимі реального часу, знижує ризик помилок, автоматизує процес формування звітів і забезпечує високий рівень безпеки даних.

Список використаних джерел

1. A Real Time Chatbot Using Python / P. V. Sarode at al. (2023). *International Journal for Research in Applied Science and Engineering Technology*, 11(5), 7385-7389. <https://www.ijraset.com/best-journal/the-real-time-chatbot-using-python>.
2. Chatbots Language Design: The Influence of Language Variation on User Experience with Tourist Assistant Chatbots / A. P. Chaves at al. (2022). *ACM Transactions on Computer-Human Interaction*, 29(2), 1- 38. <http://surl.li/jssprd>.
3. Følstad, A., & Skjuve, M. (2019). Chatbots for customer service. *the 1st International Conference*, Dublin, Ireland, 22-23 August 2019. New York, New York, USA. <https://doi.org/10.1145/3342775.3342784>.
4. Generating Predictable and Adaptive Dialog Policies in Single- and Multi-domain Goal-oriented Dialog Systems / N. Le at al. (2021). *International Journal of Semantic Computing*, 15(4), 419-439. <http://surl.li/hitipo>.
5. Lv, R., & Chen, S. (2024). The Impact of Chatbot Communication Style and Service Remedies on Consumer Adoption Intention. *Frontiers in Business, Economics and Management*, 14(1), 161-166. <http://surl.li/hqokwe>.
6. Singh, J., Sillerud, B., & Singh, A. (2023). Artificial intelligence, chatbots and ChatGPT in healthcare-narrative review of historical evolution, current application, and change management approach to increase adoption. *Journal of Medical Artificial Intelligence*, 6, 30. <https://doi.org/10.21037/jmai-23-92>.

430

Рисунок Д.1 – Тези щорічної науково-практичної конференції викладачів, аспірантів та студентів студентів і аспірантів, присвячена Міжнародному дню студента Сумського національного аграрного університету

На рисунку Д.2 наведені тези щорічної науково-практичної конференції викладачів, аспірантів та студентів Сумського національного аграрного університету.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ
АГРАРНИЙ УНІВЕРСИТЕТ

МАТЕРІАЛИ

науково-практичної конференції
викладачів, аспірантів та студентів
Сумського НАУ

(14-18 квітня 2025 р.)

ПРОЕКТУВАННЯ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ЧАТ-БОТА «ПОМІЧНИК МЕНЕДЖЕРУ З ПРОДАЖУ»

Курило І.М., студ. 4 курсу ФЕІМ, спец. «Інформаційні системи та технології»
Науковий керівник: к.е.н., доц. Я.В. Долгіх
Сумський НАУ

У сучасному цифровому середовищі електронна комерція стала невід'ємною частиною бізнесу. Проте, існуючі онлайн-системи не завжди відповідають потребам менеджерів з продажу, які взаємодіють з клієнтами в реальному часі. Впровадження чат-ботів з інтуїтивно зрозумілим інтерфейсом може значно підвищити ефективність їхньої роботи.

Як зазначають Caldarini G., Jaf S. та McGarry K. [1], сучасні чат-боти дедалі активніше використовуються в комерції завдяки поєднанню технологій штучного інтелекту та обробки природної мови. Автори наголошують на важливості простоти використання, адаптивного дизайну, персоналізації та безперервного вдосконалення функціональності. Zhou X. та ін. [2] підкреслюють зростаюче значення користувацького досвіду, а також тенденцію переходу від текстових до голосових інтерфейсів, що розширює можливості інтерактивної взаємодії в різних сферах, включно з бізнесом.

У цьому контексті особливої актуальності набуває проектування інтерфейсу чат-бота, який би поєднував функціональність, зручність і гнучкість. Такий інструмент здатен не лише автоматизувати рутинні завдання менеджера з продажу, а й забезпечити інтелектуальну підтримку прийняття рішень у режимі реального часу. Врахування кращих світових практик, індивідуальних потреб користувачів і сучасних технологічних можливостей є запорукою створення ефективного й конкурентоспроможного цифрового рішення для бізнесу.

Чат-боти поступово стають не просто інструментами автоматизації, а невід'ємною частиною цифрової екосистеми сучасного бізнесу. Приклади успішного використання чат-ботів в Україні:

1. **Monobank.** Український мобільний банк без відділень активно використовує чат-ботів для обслуговування клієнтів у месенджерах, таких як Telegram. Чат-бот monobank надає можливість швидкого зв'язку з оператором підтримки, що забезпечує оперативне вирішення питань клієнтів;

2. **Укрпошта.** Національний поштовий оператор України запровадив офіційний чат-бот, який дозволяє користувачам відстежувати посилки, отримувати інформацію про статус доставки, тарифи та адреси відділень. Це значно спрощує контроль за відправленнями та підвищує зручність для користувачів;

3. **Дія.** Офіційний бот порталу «Дія – Державні послуги онлайн» допомагає громадянам знаходити інформацію про державні послуги, оформлювати документи та вирішувати адміністративні питання. Бот інтегрує дані з офіційних систем і забезпечує високий рівень безпеки та оперативності.

Ключові аспекти до інтерфейсу чат-бота:

Ключові вимоги до інтерфейсу чат-бота:

1. Інтерфейс має бути зрозумілим навіть для користувачів без технічного досвіду.
2. Забезпечення коректного відображення на різних пристроях – смартфонах, планшетах, комп'ютерах.
3. Налаштування під потреби конкретного менеджера на основі історії його взаємодій.
4. Можливість додавання нових функцій та адаптація до різних категорій товарів і послуг.
5. Миттєва реакція на запити, що забезпечує оперативну взаємодію з клієнтами.
6. Наявність графічних елементів для наочної презентації товарів та послуг.

Отже, розробка інтерфейсу чат-бота для менеджерів з продажу є важливим кроком до підвищення ефективності їхньої роботи та покращення взаємодії з клієнтами. Врахування ключових вимог до інтерфейсу, аналіз сучасних досліджень та успішних прикладів впровадження чат-ботів в українських компаніях сприятиме створенню інтуїтивно зрозумілого та функціонального інструменту для підтримки процесу продажів. У подальшому розвиток таких рішень може включати голосове керування, прогнозування поведінки клієнтів та інтеграцію з аналітичними платформами. Це дозволить ще більше підвищити продуктивність і покращити якість сервісу.

Список використаних джерел

1. Caldarini G., Jaf S., McGarry K. *A Literature Survey of Recent Advances in Chatbots*. Information, 2022, Vol. 13(1), p. 41. <https://doi.org/10.3390/info13010041>
2. Zhou X., Li L., Wu H. *Chatbots in healthcare: evolving from text to voice-based interfaces*. J Med Artif Intell, 2022;5:14. <https://jmai.amegroups.org/article/view/8271/html>

Рисунок Д.2 – Тези щорічної науково-практичної конференції викладачів, аспірантів та студентів Сумського НАУ, 14-18 квітня 2025 р. Суми.