

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ  
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ  
Кафедра прикладної інформатики**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_ Гамаюн В.П.  
(підпис) (ПІБ)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021р.

**ДИПЛОМНИЙ ПРОЕКТ  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”**

**Тема:** Інтегрування чат-ботів в систему продажів

**Виконавець:** \_\_\_\_\_ Масний Максим Вікторович  
(підпис) (ПІБ)

**Керівник:** \_\_\_\_\_ Толстікова Олена Володимирівна  
(підпис) (ПІБ)

**Нормоконтролер:** \_\_\_\_\_ Боровик Володимир Миколайович  
(підпис) (ПІБ)

**Київ 2021**

## ВСТУП

У час масового розвитку інформаційних технологій (ІТ) з'являється можливість спостерігати стрімкий процес росту у всіх галузях, де можна застосувати програмування та проектування систем. Не виключенням стала сфера продажів та надання послуг.

Зростання бізнесу залежить від доречного використання інформаційних технологій для забезпечення ефективної підтримки ключових бізнес-процесів компанії. Питання про збільшення автоматизації роботи в компанії є одним із головних, на що впливає розвиток технічного прогресу.

ІТ фірми та корпорації переорієнтовуються на розвиток і об'єднання бізнес-планів та інформаційних стратегій. Це, як наслідок, сприяє утворенню нових програмних продуктів, що дає поштовх до розвитку самої компанії та світових методик ведення корпоративних процесів.

Одним із бізнес-процесів, який підлягає автоматизації, є технічна підтримка та спілкування з клієнтами. За статистикою ІВМ, підприємства витрачають 1,3 трильйони доларів на 265 мільярдів звернень в технічну підтримку компаній[1]. Ефективним зменшенням цих витрат є використання чат-ботів в обслуговуванні клієнтів. Це рішення пришвидшить час відгуку на питання покупця та надасть агентам з продажів можливість зосередитися на більш складній роботі.

Мета чат-ботів – підтримка та розширення ділових команд у стосунках із клієнтами. Його можна використовувати в додатках для спілкування, таких як Facebook Messenger, Telegram, Viber, Slack, WhatsApp.

У дипломній роботі потрібно дослідити питання інтеграції чат-ботів в систему продажів компанії та розглянути можливі рішення для створення віртуального сервісу підтримки зв'язку з клієнтами. Провести дослідження сучасних методів їх реалізації.

Відповідно до основних функціональних можливостей чат-ботів, які набирають популярність у найбільших компаніях по усьому світі, варто

провести аналіз структури системи, сфер використання, класифікацій та запропонувати абстрактну структуру чат-боту. Сервіс повинен відповідати необхідним вимогам для розгортання системи та функціонування її в структурі організації.

Розглянути важливі для роботи чат-боту рішення та платформу, на якій буде розроблятися програмне забезпечення, проаналізувати і обрати архітектуру та інструменти для побудови системи.

Програмний продукт повинен вирішувати такі питання:

- оптимізація процесів, які повторюються з великою частотою - вирішення рутинних питань, що дає змогу працівникам компанії займатись більш трудомісткими задачами. Це призводить до більш стрімкого розвитку та росту бізнесу;

- зворотній зв'язок з кінцевим споживачем - бот може відповідати на питання цілодобово, що підвищує його перевагу перед кол-центром. Добре пропрацьований у цьому плані сервіс визначає успішність бізнесу в області B2C (Business to Consumer - Бізнес для споживача);

- продажі - чат-бот, який веде зв'язок з клієнтом через сайт або месенджер, здатен не тільки на шаблонні відповіді, а й на збільшення кількості нових клієнтів та заключених угод по наданню послуг.

Чат-бот повинен бути спроектованим так, щоб мати змогу масштабуватися, розширювати функціональні можливості, підтримувати високу швидкість відповіді та бути відмовостійким до великих навантажень від користувачів.

# РОЗДІЛ 1

## ОСНОВНІ ПОНЯТТЯ ТА ЗАСТОСУВАННЯ ЧАТ-БОТІВ

### 1.1. Що таке чат-бот та його основні функціональні можливості

Чат-бот (*англ. chatbot*) - це програмний продукт, який дає змогу моделювати реальну розмову з клієнтом [2]. Цей віртуальний компаньйон представляє собою автоматизовану систему спілкування з користувачами та допомагає підприємцям стати ближче зі своїми покупцями (рис. 1.1). Окрім того, швидкість опрацювання заявки, яку залишив клієнт, чат-ботом допомагає завойовувати його лояльність та бажання купувати саме в цій компанії.

Більшість ботів запрограмовані на певні ланцюги розвитку подій, що ведуть до певного результату та надання запрограмованих даних користувачеві. Але досягнення в області штучного інтелекту (ШІ) в комбінації з розвитком месенджерів сприяють їх розвитку.

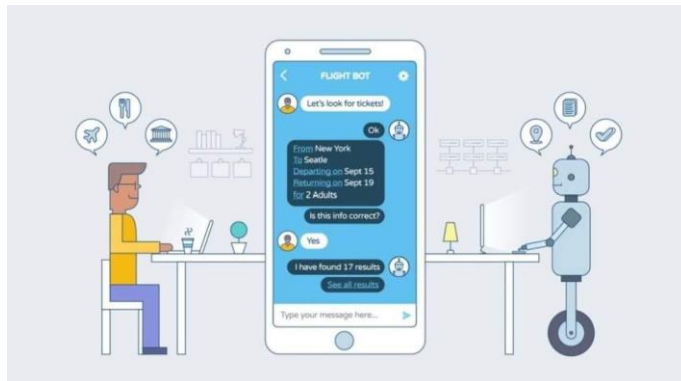


Рис. 1.1. Наглядний приклад взаємодії клієнта з чат-ботом

Основною ціллю використання цього програмного продукту є допомога клієнтам якомога швидше досягти своєї кінцевої мети: отримання додаткової інформації чи продаж товару або послуг. Зрештою, бота можна запрограмувати так, щоб він міг цілодобово повторювати успіх агентів по роботі з клієнтами.

Кафедра ПІ				НАУ 21 16 78 – 000 ПЗ			
Виконав	Масний М.В.			Основні поняття та застосування чат-ботів	Лім.	Арк.	Аркушів
Керівник	Толстікова О.В.					10	16
Консульт.					122 ТП-416Б		
Нормоконт	Боровик В.М.						
Зав. каф.	Гамаюн В.П.						

Для правильного підходу потрібно зрозуміти, які є способи інтегрування чат-бота у необхідний проект чи бізнес та які приховані від звичайного користувача «підводні камені» має в собі цей метод [3]. Основними базовими питаннями можуть бути:

- визначення очікуваного результату - потрібно розуміти, що використання чат-бота не може, поки що, замінити повністю використання людей в обслуговуванні покупців, але може звести деякі задачі до вагомого мінімуму. Потрібно звернути увагу, що для користувача платформою важливо вести «живу» бесіду, тому не варто намагатися видати діалог з ботом за реальну людину і дати змогу переключитись на менеджера;

- ґрунтовно пізнати тему роботи з клієнтами - необхідно персоналізувати взаємодію з покупцем та централізувати інформацію про обслуговування користувачів, щоб вони мали змогу легко себе ідентифікувати для повторного використання «віртуального» менеджера. В іншому випадку це може призвести тільки до негативного досвіду від роботи з платформою;

- спрямувати увагу на дані - зберігання та збір, а потім і аналіз даних надає можливість відслідковувати тенденції. Це допоможе бренду створювати клієнтські профілі, які допомагають ще більше персоналізувати підходи до комунікації з клієнтами, а також отримувати дані по навігації сайтом, мобільним додатком або ж по самому чат-боту;

- продумати дизайн та структуру - створення ботів має певні шаблони та правила, на які потрібно звернути увагу для покращення досвіду їх використання у покупців. Орієнтація на знання розробки сайтів та мобільних додатків є помилковою у будіванні свого віртуального помічника, тому варто приділити особливу увагу цій темі перед початком роботи над ним. Необхідно розглянути питання: «Як має звучати ваша компанія з уст клієнта?», ретельно обміркувати особистий бренд.

## **1.2. Використання чат-ботів**

Використання чат-ботів має певні плюси для автоматизації сервісу обслуговування покупців [4], а саме:

1) забезпечення компанією технічної підтримки та цілодобового сервісного обслуговування для клієнтів. Оформлення замовлень в режимі онлайн, а тим більше 24/7, вагомо виділяє компанію серед інших конкурентів;

2) перенесення на чат-бота відповіді на популярні питання зменшує час очікування відповіді для користувача, що також позитивно впливає на довіру до магазину чи будь-якого бізнесу;

3) допомагають охоплювати більше клієнтів. Використання месенджерів присутнє у вагомій кількості людей, тому застосування такого інструменту, як чат-бот, є цілком доцільним з точки зору збільшення об'єму продажів. Щоб залучити якомога більший об'єм нового трафіку клієнтів чат-бот повинен відповідати необхідним для цього вимогам:

- працювати без перебоїв та помилок;
- відповідати особистому бренду або компанії;
- забезпечувати використання лінгвістики та правил комунікування;
- мати чіткі та зрозумілі задачі для виконання;
- надавати достовірну інформацію та FAQ;
- працювати на різних пристроях.

4) плідна співпраця з «живими» менеджерами. Чат-боти опрацьовують обмежений спектр питань, тому при необхідності і при виникненні ситуації, з якою програма матиме труднощі, варто визначити людину-агента та перенаправити її на рішення даного питання (див. рис. 1.2). Тому сміливо можна сказати, що боти в певній мірі звільняють консультанта від дрібної роботи та допомагають переключитися на більш складні та термінові задачі;



Рис. 1.2. Передання чат-ботом складної ситуації менеджеру

5) Допомагають зменшити операційні витрати на співробітників компанії. Для більшості компаній, а тим більше молодих, мати цілодобову підтримку є занадто затратним, а саме чат-бот може вирішити це питання;

6) Надають сучасні та прогресивні інструменти для інтернет-маркетингу та продажів. Добре продумана структура віртуального менеджера може сприяти заключенню угод та закриттю продажів за допомогою використання надійних маркетингових стратегій. І це надає додаткові можливості:

- чат-бота можна налаштовувати на встановлення індивідуальної взаємодії з певним клієнтом, даючи надійні та перевірені поради, орієнтуючись на минулий досвід співпраці та продажів даному покупцеві;
- чат-боти надають змогу надавати повний супровід по воронці продажів за певними маркетинговими рекомендаціями та не заставляють клієнта очікувати в разі виникнення певних питань;
- є можливість налаштувати чат-бота певним чином, щоб він мав змогу передбачати всі кроки, які клієнт може зробити на шляху отримання бажаного товару, та його поведінку при покупці. Це дає змогу відправити потрібні акційні пропозиції тільки відповідним користувачам, що збільшує цільове попадання для заключення угоди (див. рис. 1.3);

- також чат-боти можуть охоплювати величезну кількість клієнтів через соціальні мережі, на що не здатна ніяка людина, яка співпрацює з клієнтами;
- впровадження такого автоматизованого помічника допоможе підвищити рівень цінності бренду та компанії в очах покупців. На сам перед тому, що при взаємодії з клієнтами бот може збирати інформацію про самого клієнта та його відгуки, після чого дані аналізуються і, в залежності від результату, в майбутньому підбирається певний ефективний підхід до комунікації з користувачем платформи для більшого його задоволення від співпраці.
- чат-боти не тільки надають певні рекомендації та допомогу у вирішенні питань, які могли виникнути при використанні платформи, а й впроваджують продажі прямо в месенджері, що зменшує кількість відволікаючих факторів для клієнта і сприяє найлегшому закриттю угоди;

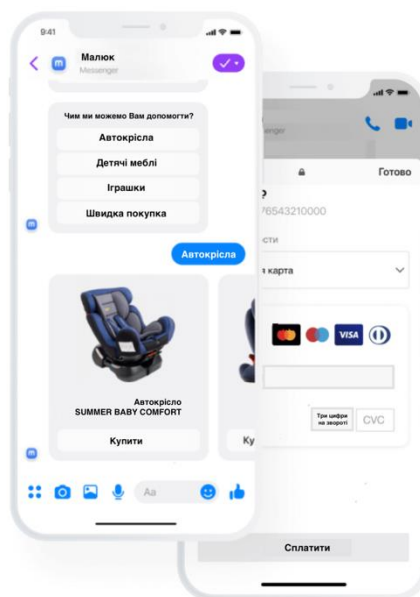


Рис. 1.3. Приклад інтеграції онлайн-оплати в чат-бота

7) На даному етапі розвитку прикладного програмного інтерфейсу месенджерів можна підключити платформи для здійснення онлайн-оплат в роботу бота;



8) Покращення роботоспроможності та функціональності відбувається за рахунок аналізу оцінки та оптимізації. Роботоспроможність та продуктивність покращуються за рахунок спостережень за такими напрямками; досвід користувача, якість та зручність використання, його лінгвістичні вміння. Покращення чат-бота має такі аспекти:

- привітальні та запрошувальні розсилки;
- додавання нової та сезонної продукції;
- різноманітні інформаційні блоки для поширення бренду;
- актуальні та вигідні знижки.

Наразі використання чат-ботів є хорошою альтернативою великому відділу продажів та технічній підтримці, тому може здаватись привабливою ідеєю автоматизації. Проте даний метод несе свої ризики і потрібно якомога відповідальніше поставитись до створення віртуального помічника. До мінусів використання ботів належать [5]:

- не всі бізнес-моделі підтримують оптимізацію за допомогою даного методу. Для початку потрібно проаналізувати свою компанію, продукт та аудиторію, тому що не завжди впровадження даного ПЗ призводить до покращення продажів чи комунікації з покупцями;
- втрата сенсу відгуків від клієнтів через використання віртуальної техпідтримки, яка не відразу реагує на актуальні проблеми користувачів;
- одним з недоліків є невміння розпізнавати команди, які не були закладені в основу логіки бота. Люди можуть висловлюватись не точно по відношенню до товару чи послуги, в такому разі це буде викликати труднощі у наданні допомоги;
- відсутність емоцій є важливим бар'єром у спілкуванні з клієнтами. «Холодний розум» не завжди приносить позитивні наслідки, тому боти втрачають можливість діяти в залежності від емоцій, що може відштовхувати та лякати покупців;
- необхідність в постійній оптимізації. Від розвитку та оптимізації сервісу буде залежати його відповідність вимогам клієнтів та його надійна

працездатність. Слідкування за правильністю інформації, що надається клієнтам, і належність представлення бренду користувачам є трудомісткою по об'єму роботи задачею. По мірі змін в актуальності ринку та товарів, необхідно аналізувати досвід комунікації з користувачами та виявити нагальні питання, які варто вирішити.

### **1.3. Аналіз сфер використання чат-ботів**

Спектр використання чат-ботів є дуже широким, адже їх можна використовувати для більшості бізнес-моделей. Особливо корисними вони є для компаній, яким важливо інформувати клієнтів про важливі заходи, акції або про технічні неполадки на сервері. Боти є зручним для використання інструментом поширення інформації та підтримки лояльності покупців до особистого бренду фірми.

Підтримку чат-ботом можна реалізувати на сайті компанії або ж у месенджері. Додатки для спілкування є широко розповсюдженим в наш час і кількість людей, що їх використовує, збільшується[6].

Чат-боти в залежності від алгоритму роботи бувають двох типів:

- ті, що обмежені запрограмованим числом команд та відповідають на конкретні запити покупців;
- ті, в яких використані нейронні мережі та ШІ, що дозволяє їм удосконалювати свої вміння вести бесіду з користувачем.

Другий тип ботів завойовує більшу прихильність серед покупців, тому що діалог з ними приймає вигляд живої розмови.

Також боти можуть поділятися за можливістю взаємодії з користувачем. Вони можуть мати різну форму комунікації: кнопкова форма ведення комунікації, текстова та вбудовані (inline) боти [7].

В перших за основу взята форма спілкування, коли людина при користуванні натискає кнопки, а бот в свою чергу опрацьовує інформацію, яка закладена в цих кнопках (рис. 1.4). Вони є найпопулярнішими, які використовують в месенджерах.

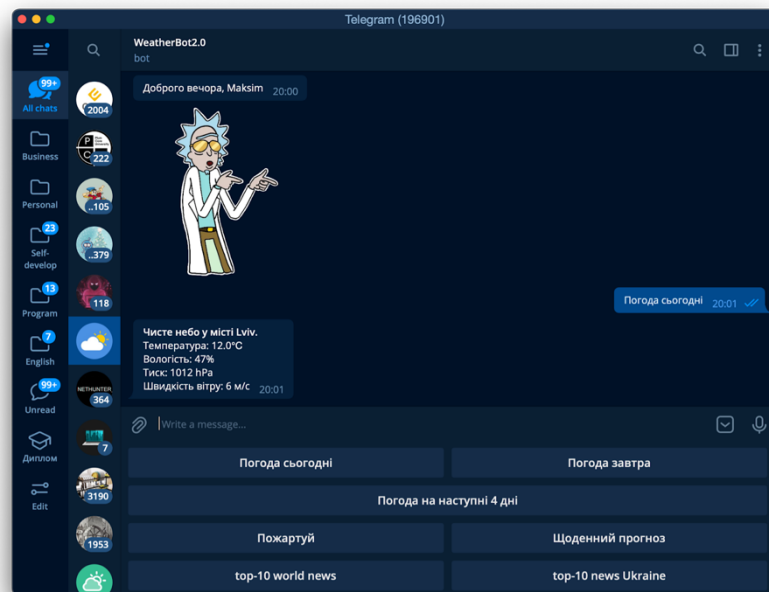


Рис. 1.4. Кнопковий бот

Другі - більш складні для опрацювання команд, тому що боту потрібно знайти слова, на які потрібно реагувати. Їх реалізують за допомогою регулярних виразів пошуку в тексті необхідної інформації для надання відповіді (див. рис. 1.5). Розвинутіші чат-боти використовують нейронні мережі для ведення “живої” розмови.

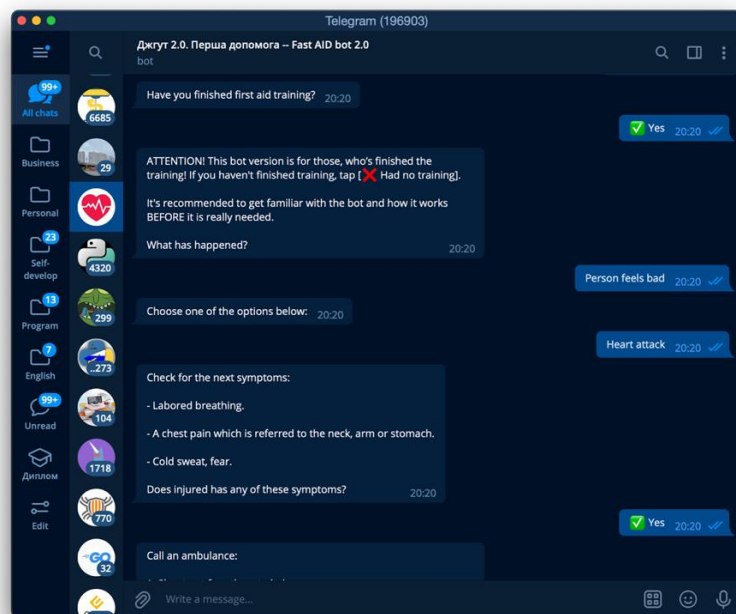


Рис. 1.5. Текстовий бот

Третій варіант - це вбудовані боти, які використовують в бесіді з кимось, вони запрограмовані таким чином, що користувач має доступ до нього за допомогою введення в поле для повідомлення посилання на цього бота (див. рис. 1.6).

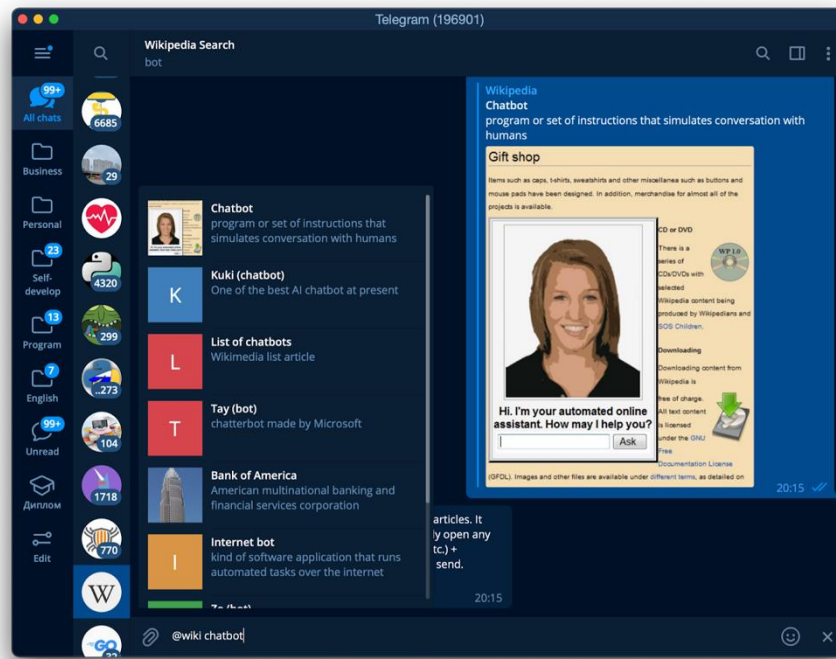


Рис. 1.6. Вбудовані (inline) боти

Класифікація чат-ботів за сферою використання:

1) Інтернет-магазин - це один із найпопулярніших варіантів використання ботів в бізнесі. Він виконує функцію автономного менеджера з продажу, якому не потрібно платити заробітну плату або чекати з лікарняного. За допомогою такого інструменту можна продавати товари і послуги цілодобово. Достатньо лише продумати всі питання покупців і запрограмувати на них відповіді. Але, щоб бот не просто відповідав на запити, а й продавав, до нього підключають платіжні системи, кошик і з'єднання з реальним консультантом. Так можна охопити ще більшу аудиторію, збільшити продажі та прибуток компанії.

2) Агрегатор - допомагає вибрати послуги в будь-якій категорії. Він інформує користувачів про заходи, концерти або програми.

3) Персональний бот - це свого роду аналогія сайту-візитки, де приватний фахівець розповідає про себе, свої пропозиції, показує портфоліо і залишає контакти для зв'язку. Подібний чат-бот буде цікавий індивідуальним підприємцям, консультантам і тренерам, фрілансерам і іншим фахівцям, які потрапляють в категорію «самозайняті».

4) Інформер - завдання такого бота відповісти на вузько тематичні питання користувача. Може стати непоганою заміною пошуковій системі. Застосовується в сферах медицини, освіти і в будь-якому іншому соціальному напрямку.

5) Цільовий бот - призначений для взаємодії з певною групою людей, які мають намір взяти участь у заході або у них є спільні інтереси. Цей різновид чат-бота підходить для симпозіумів, конференцій і т.д. З його допомогою організатори отримують зворотний зв'язок від учасників, нагадують їм про дату проведення, сповіщають про зміни в розкладі або програмі виступу, організують спілкування між ними.

6) Функціональний бот - може виконувати будь-які функції, пов'язані з бізнесом. Підходить майже для всіх сфер діяльності. За допомогою даного інструменту можна продавати товари, пропонувати послуги, навчати чого-небудь, рекламувати і багато іншого.

#### **1.4. Визначення необхідності вимог до проекту**

Постановка вимог до програмного забезпечення (ПЗ) є одним з перших та найголовніших кроків. Маючи певні завдання, які повинна виконувати система, програміст розуміє, яким чином потрібно організувати роботу над програмою. Вимоги допомагають продумати функціональні та нефункціональні можливості ПЗ. При проектуванні будь-якого проекту потрібно виділити сферу діяльності, в якій буде працювати програма, та основні чинники, а саме

роботоспроможність та відмовостійкість ПЗ, щоб уникнути моментів «падіння» та припинення функціонування сервісу.

Визначення головних задач, які повинні виконуватись системою, мають передувати початку проектування чат-бота, тому що неясність у вимогах може призвести до проблематики в реалізації сервісу.

Необхідні кроки, які потрібно зробити для постановки вимог [8]:

- зрозуміти структуру та форму системи, спрямувавши увагу на її функціях та розгляд того, як вони задовільняють та виконують запити користувачів;
- оцінити вичерпність, обґрунтованість системи та відповідність для середовища, в якому програмне забезпечення буде використовуватись;
- використовувати дані про модель, що створюється, та інформацію з пунктів вище для визначення можливості побудови сервісу та управління його масштабом перед тим, як робити значні інвестиції до проекту.

На даному етапі можна перейти до наступного рівня конкретизації та створити більш повну, глибоку модель вимог до системи, над розробкою якої працюємо. Процес уточнення може включати в себе використання формальних засобів забезпечення якості, перегляди, ревізії та моделювання. В системах, де наслідки збоїв не є катастрофічними, рівень трудомісткості більш помірний. У цих випадках просто необхідно сформулювати визначення системи досить чітко, щоб його могли зрозуміти всі учасники процесу, а також для того, щоб забезпечити ефективні умови розробки та «досить високу» ймовірність успіху. Керуючись прагматичними і економічними міркуваннями, слід створити достатню кількість специфікацій вимог, щоб розроблена програма була саме тим, чого хоче користувач.

Отже, існує велика кількість «різновидів» вимог, але для сервісу, який буде реалізований в даній роботі, буде висвітлено два типи:

- функціональні вимоги;
- нефункціональні вимоги.

При виставленні вимог, варто звернутись до класифікації вимог до програмних систем (FURPS)[9], яку запропонував Роберт Грейді (Robert Grady) в 1992 році [10]. Класифікація складається з п'яти груп вимог:

- функціональність (functionality) – включає в себе властивості та можливості системи;
- практичність (usability) – досвід користувача, який складається з таких пунктів: людський фактор, послідовність використання системи;
- надійність (reliability) – надійність у використанні: частота збоїв та «падіння», відмовостійкість та відновлюваність сервісу;
- продуктивність (performance) – масштабування, потужність, ефективність використання та споживання ресурсів;
- можливість підтримки (supportability) – вимоги до підтримки: гнучкість, підтримка, модульність, розширюваність.

#### **1.4.1. Функціональні вимоги чат-боту**

Функціональні вимоги (Functional Requirements) описують як саме веде себе сервіс в експлуатаційний період. Ці потреби зазвичай орієнтовані на дії, які виконує користувач, та як система на це реагує.

Для чат бота функціональні вимоги потрібно виділити для клієнта та для менеджера. Кожен з них має свій варіант взаємодії з сервісом і тому потрібно описати потреби в залежності від ролі користувача системи.

Клієнт повинен мати змогу:

- a) переглянути акційні пропозиції магазину;
- b) отримувати дані про товар;
- c) оформити замовлення:
  - 1) оплативши через платіжну систему.
  - 2) зв'язавшись з менеджером, щоб замовити товар з можливістю сплати за товар при отриманні.
- d) підписатись/відписатись від акційної розсилки;
- e) зробити повернення товару, якщо він не був в експлуатації;
- f) зв'язатися з менеджером.

Менеджер повинен мати змогу:

- a) вносити зміни в товарну позицію та його опис;
- b) отримувати повідомлення про прохання клієнта зв'язатись з менеджером компанії.

#### **1.4.2. Нефункціональні вимоги чат-боту**

Нефункціональні вимоги (Non-functional Requirements) – це вимоги, які, як правило, використовують для опису деяких «атрибутів системи» або «атрибути системного середовища». До них відносять:

- атрибути якості;
- обмеження в проектуванні;
- реалізація, зовнішні або прикладні програмні інтерфейси (ППІ).

Якщо звернутись до класифікації вимог до програмних систем (FURPS), то до нефункціональних вимог належать:

- a) практичність (usability);
- b) надійність (reliability);
- c) продуктивність (performance);
- d) можливість підтримки (supportability).

Вимога практичності спрямована на створення простого у використанні інтерфейсу програмного забезпечення. Необхідно описати, наскільки просто новий користувач може оволодіти сервісом і чи не принесе це труднощів, тому важливо додати побільше текстових повідомлень з описом елементів або кроків, які необхідно зробити покупцеві, щоб досягти поставленої цілі.

Також варто порівняти практичність використання створеної системи з працюючими схожими сервісами, які вже відомі та здобули у користувачів певного успіху. Це порівняння здійснюється через досвід користувача (UX).

Потрібно віднайти вже існуючі системи інтерактивних підказок, програм-помічників, засобів попередження, посібників для користувача та інших форм документації можливостей програмного забезпечення для спрощення орієнтації по сервісу для клієнта.



Також важливим є використання угод та стандартів, які спрямовані на опис розробки людино-машинного інтерфейсу. Це необхідно, щоб сервіс працював не гірше аналогів та мав індивідуальні якості, які будуть надавати йому вираженого повноцінного вигляду та виділятимуть серед інших створених систем.

Вимога до надійності описує можливість помилок чи збоїв в роботі сервісу. Втрата даних чи «падіння» системи на довгий час є неприйнятним, тому потреба в описі надійності є важливою при створенні програмного забезпечення.

У роботі будь-якого сервісу можуть бути помилки та збої, тому більшість користувачів вже готові до цього, але при розробці сервісу потрібно звести можливість таких ситуацій до мінімуму.

Отже, можна розглянути такі аспекти надійності:

- доступність (availability) – вимога, яка характеризується можливістю користування сервісом в період вказаного часу. Система повинна бути працездатною та готовою до використання клієнтом. У випадку роботи чат-бота - це можливість бути доступним користувачам в режимі «nonstop», тобто цілодобово;

- час між відмовою (Mean Time Between Failures, MTBF) – це величина, яка вказується в годинах, в деяких випадках у днях або місяцях. Проте важливо конкретизувати, що саме означає «відмова сервісу» для даного проекту. Для чат-бота збоєм є завершення роботи або «падіння» системи. Боти розташовані на хостингах або віддалених серверах, тому при збоях вже має бути підготовлений скрипт для відновлення роботи. Завдяки цьому час відмови зводиться до мінімуму, можливо навіть до пари хвилин;

- точність (accuracy) – цей аспект вимог визначається більш конкретно в залежності від програмного продукту, тому що він визначає точність в поданні даних, наприклад: показання часу, вартість валюти та швидкість зміни цих даних на актуальні. Проектованою системою є чат-бот для

продажів товарів, тому вказується вартість товарів з точністю до гривні та термін доставки товару до одного дня;

- час відновлення (Mean time to repair, MTTR) – вимога, яка характеризує відрізок часу, коли сервіс може бути недоступним для використання. Проте він є нормою для клієнтів, тому що користувач попереджений про проблему та про її усунення не більше ніж за 5 хвилин. Якщо є глобальна причина відмови системи, «підняття» сервісу займе до 60 хвилин. Маючи змогу автоматизувати відновлення програмного забезпечення віддалено, технічний відділ відновлює працездатність чат-бота за декілька хвилин;

- максимальний коефіцієнт помилок в коді – кількість помилок на певну кількість рядків коду чи на одну функцію. Чат-бот – це сервіс, код якого буде мати не більше однієї тисячі рядків, тому цей коефіцієнт максимально приближений до 0;

- кількість різного виду помилок – помилки розділяються на незначні, важливі та критичні. Ця вимога конкретизує поняття важливості цих помилок. Для чат-бота незначною помилкою є збій в розкладі розсилки клієнтам, що ніяк не впливає на сервіс в цілому. Важливою помилкою є «падіння» та часткова відмова працездатності віртуального менеджера. Критичною ж є помилка, коли бот втрачає дані користувачів через крадіжку чи втрату останньої записаної інформації, тому важливо регулярно робити резервні копії бази даних (БД). Це зводить ймовірність складних помилок до мінімуму.

До вимоги продуктивності зазвичай відносять:

- час відповіді для транзакції: середній або максимальний. Чат-бот працюватиме через прикладний програмний інтерфейс (application programming interface, API), тому його можливість відповіді залежить від відгуку серверів месенджерів. Але якщо кількість запитів не буде перевищувати допустимі норми, бот може гарантувати максимальну швидкість відповіді;

– пропускна спроможність – число транзакцій в секунду. Ця характеристика залежить від дозволеної пропускної спроможності серверу месенджера. У Telegram відправка йде до одного користувача з лімітом одне повідомлення в секунду;

– ємність – скількох користувачів чи транзакцій може обслуговувати сервіс. У Telegram кількість транзакцій на секунду може досягати 30 повідомлень, які можна відправити різним користувачам;

– режим зниження продуктивності – це можливість підтримувати працездатність сервісу на обмежених ресурсах, тобто коли система працює паралельно з іншими програмами та вимушена використовувати меншу кількість ресурсів, ніж їй необхідно. У таких випадках є економ-режими, за рахунок яких підтримується працездатність, але зі зменшеними можливостями. Чат-бот працює на віддаленому сервері та функціонує з месенджером, тому заважати його роботі може лише неспроможність апарату користувача використовувати мобільний додаток Telegram, Facebook Messenger, Slack або WhatsApp.

Можливість обслуговування системи заключається в можливості внесення змін та модифікації системи для оновлення можливостей та виправлення несправностей в роботі. Для підтримки сервісу необхідно обрати інструменти, які найкраще підходять до середовища програмного забезпечення. Цей пункт є важливим для подальшого розвитку та підтримки системи. Можливість обслуговування потрібно розглянути на ранніх етапах проектування та створення сервісу.

### **Висновки до розділу**

В даному розділі представлена інформаційна система, яку називають чат-ботом або віртуальним помічником, що допомагає змодельовати реальну розмову з клієнтом.

Розглянуті мінуси та плюси інтегрування чат-ботів в систему продажів. Особливо можна виділити те, що боти можуть працювати цілодобово та замінити людину у виконанні великої кількості рутинної роботи.

Проаналізовано сфери використання віртуальної підтримки. Особливо дана технологія корисна компаніям, яким важливо регулярно інформувати про важливі заходи, акції або ж про технічні неполадки в роботі підприємства своїх клієнтів.

Розглянуто основні принципи взаємодії чат-ботів з користувачами, елементи, за допомогою яких створюють віртуальні сервіси, та їхні варіанти реалізації. Представлено класифікацію чат-ботів за сферою їх використання.

Також варто відмітити, що технічна підтримка є важливою складовою будь-якого бізнесу і направлена на допомогу клієнтам якомога швидше досягти своєї кінцевої мети: отримання додаткової інформації чи допомога з покупкою товарів або послуг. Розглянуті функціональні та нефункціональні вимоги даної системи, а також представлені класифікації чат-ботів для розглядання їх архітектурних особливостей.

Чат-бот повинен відповідати основним питанням впровадження його в бізнес-процеси компанії. Інтегрування в компанію даної системи дасть приріст в швидкості та якості обробки онлайн-замовлень через мережу Інтернет. Саме тому дана технологія принесе бізнесу неабияку користь та економить кошти на подальший розвиток. Також вагомим плюсом у використанні чат-ботів є те, що вони ефективно взаємодіють з клієнтами на будь-яких масштабах.

## РОЗДІЛ 2 РОЗГЛЯД ІНСТРУМЕНТІВ ТА АРХІТЕКТУРИ ДЛЯ СТВОРЕННЯ ЧАТ-БОТУ

### 2.1. Вибір платформи для чат-боту

Платформа для чат-боту - це сервіс, за допомогою якого можливо створити віртуального менеджера, використовуючи прикладний програмний інтерфейс (application programming interface, API). Більшість популярних месенджерів підтримують використання ботів, а саме: Facebook Messenger, Viber, Telegram, Slack та інші. Кожна з платформ має свої можливості для написання бота. Важливо перед створенням чат-боту проаналізувати плюси та мінуси кожної (табл. 2.1).

Розглянемо можливості та функціонал всіх описах платформ - це допоможе обрати найгнучкішу з них для створення на її основі чат-боту. Slack є більш корпоративним менеджером, який використовують для об'єднання працівників компанії в одну мережу для зручного спілкування, тому ця платформа обмежена в можливостях чат-ботів [8].

Таблиця 2.1

Порівняння платформ за функціональними можливостями для чат-ботів

Платформа	Telegram	Viber	Facebook Messenger
<b>Взаємодія з користувачем</b>	Кнопки та команди, голосові та текстові повідомлення, коментарі та пошук інформації	Кнопки та текстові повідомлення	Кнопки та текстові повідомлення

Кафедра ПІ				НАУ 21 16 78 – 000 ПЗ			
<i>Виконав</i>	<i>Масний М.В.</i>			Розгляд інструментів та архітектури для створення чат-боту	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Толстікова О.В.</i>					27	15
<i>Консульт.</i>					122 ТП-416Б		
<i>Нормоконт</i>	<i>Боровик В.М.</i>						
<i>Зав. каф.</i>	<i>Гамаюн В.П.</i>						

<b>Платформа</b>	<b>Telegram</b>	<b>Viber</b>	<b>Facebook Messenger</b>
<b>Інтегрування з іншими сервісами</b>	+	-	-
<b>Inline режим (використання в інших чатах)</b>	+	-	-
<b>Інтерактивна взаємодія</b>	Гра з користувачем в одну зі створених ігор, продаж товару	Створення постів у вигляді каруселі з товарами	Структуровані елементи (до 10 в одному повідомленні)
<b>Панель адміністратора</b>	Присутнє і має досить широкий функціонал	Мінімальне (тільки отримання даних з боту)	Мінімальне (при необхідності має змогу переключити діалог на реальну людину)
<b>Оплата за товар</b>	+	+	+

Проаналізувавши можливості та функції кожної з платформ, можна визначити, що месенджер Telegram є найоптимальнішим для написання чат-боту за підтримки зв'язку з клієнтом у майбутньому.

Важливим фактором також є кількість користувачів платформи. За даними з сервісу Statista [11] (див. рис. 2.1) з вагогим відривом має першість Facebook Messenger серед тих, хто має можливість достатньої підтримки чат-ботів, але якщо врахувати стрімкість розвитку, Telegram є дуже перспективною платформою за кількістю охопленої аудиторії.

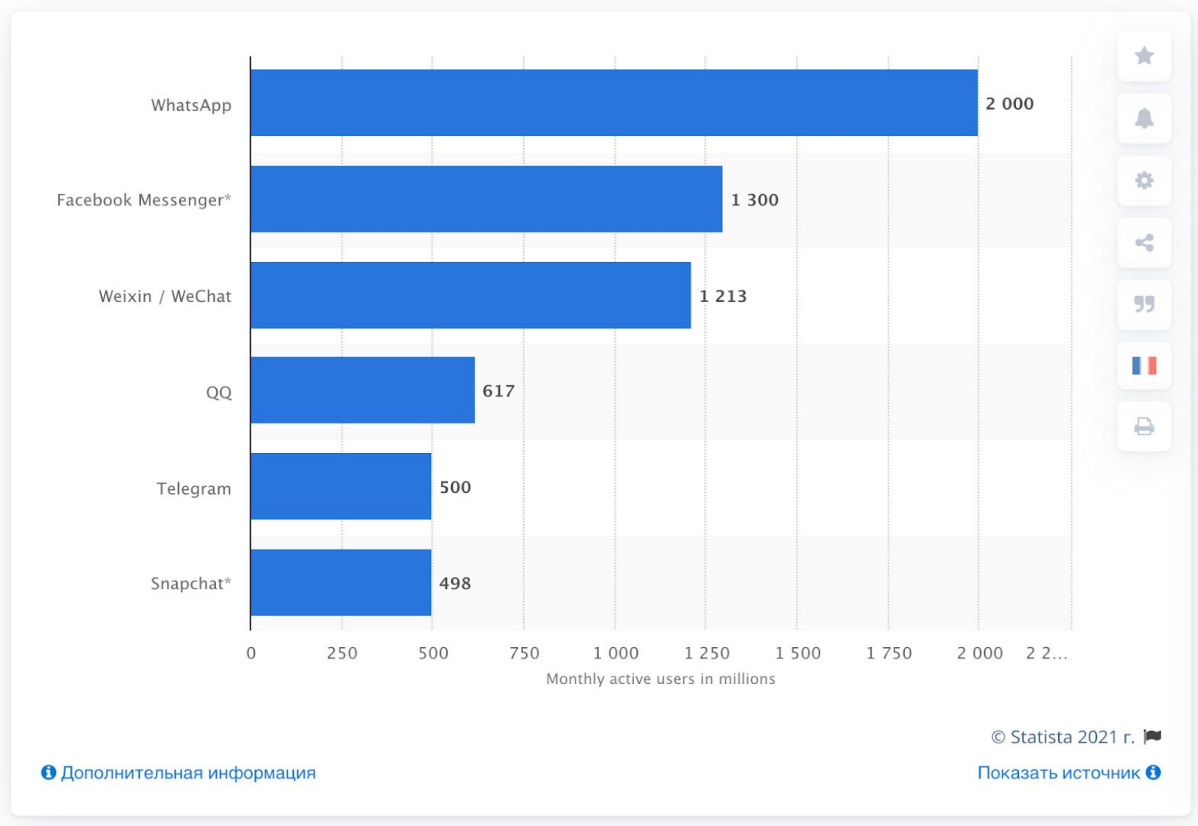


Рис. 2.1. Популярність месенджерів у світі за даними платформи Statista

## 2.2. Інструменти для створення чат-ботів

Створення чат-ботів не є проблемою для будь-якого бізнесу, так як прогрес надав велику кількість сервісів, веб-додатків та фреймворків для створення віртуального помічника. Однак для розуміння роботи бота, потрібно погрузитись в розробку та створення на більш детальному рівні, тобто використовуючи мову програмування.

Створити віртуального помічника можна на будь-якій мові програмування, за допомогою якої можна працювати з WEB API. Першими в списку інструментів для розробки є Node.js та PHP, які є основними мовами для роботи з прикладним програмним інтерфейсом. Також є багато різноманітних бібліотек для створення чат-ботів на таких мовах, як Java та Python.

Важливо також обрати сучасну мову програмування, наприклад JavaScript або Python [12], які мають достатній потенціал для розвитку та велику кількість програмістів. Це допоможе підтримувати бота більший відрізок часу з використанням сучасних технологій.

За допомогою цих інструментів створюється так званий бекенд, тобто серверна частина. Саме комп'ютер в реальному часі очікує запит на опрацювання даних від веб-інтерфейсу (Front-end, фронтенд) та в залежності від вхідних даних надає відповідь клієнту. Найчастіше фронтенд виступає месенджером або сайтом, які зв'язані з сервером чат-бота. Важливо відмітити, що бекенд має змогу працювати не з одним веб-інтерфейсом, а одночасно може опрацьовувати як з месенджером, так і з чатом підтримки на сайті компанії.

Варто розглянути сервіси для створення чат-ботів. Використання таких веб-додатків зводить до мінімуму витрати часу на створення бота. Створення віртуальних помічників за допомогою сервісів є менш гнучким, чим написання боту за допомогою мов програмування. Однак веб-додатки мають достатній функціонал для створення віртуальних помічників для бізнесу або будь-якої компанії.

### **2.3. Вибір інструментів для створення чат-ботів**

Створити чат-бота в месенджері Telegram можна двома способами:

- написати на мові програмування - розробка чат-бота буде виконуватись за допомогою Python, прикладом може слугувати віртуальна технічна підтримка застосунку "Дія: Державні послуги онлайн" [13];
- сконструювати за допомогою чат-бот сервісів, наприклад віртуальна помічниця сучасного медичного центру краси та здоров'я CPG [14];

#### **2.3.1 Аналіз бібліотек для створення чат-ботів**

Найрозповсюдженішими для Python бібліотеки є такими:

- aiogram;
- python-telegram-bot;
- telethon.

Кожна з цих бібліотек має свої плюси та недоліки. Щоб обрати необхідну, потрібно розглянути їх можливості та функціонал. Це допоможе звернутись саме до того інструменту, який буде виконувати необхідну нам роботу.



Aioogram - є асинхронною бібліотекою для Telegram Bot API [15], яка є достатньо простою та зручною у використанні при розробці ботів. Фреймворк має достатньо вагомні плюси, які виділяють його на фоні альтернативних інструментів, а саме:

- асинхронність - за допомогою використання `asyncio` та `aiohttp` надає значної продуктивності в роботі клієнт-серверних додатків;
- надає можливість писати гнучкий та “читаємий” код програми;
- швидкість роботи - архітектура фреймворка надає можливість створювати швидкі та продуктивні додатки.
- має скінченний автомат (FSM) [16] - може розв’язувати велику кількість задач.

Python-telegram-bot - бібліотека для “простих” чат-ботів. Достатньо проста у вивченні та добре підійде створення не високопродуктивних сервісів. Має достатній функціонал та можливості для розробки чат-ботів загальної необхідності, що матимуть невелику навантаженість та малу кількість користувачів. Також має можливість працювати асинхронно, але архітектура фреймворка не дає достатньої продуктивності в роботі на відміну від аналогів.

Telethon - також асинхронна бібліотека для створення чат-ботів, є конкурентом по продуктивності для aioogram. Цей фреймворк більше направлений на розширення можливостей користувача за рахунок мобільного протоколу MTProto [17], що є альтернативою bot Api. Важливо відмітити, що використання MTProto застосовується без будь-яких залежностей, що спрощує роботу з бібліотекою. У даного фреймворка є ще декілька плюсів, що надають можливість йому конкурувати з іншими бібліотеками для створення чат-ботів. Наприклад:

- сумісність з кодом написаним на різних версіях Python. Це є важливим фактором, якщо проект написаний на версіях 3.4 та нижче. Причиною є те, що з приходом версії 3.5 у Python почали впроваджувати асинхронність, а це є правильний крок для розвитку, тому Telethon також розвивається у цьому напрямку. Проте можливо використовувати більш давні версії фреймворку для коду з використанням потоків;

- зручність у використанні в ситуаціях для швидких скриптів, яким не потрібне оновлення. Варто не використовувати асинхронність в цих кусках коду просто, а працювати з послідовним кодом. Проте це не шкодить продуктивності всього додатку загалом.

- швидкість роботи - за допомогою асинхронності можна пришвидшити роботу сервісу.

- навчання - використання бібліотеки несе в собі вміння користуватись асинхронним програмуванням. Документація фреймворка всіяко допомагає розробникам розвиватись та збільшувати свої знання в мікрооптимізації сервісів.

З розгляду трьох бібліотек aiogram, python-telegram-bot, telethon результатом є те, що python-telegram-bot - є дуже простою для створення високонавантаженої системи, тому для написання якісного сервісу краще використовувати telethon або aiogram, які підтримують асинхронність.

Найоптимальнішим варіантом є використання бібліотеки aiogram. Оскільки даний фреймворк може розв'язувати велику кількість задач, асинхронний та швидко працює. За допомогою цієї бібліотеки можна писати "читаємий" код, що є важливим фактором в тестуванні та при виправленні помилок системи, та вона є не складною в освоєнні.

### 2.3.2. Аналіз чат-бот сервісів

Створення чат-ботів можливе не лише за допомогою написання коду, а й за допомогою сервісів для конструювання ботів. Ці сервіси надають необхідні можливості та функціонал розробки віртуального помічника.

Написання чат-бота на мові програмування може зайняти деякий час для освоєння інструментів та всіх можливостей API месенджера, в який є намір інтегрувати віртуального менеджера. Окрім використання фреймворку для створення боту, потрібно задіяти велику кількість інструментів для збереження інформації або ж опрацювання web-запитів. Також важливим фактором є підтримання функціонування системи в реальному часі, тому потрібно бекенд бота розмістити на віртуальному сервері.

Сервіси для розробки віртуального помічника та його архітектури беруть на себе всі питання по функціонуванні чат-бота та зберіганні всієї важливої особистої інформації клієнтів і їх взаємодію з системою. Що значно спрощує реалізацію задуманого сервісу тех-підтримки.

Чат-бот сервіси різняться своїми можливостями та функціоналом, але головним є принцип їх роботи і за цим аспектом можна поділити сервіси на три групи:

- перші створюють карту повідомлень для користувача, тобто ведуть його по завчасно спланованому шляху;
- другі використовують штучний інтелект для створення відчуття, що клієнт веде діалог з реальною людиною;
- треті комбінують в собі першу і другу групи, що допомагає використати їх переваги та максимально ефективно спілкуватись з покупцями.

Найрозповсюдженішими є сервіси, що розробляють ботів на основі карти повідомлень та команд. Таке рішення є простим у реалізації, тому що не потребує багато знань та ресурсів для розробки, а тому це і найдешевший варіант для створення чат-боту.

Карти повідомлень будуються за принципом блок-схеми - створюються послідовні елементи, наповнюються необхідною інформацією та налаштовується зв'язок між ними для формування логічного діалогу. Для ефективної роботи важливо, щоб сервіс мав візуальний редактор, в якому є можливість бачити всі елементи та зв'язки між ними. Це допомагає швидше знаходити помилки в роботі та знаходити "слабкі" місця бота. Прикладом може слугувати чат-бот, який допомагає підтримувати зв'язок з учасниками онлайн-конференції SalesHero Online: він видає учасникам програму конференції і нагадує про початок виступів [18]

Друга та третя група сервісів є більш трудомісткою та довготривалою. Для використання штучного інтелекту в проекті, його потрібно навчити діяти в реальних умовах, для яких ШІ і був спроектований. Наприклад, Watson Assistant створений компанією IBM [19]. Чат-бот надає клієнтам швидкі, послідовні і точні відповіді з будь-якого додатку, пристрою або каналу.

Використовуючи штучний інтелект, Watson Assistant навчається спілкуванню з клієнтами, покращуючи свою здатність вирішувати проблеми з першого разу. Що позбавляє користувачів розчарування, пов'язаного з довгим очікуванням, втомлених пошуків і марними чат-ботами.

Необхідно проаналізувати існуючі варіанти чат-бот сервісів для отримання максимально швидкого і недорогого рішення. Таким чином розглядати потрібно сервіси з першої групи (табл. 2.2):

Таблиця 2.2

Аналіз чат-бот сервісів

Назва	SendPulse	Flow XO	ManyChat	BotMother
<b>Вартість</b>	Базові функції безкоштовні, від 10\$ за 1000 підписників.	Базовий план - безкоштовний, про-пакет від 19\$.	Базовий план - безкоштовний, про-пакет від 10\$.	Один бот з базовими інструментами - безкоштовно, про-пакет від 19\$.
<b>Підтримуванні платформи</b>	Facebook, WhatsApp, Telegram, Вконтакте.	Facebook, Telegram, Slack, Twilio SMS.	Facebook, Instagram.	Telegram, Facebook, Viber, Вконтакте, Однокласники та WhatsApp.
<b>Інструменти повідомлень</b>	Базові + можливість приймати оплату в чат-боті.	Базові + можливість приймати оплату в чат-боті.	Базові + можливість приймати оплату та робити дзвінки з чату.	Базові + можливість приймати оплату в чат-боті.

Продовження таблиці 2.2

<b>Назва</b>	<b>SendPulse</b>	<b>Flow XO</b>	<b>ManyChat</b>	<b>BotMother</b>
<b>Мова інтерфейсу</b>	Англійська, українська, російська, іспанська.	Англійська.	Англійська.	Російська.
<b>Інтеграція</b>	Месенджери, сайт, Google Analytics.	Месенджери, сайт, Google Analytics.	Facebook, Instagram, Google Sheets, Shopify.	Месенджери, сайт, Google Analytics і Google Sheets.
<b>Дані від користувача</b>	Зберігає номер телефону та email, можливий безкоштовний експорт.	Зберігає всі дані, що надає клієнт.	Робить запити на отримання даних користувача, зберігає їх та сегментує клієнтів.	Може робити запити на отримання ПІБ клієнта, номер та його адресу, геолокацію.
<b>Шаблони для створення ботів</b>	Тільки в pro-пакеті.	Є шаблони для сервісів бронювання квіток, столиків в ресторани, реєстрації на захід та FAQ.	Шаблони для ресторанів, агентства по нерухомості, інтернет-магазин, сфери послуг та магазин кави.	Відсутні.

Продовження таблиці 2.2

Назва	SendPulse	Flow XO	ManyChat	BotMother
<b>Додаткові платні послуги</b>	Кількість підписників на бота, шаблони, необмежена кількість ботів, брендування компанії.	Можливість експорту даних клієнтів, видалення логотипу сервісу в чаті, пріоритетна підтримка.	Більше шаблонів, необмежена кількість автоматичних розсилок, оплата в боті, А/В тестування, інтеграція з CRM.	Можливість додавати галерею до повідомлення, функціонал для розсилок, оплати, FAQ та запис даних до Google Sheets.

Для розробки чат-ботів для месенджера Telegram можливо використовувати три з чотирьох варіантів, а саме: SendPulse, ManyChat та BotMother. Проте важливим фактором є безкоштовне експортування даних клієнтів, тому обрати варто сервіс для створення чат-ботів SendPulse.

#### 2.4. Проектування інтерфейсу чат-бота

На даному етапі важливо продумати якісну взаємодію з клієнтом. Це є одним із ключових факторів успіху сервісу. Необхідно визначити конкретний функціонал, тематику та аудиторію. Інтегрування чат-ботів в систему продажів передбачає, що віртуальний помічник буде використовуватись в інтернет-магазинах або компаніях, які продають свої продукти [20]. Саме тому варто зосередитись на можливості бота допомогти клієнтові замовити товар або ж зацікавити новинками чи акційними пропозиціями магазину.

Важливо надати віртуальному помічникові «індивідуальності» компанії. Для цього потрібно зробити так, щоб клієнт міг ідентифікувати за чат-ботом магазин, з яким він хоче співпрацювати. Це можна зробити за допомогою аватарки, додавання опису та відповідності бота до самої компанії.

При використанні віртуального помічника клієнт має почувати себе комфортно та розуміти, що саме може надати йому чат-бот. Для цього потрібно створити справку, де користувач зможе знайти всі необхідні йому команди (рис. 2.2). Перше повідомлення має приходити відразу після команди /start та інформувати людину як «спілкуватись» з помічником.

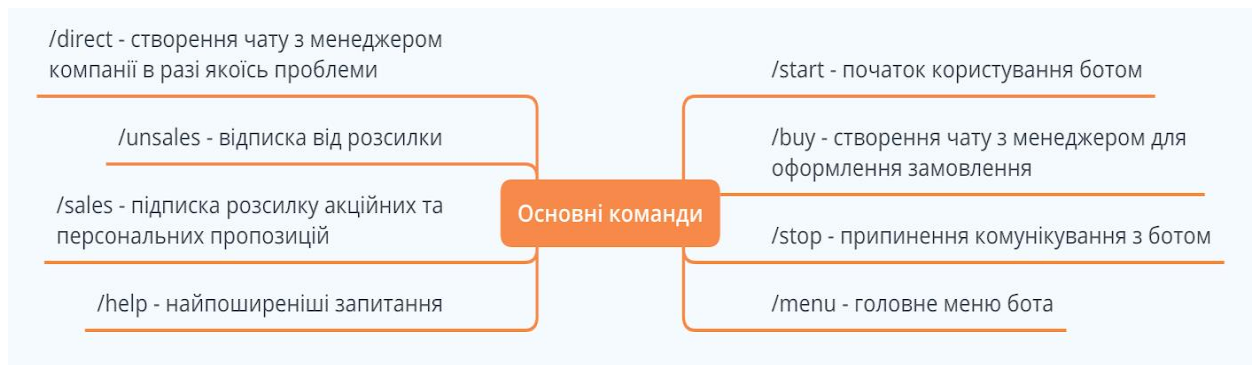


Рис. 2.2. Основні команди чат-боту

Наступним етапом в проектуванні інтерфейсу є створення логіки спілкування, тобто карти повідомлень. Розпочати слід з головних елементів управління чат-бота (рис. 2.3).

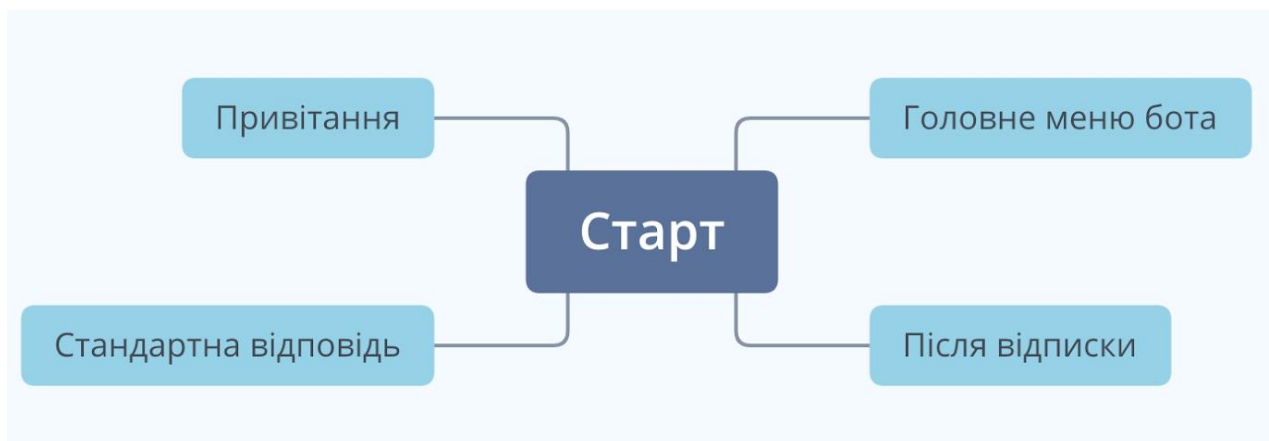


Рис. 2.3. Головні елементи управління чат-бота

Далі потрібно описати кожний елемент, тобто створити ланцюг діалогу користувача:

а) Привітання - це діалог, який активізується відразу після натискання клієнтом клавіші start або введення команди /start. Ця віртуальна бесіда

знайомить користувача з ботом та навпаки. Після ознайомлювальної інформації, віртуальний помічник запрошує у клієнта ім'я, номер телефону та електронну пошту. Проте якщо користувач не хоче надавати ці дані, він потрапляє до головного меню бота (рис. 2.4);



Рис. 2.4. Діалог «привітання»

b) Головне меню - це основна відправна точка для всіх користувачів. Покупці «йдучи» по карті повідомлень будуть повертатись сюди ну або ж відписуватись. Цей діалог має три вітки (див. рис. 2.5):

c) Зв'язок з менеджером з привожу замовлення - покупець йде цим «шляхом» для того, щоб зробити замовлення;

d) Показати акційні пропозиції - цей діалог допомагає сегментувати покупців, які готові купувати зі знижкою або ж зацікавились новинками в інтернет-магазині. Після обрання цього пункту клієнту кожного дня з 9 по 18 годину будуть приходити повідомлення з товарами, на яких акційна ціна або ж вони тільки надійшли до магазину;



е) Поширені запитання FAQ - це меню відповідей на питання, що люди задають найчастіше. Тут є інформація про магазин, про способи доставки товару, способи оплати замовлення та який товар підлягає поверненню;

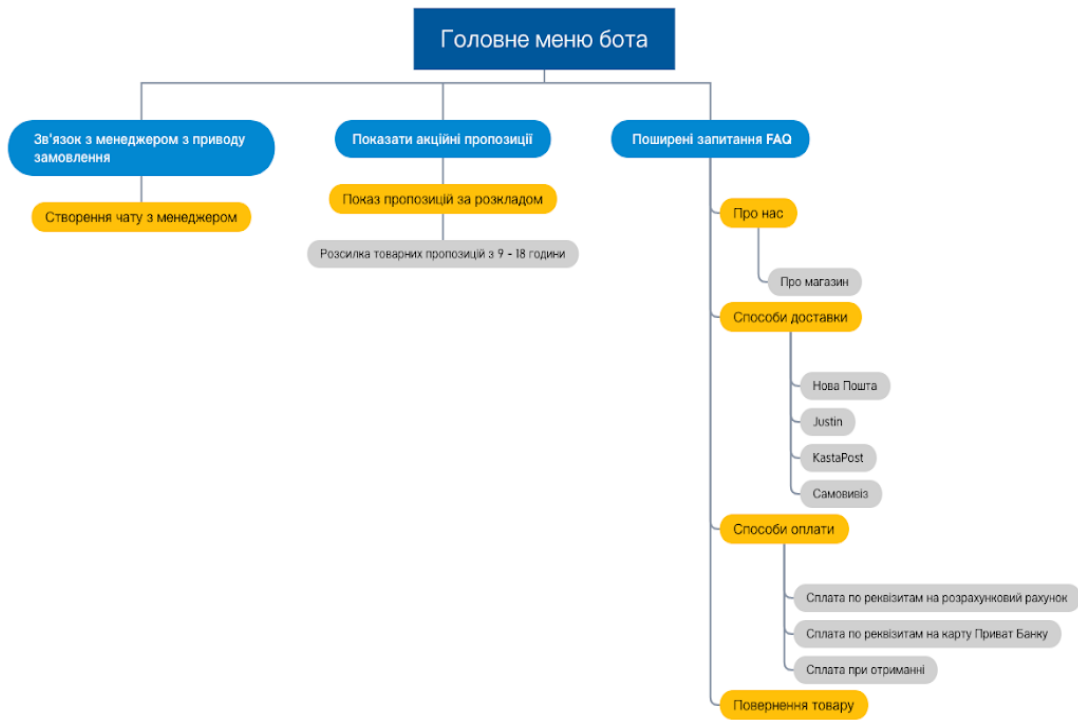


Рис. 2.5. Головне меню бота

ф) Після відписки - цей діалог спрацьовує після того, як користувач відпишеться від боту за допомогою команди /stop. Після чого йому віртуальний помічник пропонує залишити відгук про користування, щоб в майбутньому аналізувати зворотній зв'язок для покращення роботи (див. рис. 2.6);

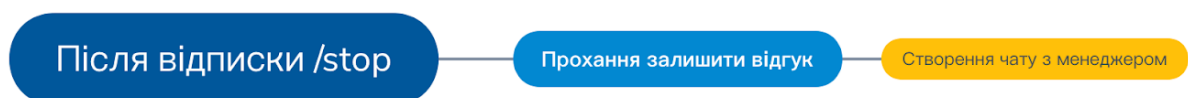


Рис. 2.6 Діалог «Після відписки»

g) Стандартна відповідь - це відповідь клієнту на невідому для бота команду або ж повідомлення. Чат-бот приносить вибачення за незручності та надає вибір діалогів з головного меню: Зв'язок з менеджером з приводу замовлення, показати акційні пропозиції та поширені запитання FAQ (див. рис. 2.7).

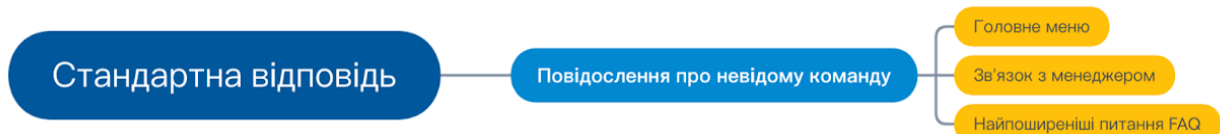


Рис. 2.7. Діалог «Стандартна відповідь»

Проектуючи шлях повідомлень потрібно уникати глухих ситуацій, коли людина не знає, що їй необхідно зробити. Саме тому варто використовувати такі інструменти для створення повідомлень:

- текст - це основний елемент повідомлення, який несе найголовнішу інформацію;
- картинка або gif-анімацію - за допомогою зображення можна направляти або сповіщати користувачів;
- кнопки - незамінний елемент для отримання зворотного зв'язку від клієнта;
- швидкі відповіді - допомагають збирати інформацію про людину;
- файли - за допомогою них можна передавати користувачеві необхідні дані або ж дельну інформацію про товар;
- імітацію вводу тексту - створює видимість, що текст вводиться кимось, а не по шаблону відправляється;
- карточки товару - цей елемент несе в собі повну інформацію про товар або послугу та має можливість провести оплату в месенджері. Це допомагає пришвидшити заключення угоди та проведення перемовин.

## **Висновки до розділу**

Одним з головних елементів для написання чат-боту є визначення сервісу, з яким буде інтегровано віртуального помічника. Проаналізувавши можливості та функції популярних платформ, можна визначити, що месенджер Telegram є найоптимальнішим для написання чат-боту за підтримки зв'язку з клієнтами у майбутньому.

У розділі розглянуто варіанти реалізації віртуального помічника, якого можна створити двома способами: написавши сервіс однією з популярних мов програмування або ж створивши його через сервіс для проектування чат-ботів.

Розробляється віртуальний помічник буде на мові програмування Python за допомогою асинхронного фреймворку aiogram, який взаємодіє з прикладним програмним інтерфейсом платформи Telegram. Також для демонстрації та проведення експерименту буде спроектовано чат-бота за допомогою сервісу SendPulse. Що покаже, яким чином можна створити віртуальну технічну підтримку та інтегрувати бота в систему продажів.

Згідно аналізу вимог до чат-боту та його функціонування спроектовано необхідний інтерфейс для сервісу, який буде відповідати поставленому завданню.

## РОЗДІЛ 3

# РЕАЛІЗАЦІЯ ЧАТ-БОТУ ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ ТА СЕРВІСУ ДЛЯ ЇХ СТВОРЕННЯ

### 3.1. Підготовка інструментів для створення чат-боту

Розробка чат-боту розпочинається з підготовки інструментів, які будуть використовуватись у створенні віртуального помічника. Потрібно зробити дерево проекту для зручної роботи з файлами та імпортування необхідних структурних елементів, а також встановити необхідні для роботи бібліотеки.

Першим кроком буде створення репозиторію на веб-сервісі для контролю версій github [21] (рис. 3.1), що допоможе в майбутньому не втратити написаний код через технічні проблеми та дасть змогу додавати нові можливості чат-боту.

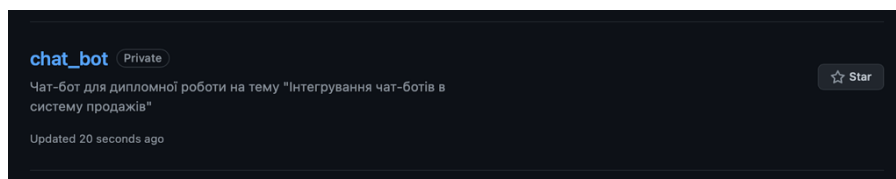


Рис. 3.1. Репозиторій проекту на github.com

Далі потрібно «клонувати» створену кореневу папку на локальний комп'ютер за допомогою терміналу або командного рядка (рис. 3.2), в залежності від операційної системи Unix або Windows відповідно.

```
→ diploma git clone https://github.com/mmasniy/chat_bot.git
Cloning into 'chat_bot'...
warning: You appear to have cloned an empty repository.
→ diploma _
```

Рис. 3.2. Клонування проекту

Кафедра ПІ				НАУ 21 16 78 – 000 ПЗ			
Виконав	Масний М.В.			Реалізація чат-боту за допомогою мови програмування та сервісу для їх створення	Лім.	Арк.	Аркушів
Керівник	Толстікова О.В.					42	15
Консульт.					122 ТП-416Б		
Нормоконт	Боровик В.М.						
Зав. каф.	Гамаюн В.П.						

Отримавши порожню папку, ми створюємо необхідні директорії, де будуть окремо зберігатись структурні елементи. Це допоможе ефективно писати код та спростить навігацію по проекту. Для зручної роботи потрібно створити папки та файли (рис. 3.3):

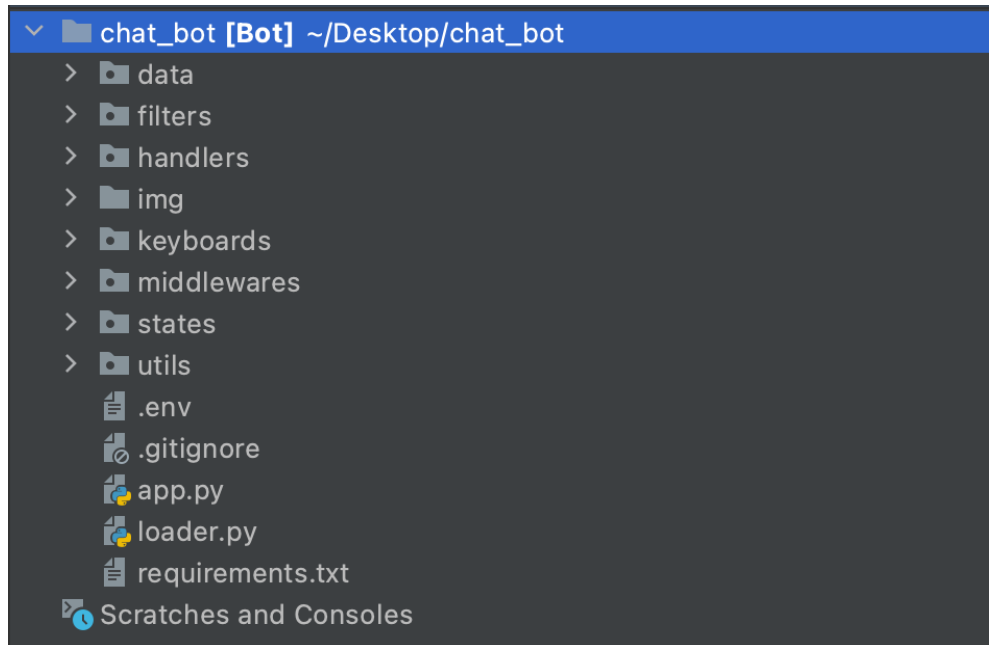


Рис. 3.3. Структура проекту

- data - для даних, за допомогою яких ми можемо взаємодіяти з телеграмом та зберігати інформацію по проекту;
- handlers - для файлів, що будуть містити обробники подій для відслідковування команд та реагування на дії користувача;
- filters - в ній містяться файли з фільтрами для введених даних від користувача;
- middlewares - для зберігання файлів з обробниками команд перед їх виконанням;
- states - для файлів, в яких створені класи команд для машини станів;
- utils - папка для додаткових структурних елементів, наприклад: функцій, класів і тд;
- img - папка для зберігання зображень, що будуть використовуватись в проекті;

- requirements.txt - файл з інформацією про версії бібліотек, що використовуються в проєкті;
- app.py - основний файл, де запускається в роботу бот за допомогою методу start\_polling об'єкта executor;
- loader.py - файл, де створюються об'єкти Bot та Dispatcher;
- .env - прихований файл з токенами взаємодії бота з серверами Telegrama та платіжною системою;
- .gitignore - прихований файл, який допомагає вказати файли або папки, які не потрібно вивантажувати на сервіс github.com.

Наступним кроком буде створення віртуального середовища для встановлення необхідних версій бібліотек, які записані в файлі requirements.txt. Встановити дані бібліотеки можливо за допомогою терміналу та команди pip3 для третьої версії Python (рис. 3.4).

```

→ diploma cat chat_bot/requirements.txt
aiogram==2.7
aiohttp==3.6.2
aioredis==1.3.1
python-dotenv

→ diploma python3 -m venv bot_env
→ diploma ls
bot_env chat_bot
→ diploma source ./bot_env/bin/activate
(bot_env) → diploma pip3 install -r ./chat_bot/requirements.txt
Collecting aiogram==2.7 (from -r ./chat_bot/requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/ae/20/2fb5e5bccf77bd5883b310f6ba431550d8462eec7bfada556866c65ba06c/aiogram-2.7-py3-none-any.whl
Collecting aiohttp==3.6.2 (from -r ./chat_bot/requirements.txt (line 2))
  Using cached https://files.pythonhosted.org/packages/c2/f7/f0ad3dbace4762fef5d80aa4124b41bf218e4c4dd6d387a86cede707d9a4/aiohttp-3.6.2-py3-none-any.whl

```

Рис. 3.4. Створення віртуального середовища та встановлення необхідних бібліотек

На даному етапі ми маємо готове середовище для розробки чат-бота за допомогою мови програмування Python. Тепер потрібно зареєструватись на веб-сервісі SendPulse, щоб створювати чат-бота в конструкторі (див. рис. 3.5). Сервіс на безоплатній основі надає можливість створити одного чат-бота, який буде приймати до 10000 повідомлень на місяць.

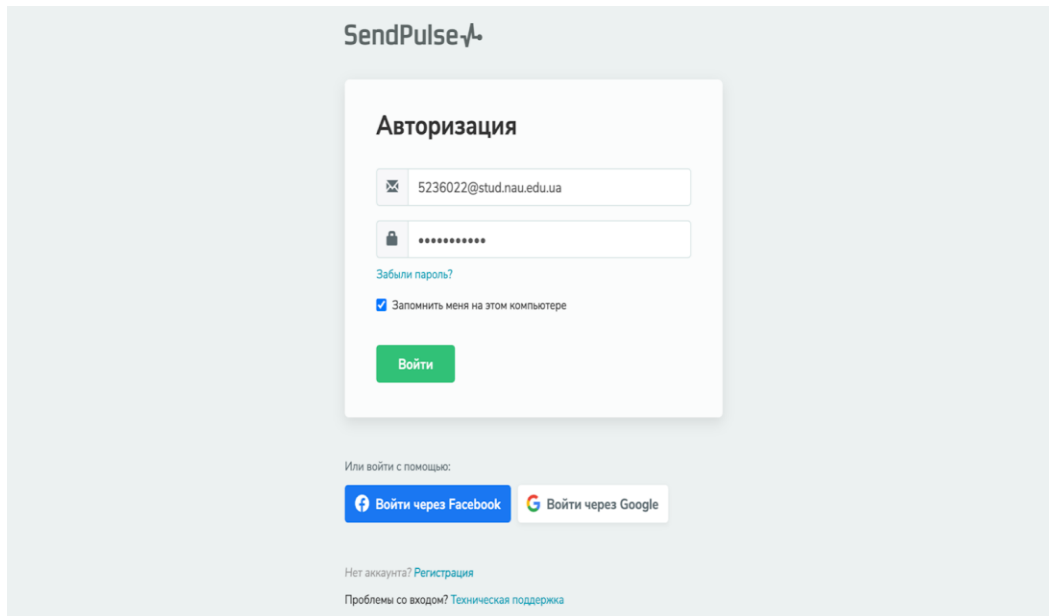


Рис. 3.5. Реєстрація на сервісі SendPulse

### 3.2. Порівняння створення чат-ботів за допомогою Python та SendPulse

Створення чат-бота розпочинається з отримання токена для взаємодії з Telegram Bot Арі через бота @BotFather (рис. 3.6). Токен виступає “ключем” для отримання даних з серверу Telegram.

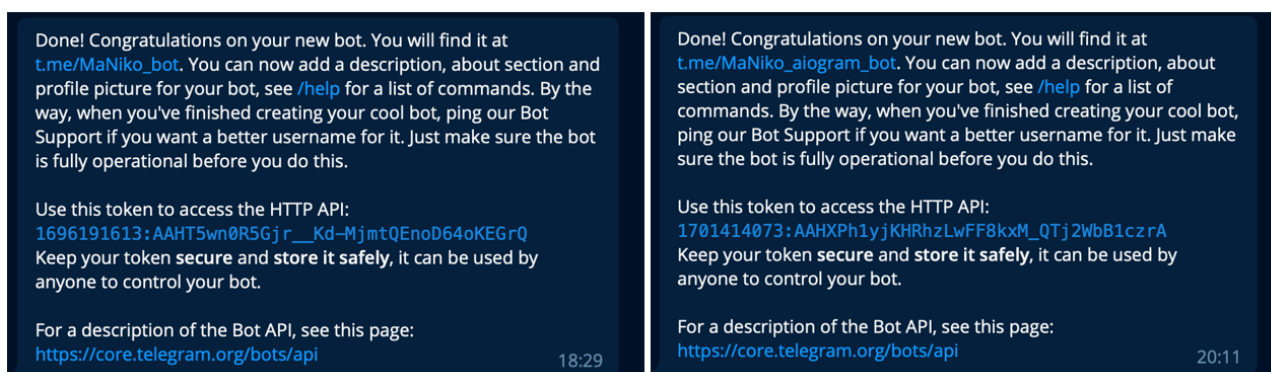


Рис. 3.6. Отримання токенів для чат-ботів

Після отримання токенів, потрібно зв’язати їх з написаним кодом (1) або сервісом для створення чат-ботів (2), що допомагає функціонувати ботам та отримувати оновлення інформації з серверів Telegram після взаємодії з користувачем (див. рис. 3.7).

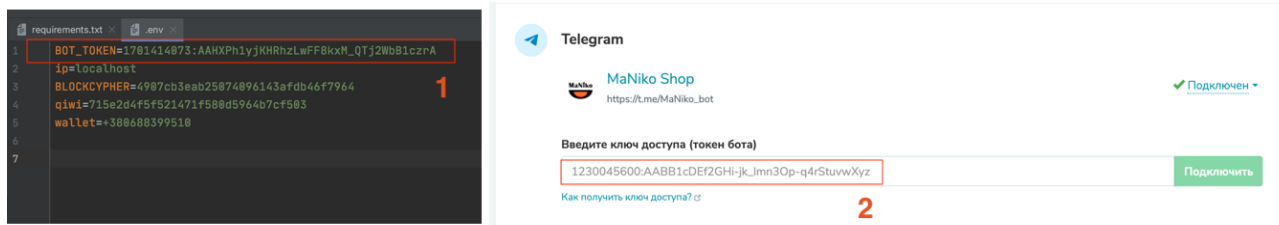


Рис. 3.7. Встановлення токенів для чат-ботів

Створення бота за допомогою Python з використанням aiogram заключається в створенні об'єктів (рис. 3.8):

- a) Bot - зв'язується з токеном Telegram та надає змогу взаємодіяти на пряму з користувачем, тобто відправляти повідомлення та різноманітний контент;
- b) MemoryStorage - в ньому зберігаються кожен користувач та його команди і взаємодія з чат-ботом;
- c) Dispatcher - це обробник подій, який опрацьовує вхідні оновлення: повідомлення, відредаговані повідомлення, запити тощо.

```

requirements.txt x .env x app.py x loader.py x
1 from aiogram import Bot, Dispatcher, types
2   from aiogram.contrib.fsm_storage.memory import MemoryStorage
3
4 from data import config
5
6 bot = Bot(token=config.BOT_TOKEN, parse_mode=types.ParseMode.HTML)
7 storage = MemoryStorage()
8 dp = Dispatcher(bot, storage=storage)
9 |
  
```

Рис. 3.8. Створення основних об'єктів чат-боту

Створення чат-бота в сервісі SendPulse є значно легшим. Потрібно тільки ввести токен бота та підтвердити з ним взаємодію через команду /start, з якої розпочинається робота з будь-яким чат-ботом в Telegram. Після цього відкривається конструктор, в якому будуть створенні команди за замовчуванням (рис. 3.9).



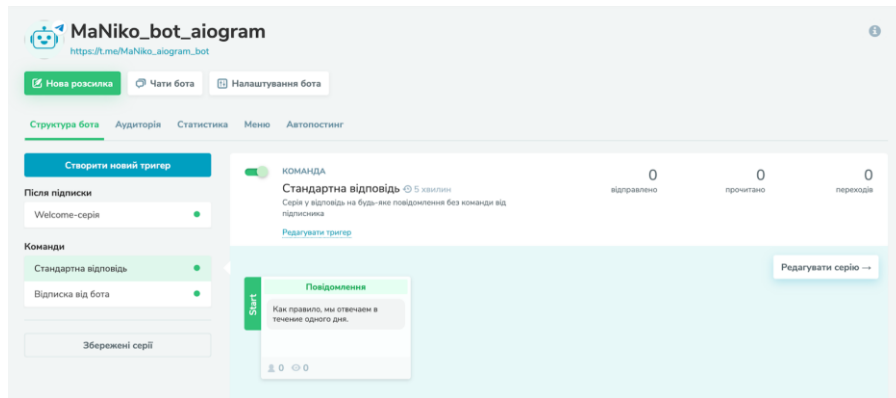


Рис. 3.9. Створений чат-бот за допомогою сервісу SendPulse

Запустивши бота в сервісі SendPulse, віртуальний помічник вже є ГОТОВИМ до роботи з клієнтами. Проте для запуску чат-бота, написаного на Python, потрібно імпортувати об'єкт `executor` з фреймворку `aiogram` та запустити його в режимі тривалого очікування запиту за допомогою методу `start_polling` (див. табл. 3.1).

Таблиця 3.1.

### Опис параметрів методу `start_polling`

Назва параметру:	Можливі прийняті значення:
<code>dispatcher</code>	<code>Optional[asyncio.AbstractEventLoop]</code>
<code>loop</code>	<code>Optional[asyncio.AbstractEventLoop]</code> (default: <code>None</code> )
<code>skip_updates</code>	<code>bool</code> (default: <code>None</code> )
<code>reset_webhook</code>	<code>bool</code> (default: <code>True</code> )
<code>on_startup</code>	<code>Optional[Callable]</code> (default: <code>None</code> )
<code>on_shutdown</code>	<code>Optional[Callable]</code> (default: <code>None</code> )
<code>timeout</code>	<code>int</code> (default: <code>20</code> )
<code>relax</code>	<code>float</code> (default: <code>0,1</code> )

Назва параметру:	Можливі прийняті значення:
allowed_updates	Optional[List[str]] = None

Найголовнішим параметром для даного методу є dispatcher (рис. 3.10), який містить в собі всі обробники подій, створені в коді. При отриманні оновлення від серверу про дії користувача, саме диспетчер реагує на це та знаходить необхідний обробник подій (handler).

```

15 ▶ if __name__ == '__main__':
16     from aiogram import executor
17     from handlers import dp
18
19     executor.start_polling(dp, on_startup=on_startup)
20

```

Рис. 3.10. Запуск бота за допомогою об'єкту executor

Наступним етапом в розробці чат-боту є створення команд, які користувач може виконувати, ланцюгів взаємодії бота з клієнтом та обробника подій в разі невідомих команд.

Команди являють собою функції, які «обернені» декоратором `@dp.message_handler()`. За допомогою нього і відбувається аналіз вхідних команд або тексту. Декоратор приймає на вхід метод парсингу вхідних даних. Це може бути назва команди, пошук за ключовим словом, стан, який надає клієнту машина станів (рис. 3.11).

```

:param commands: list of commands
:param regexp: REGEXP
:param content_types: List of content types.
:param custom_filters: list of custom filters
:param kwargs:
:param state:
:param run_task: run callback in task (no wait results)
:return: decorated function

```

Рис. 3.11. Опис декоратора `@dp.message_handler()`

За допомогою dispatcher та «фільтрів», якими виступають декоратори, на будь-яку команду користувача знаходиться необхідний обробник подій.

Команди ж в Sendpulse створюються за допомогою кнопки «Створити новий тригер». Після чого отримуємо меню налаштування команди, де вказуємо (рис. 3.12):

- a) повторний виклик тригера - обставини, за яких буде викликано команду повторно;
- b) команда - вказуємо назву команди або ж ключові слова, на які буде реагувати новостворений тригер;
- c) повторний виклик - інтервал часу, коли введення однакових команд буде ігноруватись.

Хендлери є основними елементами взаємодії, за допомогою них створюють ланцюги користувацьких повідомлень. Саме handler реагує на оновлення подій, які надходять ботові від серверу Telegram, та відповідає на ці зміни в залежності від своїх функцій. Проте взаємодія з клієнтом відбувається з використанням повідомлень, які можуть містити в собі як текстову інформацію, так і медіа, аудіо файли. Також важливим елементом є клавіатура, яка спонукає користувача на дії.

Створення тригера

Повторний виклик тригера:  Після підписки  
 Подія A360 <sup>NEW</sup>  
 Команда

Команди: Запустити серію, якщо від підписника отримано повідомлення з текстом

Слова чи фрази без врахування регістру. Наприклад: вітаю, привіт, яка вартість.  
При неточному збігу підписнику прийде уточнення

Повторний виклик: Ігнорувати введення однакових команд і виконувати не частіше одного разу в

Чати

Рис. 3.12. Створення тригера-команди в SendPulse

Клавіатури є двох типів: `InlineKeyboardMarkup` (див. рис. 3.13) та `ReplyKeyboardMarkup` (див. рис. 3.14), які в свою чергу складаються з кнопок `InlineKeyboardButton` та `ReplyKeyboardButton` відповідно. За допомогою них користувач може обирати необхідний йому шлях взаємодії з віртуальним помічником. Саме клавіатура є провідником по ланцюгу повідомлень. Кожна кнопка пов'язана і має результат виконання.

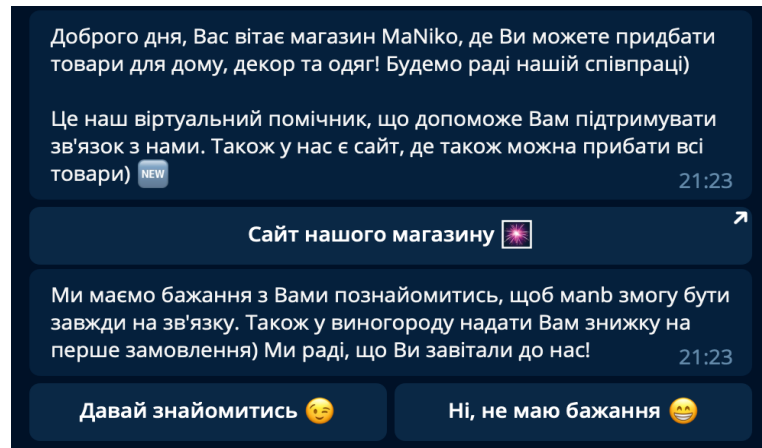


Рис. 3.13. `InlineKeyboardMarkup`

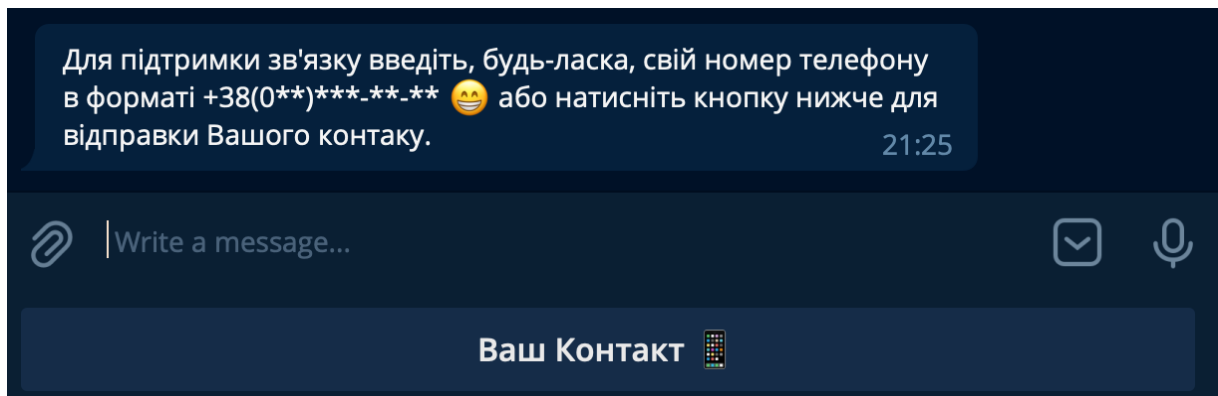


Рис. 3.14. `ReplyKeyboardMarkup`

Елемент `Keyboard` приєднується до повідомлення, що надсилається клієнту. Створення клавіатури в сервісі `SendPulse` відбувається при додаванні елемента «Повідомлення» в ланцюг повідомлень (див. рис. 3.15).

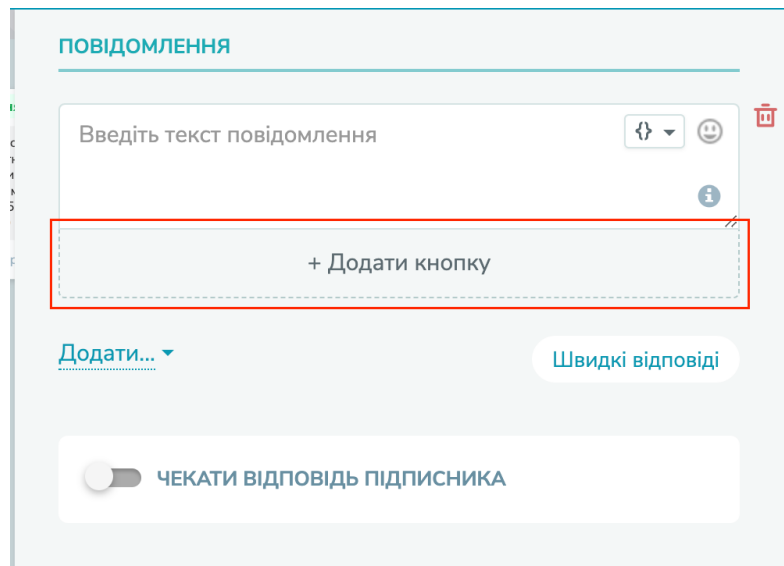


Рис. 3.15. Додавання кнопки до повідомлення в SendPulse

Додавання клавіатури до повідомлення в коді відбувається при передаванні створеної клавіатури в якості параметру в метод об'єкту `bot.send_message()` (див. рис. 3.16).

```
36
37 cancel_chat = InlineKeyboardMarkup(row_width=1, inline_keyboard=[])
38
39 site = InlineKeyboardButton(text="Сайт нашого магазину 📄", url="https://maniko.com.ua")
40
41 cancel_chat.add(site, go_to_menu)
42
43 @dp.callback_query_handler(state=Admin.Close_chat)
44 async def close_choose_chat(call: CallbackQuery):
45     await call.answer()
46     await dp.storage.wait_closed()
47     # await dp.storage.wait_closed()
48     await bot.send_message(admins[0], "Чат з клієнтом було завершено!")
49     await bot.send_message(queue_to_communication[0], "Чат з менеджером був закінченим, дякуємо за співпрацю!",
50                             reply_markup=cancel_chat)
51     del dict_to_communication[str(queue_to_communication[0])]
52     del queue_to_communication[0]
```

Рис. 3.16. Додавання кнопки до повідомлення в коді проекту

Ланцюг повідомлень складається з обробників команд або вхідної інформації, якою можуть бути зворотні дані (callback data) з обраної клієнтом кнопки на клавіатурі. Саме це допомагає заздалегідь створити необхідні хендлери і проводити користувача по необхідному йому шляху, що можна побачити в серії повідомлень в SendPulse (див. рис. 3.17).

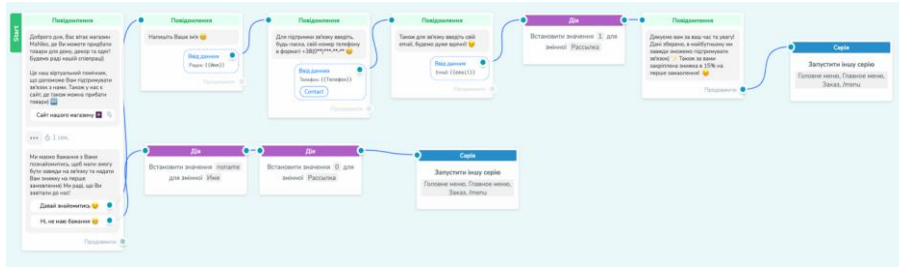


Рис. 3.17. Welcome-серія повідомлень в створеному чат-боті сервісом SendPulse

Ланцюг повідомлень у кодї має таку ж послідовність, що явно виділяє перевагу сервісу по розробці чат-ботів над розробкою на мові програмування. Розроблена логіка в сервісі SendPulse слугує базою для створення віртуального помічника за допомогою Python.

Чат-бот має такі серії повідомлень, що були розроблені в SendPulse:

- a) welcome-серія - ланцюг повідомлень після введення команди /start;
- b) стандартна відповідь - відповідь бота, якщо введене повідомлення або ж команда невідомі для бота;
- c) відписка від бота - взаємодія з клієнтом при введенні команди /stop;
- d) акційні пропозиції - ланцюг повідомлень персональних або акційних пропозицій для користувачів бота;
- e) головне меню - головне меню бота;
- f) faq - меню найпоширеніших запитань;

### 3.3. Тестування розроблених чат-ботів

Перевірка працездатності чат-ботів заключалася в ручному тестуванні, що відбувалось під час створення нового елемента або ланцюга повідомлень кожного з ботів. Тестування розроблених систем відбувались по заздалегідь продуманим тест-кейсам та з урахуванням поставлених вимог до віртуальних помічників, а саме:

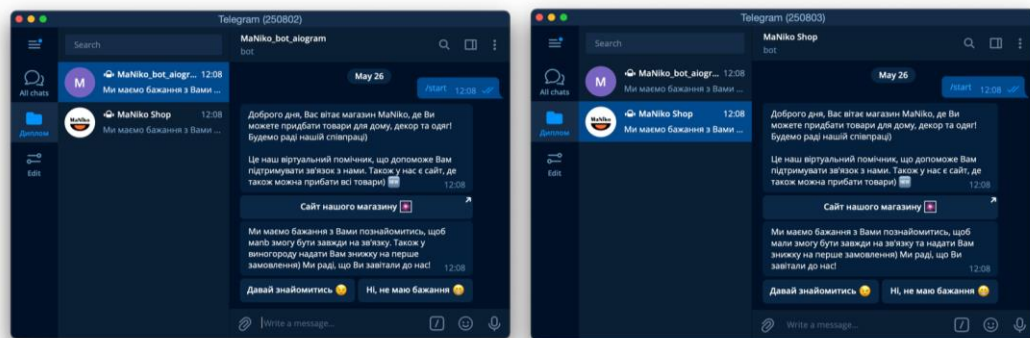
- a) інформативність відповідей;
- b) легкість використання;
- c) швидкість відповіді;
- d) якість обробки повідомлень від клієнта;

е) логічність в ланцюгах повідомлень.

Тестування чат-ботів відбувалось за допомогою двох пристроїв: смартфон Google Pixel 4a 6/128 GB Just Black та ноутбуку Apple Macbook Pro 2018 15,6” 512 GB Space Grey.

Користування ботом розпочинається з команди /start, яка розпочинає ланцюг повідомлень «Welcome-серія» (рис. 3.18).

Чат-боти після введення команди /start

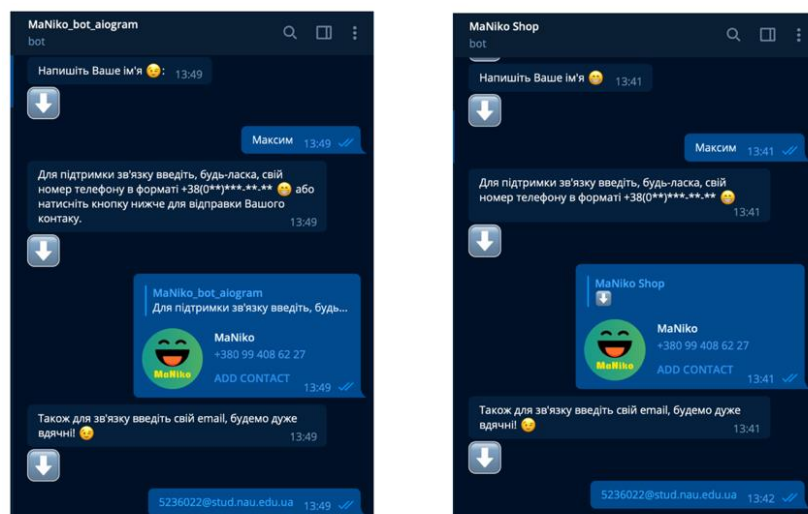


На мові програмування Python

Створений через сервіс SendPulse

Рис. 3.18. Чат-боти після введення команди /start

Саме ця послідовність знайомить нас з клієнтом: якщо він обере відповідну кнопку на повідомленні, та надсилає його дані на сервер сайту компанії або перенаправляє його в головне меню чат-боту з анонімними даними. За допомогою «Welcome-серії» ми отримуємо від користувача його ім'я, номер телефону та електронну пошту (рис. 3.19).



На Python

На SendPulse

Рис. 3.19. Welcome-серія в чат-ботах

Після «знайомства» з користувачем або ж після відмови користувача проходити по «Welcome-серії», його перенаправляє в головне меню, де він може обрати подальшу взаємодію з чат-ботом. Це може бути зв'язок з менеджером з приводу замовлення, показ акційних пропозицій або ж меню FAQ з найпоширенішими запитаннями (рис. 3.20).

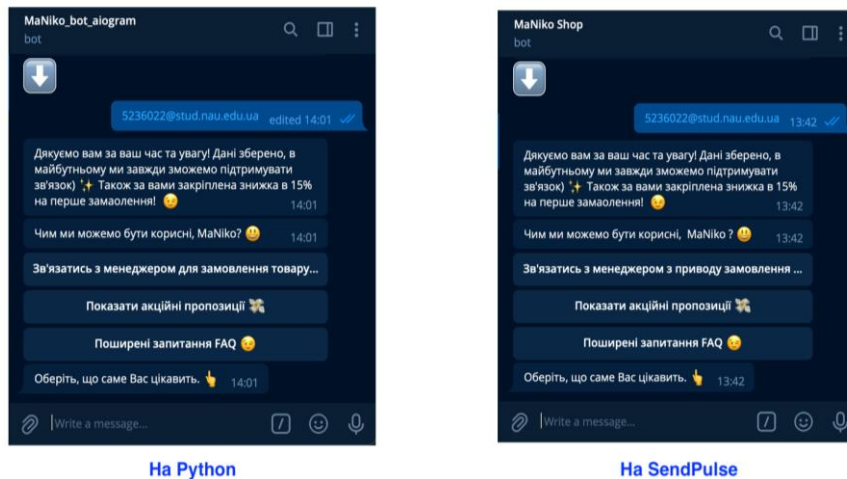


Рис. 3.20. Головне меню

Важливим є тестування на стійкість системи при великій кількості запитів від користувача. Чат-бот повинен опрацювати всі запити та повідомити клієнта про неможливість надати йому відповідь на невідому команду чи запит (рис. 3.21). В цьому випадку віртуальний помічник буде також перенаправляти користувачів в головне меню.

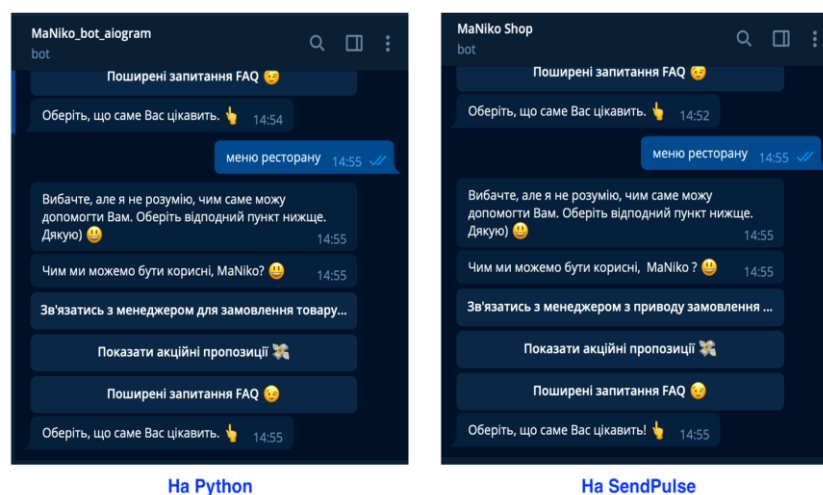


Рис. 3.21. Стандартна відповідь при невідомих командах



При великому навантаженні, тобто при спамі, коли людина посилає велику кількість повідомлень, чат-боти реагують двома способами:

- відповідають, що клієнт подає багато запитів на сервер;
- через два повідомлення про спам перестають приймати запити від даного користувача на 10 хвилин (рис. 3.22).

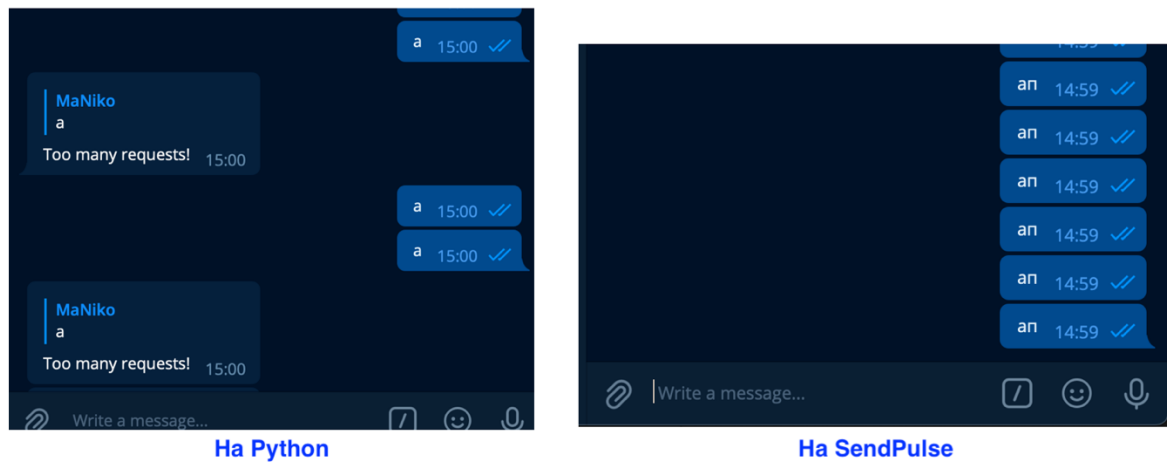
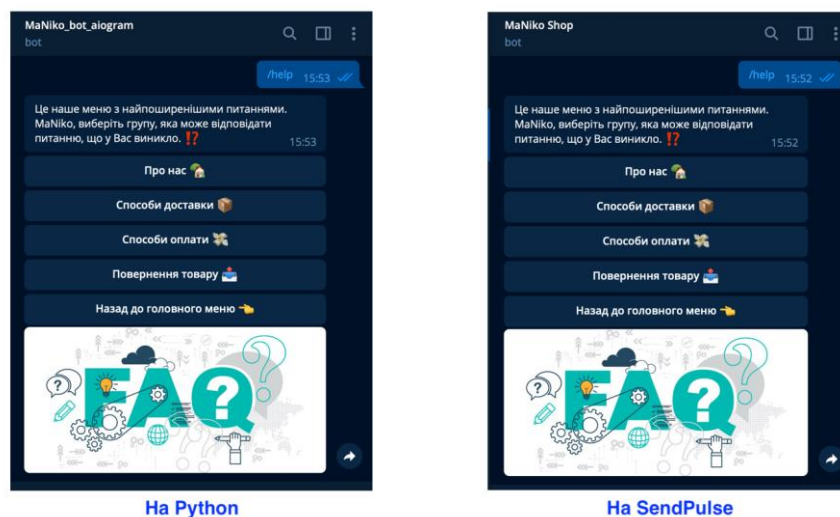


Рис. 3.22. Реакція чат-ботів при спамі

Використовуючи віртуального помічника, клієнт перш за все має намір дізнатись інформацію по питанню, що в нього виникло. Саме для цих цілей було розроблено розділ FAQ з відповідями на найпоширеніші питання (рис. 3.24) для користувачів задля пришвидшення зворотнього зв'язку від інтернет-магазину.



### Рис. 3.23. Меню чат-ботів FAQ

Інформація в цьому розділі поділена на групи, які в свою чергу поділені на підгрупи для швидкого пошуку та зручного користування довідкою. Кожна з груп має клавіатуру, яка допоможе повернутись до головно меню бота, до кореневого розділу FAQ або ж до підгрупи обраного розділу.

#### **Висновки до розділу**

У третьому розділі розроблено чат-ботів на мові програмування Python з використанням фреймворку aiogram та за допомогою сервісу SendPulse.

Розглянуто структуру проекту, завдяки якій розробка віртуального помічника значно полегшується та надає змогу масштабувати створений продукт. Також варто відзначити, що зручна структура допомагає швидше орієнтуватись в проекті та знаходити необхідні елементи.

Показана різниця в реалізації двох чат-ботів, розглянуті найголовніші структурні компоненти, з яких складається вся взаємодія між користувачем та віртуальним помічником, а саме: обробники подій, клавіатури користувача, відправка медіа-, аудіо-файлів та текстових повідомлень. Варто відмітити, що правильна реалізація та побудова веб-сервісу допомагає пришвидшити його роботу з користувачем.

Проведено тестування на двох пристроях, а також показана взаємодія ботів при введенні запитів від користувача. Показано повний цикл взаємодії клієнта з віртуальним помічником, а також відображено фільтрацію повідомлень або команд задля зменшення навантаження серверу, якщо користувач намагається надіслати одночасно багато повідомлень.

## ВИСНОВКИ

Технологічний прогрес сприяє виведенню кожної компанії на якісно новий рівень у комунікації з клієнтами. Одним із методів спілкування стають чат-боти, і їх використання набиратиме обертів у найближчому майбутньому.

Чат-боти набирають все більшої популярності серед власників та керівників бізнесу незалежно від сфери діяльності компанії. Це пояснюється тим, що віртуальні помічники забезпечують для компанії цілодобову безперебійну технічну підтримку, допомагають охопити більше покупців, збільшують швидкість відповіді на поширені запитання від клієнтів та надають можливість тісно співпрацювати з менеджерами компанії в разі необхідності, що в разі збільшує ефективність роботи працівників.

Компаніям слід зосередитися на швидкості відповідей та встановленні «живого» спілкування, використовуючи чат-боти, адже клієнтам важливо відчувати значущість при обслуговуванні. Завдяки боту можливо в разі примножити позитивний відгук про сервіс бізнесу від його клієнтів.

У ході дослідження теми дипломної роботи було виявлено, що широкий спектр можливостей чат-ботів дає змогу застосовувати їх у майже будь-якому бізнесі саме завдяки гнучкості створення. Також проаналізовано сфери використання віртуальної підтримки. Особливо дана технологія корисна компаніям, яким важливо регулярно інформувати своїх клієнтів про важливі заходи, акції або ж про технічні неполадки в роботі підприємства.

Також було встановлено, що технічна підтримка направлена на допомогу клієнтам якомога швидше досягти своєї кінцевої мети: отримання додаткової інформації чи допомога з покупкою товарів або послуг.

Слід зауважити, що віртуальний помічник має відповідати основним функціональним вимогам та архітектурним особливостям для впровадження його в бізнес-процеси компанії. Інтегрування в компанію даної системи дасть приріст в швидкості та якості обробки онлайн-замовлень через мережу Інтернет. Саме тому дана технологія принесе бізнесу неабияку користь та

зеконотить кошти на подальший розвиток. Вагомий плус у використанні ботів – це їх ефективність взаємодії з клієнтами на будь-яких масштабах.

Принципова відмінність між розробкою чат-ботів за допомогою конструктора та мови програмування полягає у складності реалізації поставленої задачі.

Створення віртуального помічника за допомогою Python потребує навичок програмування, а головне – проектування масштабованих систем задля подальшого розвитку програмного забезпечення. Це дає можливість впроваджувати нові структурні елементи незалежно від їх складності в реалізації.

Проте використання веб-сервісів для реалізації чат-боту буде корисним для створення тестового варіанту, який можна використати для аналізу інтегрування даного підходу в бізнес-модель компанії.

Показана різниця в реалізації віртуальних помічників, розглянуті найголовніші компоненти, з яких складається вся взаємодія між користувачем та ботом. Проведено тестування на двох пристроях, а також показана взаємодія ботів при введенні запитів від користувача. Згідно з аналізом вимог до чат-ботів та їх функціонування, спроектовано необхідний інтерфейс для ботів, які відповідають поставленому завданню. У результаті це дає функціональні та легкі у використанні чат-боти, які відповідають на усі загальні запити користувачів.

Кожна компанія повинна зважити усі переваги та недоліки використання віртуальних помічників у продажах, але фактом залишається те, що вони вдало інтегруються у систему комунікації з клієнтами.