

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ**
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

«До захисту допущено»

Завідувач кафедри

_____ В. В. Койбічук

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавра
(бакалавр / магістр)

зі спеціальності 051 «Економіка»,
(код та назва)

освітньо-професійної програми «Економічна кібернетика та бізнес аналітика»
(освітньо-професійної / освітньо-наукової) (назва програми)

на тему: «Автоматизація взаємодії клієнтів та менеджерів у сегменті середнього та малого бізнесу»

Здобувача (ки) групи ЕКз-01с
(шифр групи)

Костюшко Іван Олександрович
(прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

Іван КОСТЮШКО
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник

асистент, PhD, Олександр КУШНЕРЬОВ
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ
Завідувачка кафедри
доцентка, к.е.н.
_____ В. В. Койбічук
“ ___ ” _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
спеціальність 051 Економіка (Економічна кібернетика та бізнес аналітика)
студенту 4 курсу, групи ЕКз-01с

Костюшко Іван Олександрович

1. Тема роботи «Автоматизація взаємодії клієнтів та менеджерів у сегменті середнього та малого бізнесу» затверджена наказом Про затвердження тем і керівників кваліфікаційних робіт наказ №0481-VI від 07.05.2024 року.
2. Термін подання студентом закінченої роботи «1» червня 2024 року
3. Мета кваліфікаційної роботи: розробка Telegram-бота для підтримки клієнтів із розділенням функціоналу для адміністраторів та користувачів, що дозволить оптимізувати процеси взаємодії з клієнтами та підвищити ефективність роботи служби підтримки.
4. Об'єкт дослідження є процеси взаємодії між користувачами та операторами служби підтримки через месенджер Telegram.
5. Предмет дослідження є методи автоматизації цих процесів за допомогою розробки Telegram-бота.
6. Кваліфікаційна робота базується на комплексному підході, що включає аналіз наукових публікацій вітчизняних та іноземних дослідників, вивчення існуючих методологій, а також застосування бібліотек мови програмування Python для досліджень в галузі автоматизації.
7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1. Аналіз поточного стану взаємодії клієнтів та менеджерів

У розділі 1: 1.1 Огляд існуючих методів взаємодії клієнтів та менеджерів.

1.2 Аналіз сучасних месенджерів.

1.3 Технічне завдання до розробки модуля взаємодії клієнтів та менеджерів.

Розділ 2. Розробка модуля автоматизації взаємодії клієнтів та менеджерів

У розділі 2: 2.1 Стек технологій які використовувався при розробці модуля автоматизація взаємодії клієнтів та менеджерів.

2.2 Описання структури і функціоналу модуля автоматизація взаємодії клієнтів та менеджерів.

2.3 Оцінка очікуваного ефекту від впровадження модуля автоматизація взаємодії клієнтів та менеджерів.

8. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Кушнерьов О.С., PhD, асистент	01.04.2024	07.05.2024
2	Кушнерьов О.С., PhD, асистент	10.05.2024	27.05.2024

9. Дата видачі завдання: «1» квітня 2024 року

Керівник кваліфікаційної роботи.

_____ (підпис)

О.С. Кушнерьов

(ініціали, прізвище)

Завдання до виконання одержав.

_____ (підпис)

І.О. Костюшко

(ініціали, прізвище)

АНОТАЦІЯ
кваліфікаційної роботи на тему
«АВТОМАТИЗАЦІЯ ВЗАЄМОДІЇ КЛІЄНТІВ ТА МЕНЕДЖЕРІВ
У СЕГМЕНТІ СЕРЕДНЬОГО ТА МАЛОГО БІЗНЕСУ»
студента Костюшко Іван Олександрович

Актуальність теми, обраної для дослідження. У сучасному світі, де цифровізація стає ключовим елементом бізнесу, ефективна взаємодія між клієнтами та менеджерами набуває особливої важливості. Автоматизація цих процесів через месенджери, зокрема через Telegram, дозволяє покращити обслуговування клієнтів та оптимізувати роботу служби підтримки.

Мета кваліфікаційної роботи бакалавра полягає в розробці Telegram-бота для підтримки клієнтів із розділенням функціоналу для адміністраторів та користувачів, що дозволить оптимізувати процеси взаємодії з клієнтами та підвищити ефективність роботи служби підтримки.

Об'єктом дослідження є процеси взаємодії між користувачами та операторами служби підтримки через месенджер Telegram.

Предметом дослідження є методи автоматизації цих процесів за допомогою розробки Telegram-бота.

Методи дослідження включають аналітичну роботу по дослідженню існуючих рішень для автоматизації взаємодії, вибір технологій та програмування самого бота. Використання Python та фреймворку aiogram для розробки, SQLite для бази даних.

Інформаційна база кваліфікаційної роботи включає дослідження літератури та аналіз існуючих методів взаємодії, вивчення можливостей месенджерів, зокрема Telegram, та використання програмних продуктів таких як PyCharm, SQLite, aiogram.

Практичний результат полягає у створенні Telegram-бота, який оптимізує взаємодію між клієнтами та менеджерами, підвищуючи ефективність роботи служби підтримки.

Теоретичні результати включають аналіз існуючих рішень та обґрунтування вибору підходів для автоматизації.

Новизна роботи полягає в адаптації сучасних технологій та інструментів програмування для специфіки взаємодії в сегменті середнього та малого бізнесу через месенджери, зокрема Telegram.

Результати кваліфікаційної роботи можуть бути використані середнім та малим бізнесом для покращення обслуговування клієнтів та оптимізації роботи служби підтримки. Впровадження розробленого Telegram-бота дозволяє скоротити час на обробку запитів клієнтів та підвищити загальну задоволеність споживачів послуг.

Рекомендується впроваджувати автоматизовані рішення для взаємодії з клієнтами у вигляді чат-ботів на базі Telegram для поліпшення швидкості та якості обслуговування.

Ключові слова: автоматизація, Telegram-бот, взаємодія з клієнтами, комунікація, чат-бот, Python, aiogram.

Зміст кваліфікаційної роботи викладено на 30 сторінках. Список використаних джерел із 32 найменувань, розміщений на 3 сторінках. Робота містить 14 малюнків, 1 таблицю, 1 формулу, 4 додатки, розміщених на 12 сторінках.

Рік виконання кваліфікаційної роботи – 2024 рік.

Рік захисту роботи – 2024 рік.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПОТОЧНОГО СТАНУ ВЗАЄМОДІЇ КЛІЄНТІВ ТА МЕНЕДЖЕРІВ.....	9
1.1 Огляд існуючих методів взаємодії клієнтів та менеджерів.....	9
1.2 Аналіз сучасних месенджерів	14
1.3 Технічне завдання до розробки модуля взаємодії клієнтів та менеджерів.....	18
2 РОЗРОБКА МОДУЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ КЛІЄНТІВ ТА МЕНЕДЖЕРІВ.....	20
2.1 Стек технологій які використовувався при розробці модуля автоматизація взаємодії клієнтів та менеджерів	20
2.2 Описання структури і функціоналу модуля автоматизація взаємодії клієнтів та менеджерів	26
2.3 Оцінка очікуваного ефекту від впровадження модуля автоматизація взаємодії клієнтів та менеджерів	33
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37
ДОДАТКИ.....	40

ВСТУП

У сучасному світі, де цифровізація стає не лише перевагою, а й необхідністю для бізнесу, питання ефективної взаємодії між клієнтами та менеджерами набуває особливої важливості. З урахуванням глобалізації ринку та зростаючої конкуренції, здатність швидко та ефективно реагувати на запити клієнтів стає критичним фактором успіху бізнесу. Швидкість змін у технологічному секторі вимагає від компаній бути гнучкими та інноваційними, адаптуючи сучасні інструменти для підтримки своєчасної взаємодії.

У сегменті середнього та малого бізнесу, де кожна можливість оптимізації процесів має велике значення, автоматизація взаємодії стає ключовим фактором успіху. Telegram, будучи одним з найпопулярніших месенджерів із високим рівнем функціональності, є ідеальною платформою для автоматизації процесів підтримки та забезпечення оперативного вирішення запитів користувачів. Зростання очікувань споживачів щодо швидкості відповідей та якості обслуговування підкреслює важливість впровадження таких рішень.

Особливо важливою є ідея розробка Telegram-бота для живого чату з менеджером. Враховуючи високий обсяг завдань, з якими стикаються менеджери, та критичне значення клієнтської підтримки в сучасному бізнес-середовищі, важко переоцінити потенційні переваги автоматизованого чату для полегшення обміну інформацією та забезпечення швидкої реакції на потреби клієнтів. Розширення мобільного доступу та поширення смартфонів збільшує потенціал досягнення великої аудиторії через такі платформи, як Telegram, що робить автоматизацію не лише зручною, але й ефективною стратегією взаємодії з клієнтами.

Тому метою даного дослідження є розробка Telegram-бота для підтримки клієнтів із розділенням функціоналу для адміністраторів та

користувачів, що дозволить оптимізувати процеси взаємодії з клієнтами та підвищити ефективність роботи служби підтримки.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналітичну роботу, що включає дослідження існуючих рішень для автоматизації взаємодії між клієнтами та службою підтримки;
- розробити технічне завдання для створення Telegram-бота, з урахуванням специфіки взаємодії користувачів та операторів служби підтримки;
- вибрати оптимальний стек технологій;
- розробити та протестувати Telegram-бот.

Об'єктом дослідження є процеси взаємодії між користувачами та операторами служби підтримки через месенджер Telegram.

Предметом дослідження є методи автоматизації цих процесів за допомогою розробки Telegram-бота.

Результати кваліфікаційної роботи можуть використовуватися у сегменті середнього та малого бізнесу, які прагнуть покращити обслуговування клієнтів і оптимізувати роботу своїх служб підтримки.

Для розробки Telegram-бота були використані такі програмні продукти: PyCharm для редагування та написання коду, SQLite як система управління базами даних, aiogram як повністю асинхронний фреймворк для розробки чат-ботів.

1 АНАЛІЗ ПОТОЧНОГО СТАНУ ВЗАЄМОДІЇ КЛІЄНТІВ ТА МЕНЕДЖЕРІВ

1.1 Огляд існуючих методів взаємодії клієнтів та менеджерів

В сучасному динамічному бізнес-середовищі, яке характеризується стрімким технологічним прогресом та постійними змінами у споживацьких уподобаннях, відбувається неупинна еволюція засобів взаємодії між клієнтами та менеджерами. Така трансформація актуалізує потребу в необхідності адаптації бізнес-структур та пошуку інноваційних підходів до комунікації з метою підтримки високого рівня задоволеності клієнтів та їхньої лояльності. Важливість цього процесу не можна недооцінювати, адже вона є критично важливою для сталого розвитку та підвищення конкурентоспроможності підприємства [2]. Інтеграція сучасних цифрових інструментів та платформ в комунікаційну стратегію дозволяє покращити розуміння потреб клієнтів, оптимізувати взаємодію та ефективність обслуговування. При цьому, особлива увага приділяється зростаючій ролі соціальних медіа та мобільних технологій, що відкривають нові можливості для двосторонньої комунікації, швидкого реагування на запити клієнтів та персоналізації пропозицій. Зберігаючи при цьому баланс між традиційними та цифровими методами комунікації, компанії можуть значно покращити взаємодію з клієнтами, що неминуче призведе до зростання їх задоволеності та лояльності [2].

На цьому етапі, особливо актуальним стає дослідження та аналіз існуючих методів комунікації, які можна класифікувати на традиційні методи взаємодії та сучасні методи. Традиційні методи, такі як особисті зустрічі, телефонні дзвінки та листування електронною поштою, довгий час слугували надійними засобами для підтримки відносин з клієнтами. Особисті зустрічі вважаються одним з найефективніших способів комунікації, оскільки вони дозволяють обговорювати важливі питання віч-на-віч, виявляти невербальні

сигнали, такі як жести, міміка, тон голосу, що сприяє кращому розумінню та встановленню довірливих відносин. Крім того, особисті зустрічі часто сприймаються як знак поваги та значення, яке приділяється співрозмовнику чи проекту [16].

Телефонні дзвінки є ще одним традиційним методом комунікації, який дозволяє швидко та ефективно обговорювати питання, не чекаючи особистої зустрічі або відповіді електронною поштою. Цей метод забезпечує миттєвий зворотний зв'язок і можливість розв'язати проблему в режимі реального часу, що особливо важливо в критичних ситуаціях [16].

Листування електронною поштою стало стандартом у професійній комунікації завдяки своїй зручності та ефективності. Електронні листи дозволяють вести документацію обговорень, легко ділитися файлами та інформацією, а також відстежувати історію кореспонденції. Цей метод є важливим для організації робочих процесів, планування зустрічей та координації проектів [16].

Враховуючи вищезазначене, традиційні методи комунікації залишаються незамінними у багатьох аспектах ведення бізнесу, незважаючи на стрімкий розвиток цифрових технологій. Вони допомагають підтримувати особистісний зв'язок, довіру та розуміння між партнерами, що є ключовими елементами для успішної ділової взаємодії.

Однак, з розвитком технологій та зміною споживацьких вподобань, комунікація зазнала значних змін [20].

Сучасні методи, включаючи соціальні мережі, мобільні застосунки, віртуальні конференції та відеодзвінки, онлайн-чати та чат-боти, пропонують нові можливості для взаємодії з аудиторією, забезпечуючи клієнтам зручність у виборі способу спілкування з компанією. Це не лише підвищує задоволеність клієнтів, але й забезпечує компаніям зворотний зв'язок в реальному часі, що є критично важливим для адаптації до змін у потребах та уподобаннях клієнтів [30].

Соціальні мережі стають важливим інструментом для підтримки зв'язку з клієнтами, дозволяючи компаніям швидко реагувати на запитання, збирати відгуки та розповсюджувати інформацію про продукти чи послуги. Мобільні застосунки забезпечують зручний доступ до послуг та інформації, покращуючи загальний досвід користувача і залученість.

Віртуальні конференції та відеодзвінки набули особливої актуальності в умовах глобалізації та розподілених команд, дозволяючи проводити зустрічі та обговорення без необхідності фізичної присутності, що сприяє економії часу та ресурсів [20].

Онлайн-чати та чат-боти відіграють ключову роль у забезпеченні швидкого доступу до інформації та підтримки, зменшуючи час очікування та підвищуючи задоволеність клієнтів. Ці інструменти дозволяють автоматизувати відповіді на часті запитання та забезпечувати персоналізовану взаємодію, що є важливим фактором у побудові довгострокових відносин з клієнтами [16].

Інтеграція сучасних цифрових методів комунікації у стратегію взаємодії з клієнтами відіграє ключову роль для досягнення успіху в динамічному бізнес-середовищі сьогодення. Швидкість реагування та гнучкість адаптації до змінних умов та вимог клієнтів стають вирішальними факторами у підтримці конкурентоспроможності компаній. У цьому контексті аналіз переваг та недоліків існуючих методів комунікації набуває особливої актуальності [20].

Аналіз традиційних та сучасних методів комунікації, представлений у таблиці 1.1, демонструє різноманіття доступних інструментів та підходів, кожен з яких має свої унікальні переваги та обмеження. Вибір методу комунікації має бути обумовлений специфікою взаємодії з клієнтами, цільовою аудиторією, а також стратегічними цілями та завданнями бізнесу.

Таблиця 1.1 – Аналіз переваг та недоліків існуючих методів комунікації

Метод комунікації	Переваги	Недоліки
Особисті зустрічі	Швидке вирішення складних питань через безпосереднє обговорення; підтримка сильних відносин і довіри між учасниками.	Значні витрати часу та ресурсів на організацію зустрічей; обмежена масштабованість та гнучкість у порівнянні з віртуальними методами.
Телефонні дзвінки	Швидкий спосіб комунікації для обговорення термінових питань; можливість чути інтонації та емоції співрозмовника.	Відсутність візуального контакту може ускладнити повне розуміння повідомлення; обмежена можливість документування розмови.
Листування електронною поштою	Записана історія спілкування, що полегшує пошук та відстеження інформації; можливість надсилання повідомлень у будь-який час, не вимагаючи негайної відповіді.	Можливість затримок у відповідях; ризик перенавантаження інформацією та спамом.
Соціальні мережі	Широкий охоплення та можливість швидкого розповсюдження інформації. велика кількість форматів для комунікації: текст, зображення, відео.	Ризик розповсюдження недостовірної інформації; можливість виникнення негативних коментарів та тролінгу.

Продовження таблиці 1.1

Метод комунікації	Переваги	Недоліки
Мобільні застосунки	Висока гнучкість та доступність комунікації; можливість інтеграції з іншими сервісами та функціями смартфонів.	Розкиданість інформації між різними застосунками; складність розробки та вимоги до оновлень.
Віртуальні конференції та відеодзвінки	Візуальний контакт та можливість ефективного обговорення без необхідності фізичної присутності; економія часу та ресурсів на подорожі.	Технічні проблеми; залежність від якості з'єднання.
Онлайн-чати та чат-боти	Швидкість отримання відповідей на стандартні питання; прямий зв'язок із менеджером.	Ризик незадоволення користувачів через стандартизовані відповіді; відсутність швидко відповіді у не робочі часи

Таким чином, розуміння переваг та обмежень кожного з методів дозволяє компаніям ефективно вибудовувати стратегії взаємодії з клієнтами, орієнтуючись на забезпечення високого рівня задоволеності та лояльності. Активне впровадження інноваційних технологій у комунікаційні процеси, таким чином, стає ключовим елементом стратегії розвитку бізнесу, що дозволяє залишатися на передовій ринкових змін та відповідати на виклики сучасного ринку. [2; 30].

1.2 Аналіз сучасних месенджерів

В умовах стрімкого розвитку технологій та збільшення обсягів інформації, месенджери стають ключовою складовою нашого повсякдення. Ці програми або застосунки, які дозволяють обмінюватися миттєвими повідомленнями через Інтернет [5], включають різноманітний функціонал, такий як текстовий чат, голосові повідомлення, відеодзвінки, передачу файлів та інші засоби комунікації, що робить їх невід'ємною частиною сучасного спілкування. Завдяки забезпеченню швидкого та зручного зв'язку між користувачами незалежно від їхнього місця перебування, месенджери суттєво спрощують комунікацію.

Аналіз ринку месенджерів є важливим для розуміння його динаміки та впливових факторів, особливо в контексті швидких змін у секторі комунікаційних технологій. Першим кроком у цьому аналізі є огляд різноманітності платформ, які існують на ринку. Цей огляд охоплює глобально відомі месенджери Telegram, Signal, Discord, WhatsApp, та Viber.

Дослідження цих різноманітних месенджерів є ключовим для ідентифікації основних тенденцій та факторів, які впливають на конкурентоспроможність і популярність окремих платформ серед користувачів. Розуміння цих аспектів допомагає виявити, як зміни в технологіях та користувацьких вподобаннях впливають на вибір та використання месенджерів у повсякденному житті.

Почнемо з огляду популярних месенджерів, кожен з яких має свої унікальні особливості та призначення для різних аудиторій.

Telegram є всім відомим багатоплатформовим месенджером, який вирізняється своєю безкоштовністю, високою швидкістю роботи та акцентом на безпеці даних. Підтримує наскрізне шифрування для всіх типів комунікації, включаючи текстові та голосові повідомлення, файли, відеодзвінки та голосові чати. Telegram пропонує широкий спектр додаткових функцій: канали, боти, публічні трансляції, опитування, групи до 200 000 користувачів, API для

розробників. Доступний на всіх платформах, включаючи мобільні телефони, планшети та комп'ютери [27].

Signal, безкоштовний месенджер з відкритим кодом, концентрується на конфіденційності та безпеці користувачів. Він забезпечує наскрізне шифрування для всіх видів комунікації, мінімізуючи ризики відстеження та зловживання інформацією, не збираючи метадані про своїх користувачів. Розроблений командою експертів з кібербезпеки, очолюваною Едвардом Сноуденом, це свідчить про високий рівень захисту [25].

Discord служить безкоштовною платформою для спілкування, яку активно використовують геймери та різноманітні спільноти для онлайн-взаємодії. Discord інтегрує функції чату, голосового зв'язку та форумів, пропонуючи систему серверів для організації спільнот і доступність на всіх основних платформах [14].

WhatsApp, який належить компанії Facebook, є одним з найпопулярніших безкоштовних месенджерів з багатомільйонною аудиторією по всьому світу. Він підтримує наскрізне шифрування для всіх видів комунікації, що забезпечує високий рівень безпеки персональної інформації та приватності користувачів. WhatsApp інтегрується з іншими сервісами Facebook, що дозволяє користувачам легко ділитися контентом та спілкуватися між різними платформами [32].

Viber, що належить компанії Rakuten, також підтримує наскрізне шифрування для всіх видів комунікації, включаючи текстові повідомлення, голосові та відеодзвінки. Viber відрізняється простотою використання та пропонує широкий спектр додаткових функцій, таких як стікери, публічні акаунти, групові чати та можливість здійснення міжнародних дзвінків. Доступний на всіх основних платформах, Viber є популярним вибором для користувачів, які шукають зручний спосіб для спілкування та обміну контентом [31].

Кожен з цих месенджерів має свої унікальні особливості та призначений для задоволення різних потреб користувачів. Від швидкості та безпеки

Telegram до приватності, яку надає Signal; від ігрової та спільнотної інтеграції Discord до широкої популярності WhatsApp і простоти використання Viber, кожен месенджер пропонує унікальний набір функцій для спілкування в сучасному цифровому світі. Вибір конкретного месенджера залежить від особистих переваг, потреб у безпеці та конфіденційності, а також від специфічних вимог до спілкування та обміну даними.

Для аналізу популярності згаданих месенджерів в Україні, ми використовуємо дані Google Trends, які є широко відомим інструментом для збору первинних даних. Значення популярності пошукових запитів в Google часто використовуються як непрямий спосіб оцінки інтересу до певних подій або тем. Відповідно до дослідження [12], Google Trends надає валідні дані, де показники коливаються від 0 до 100, іноді вказуючи на значення менші за одиницю, що відображає мінімальний, але відчутний інтерес до теми в аналізованій період. Наше дослідження охоплює період з січня 2014 року по березень 2024 року, включаючи 122 щомісячні спостереження [12].

На рисунку 1.1 представлено динаміку інтересу до месенджерів Telegram, Viber, WhatsApp, Discord, та Signal в Україні, використовуючи дані Google Trends. Для візуалізації даних було обрано лінійні графіки з різнокольоровим зображенням ліній, що спрощує процес ідентифікації тенденцій для кожного з месенджерів окремо. Кожна лінія на графіку відображає місячний інтерес до відповідного месенджера згідно з індексом пошукових запитів Google, де більші значення вказують на вищий рівень інтересу.

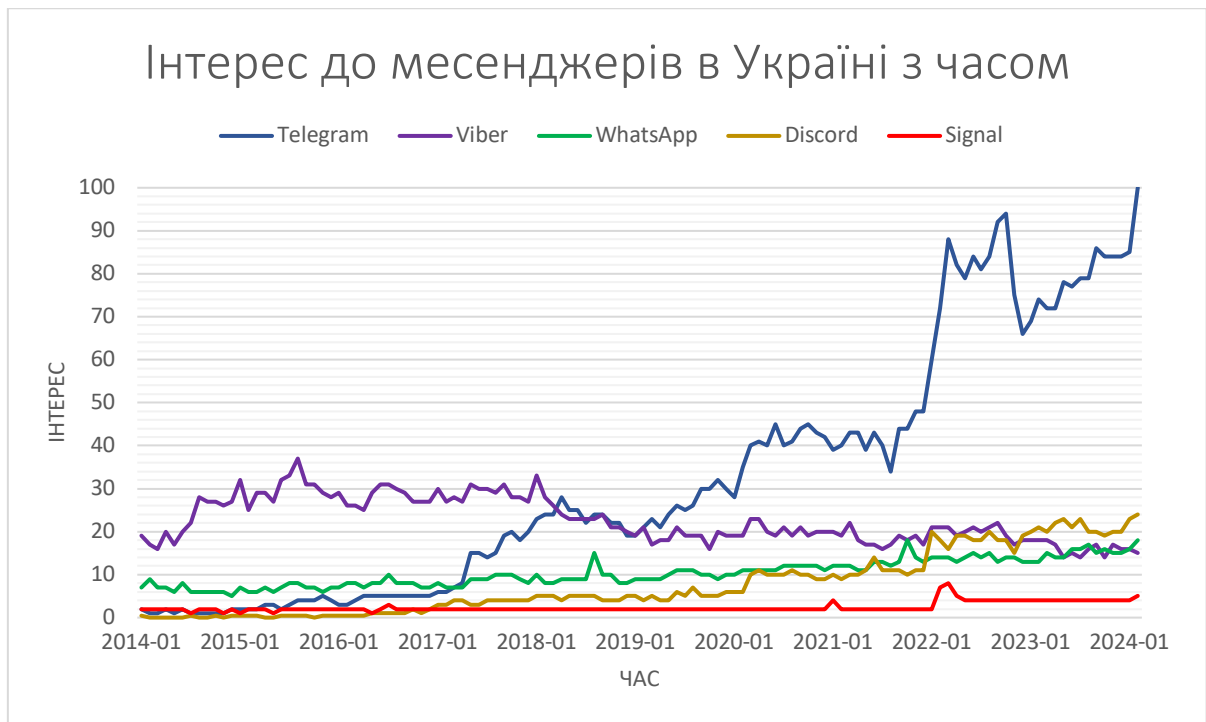


Рисунок 1.1 – Інтерес до месенджерів в Україні з часом

Аналіз графіку виявляє кілька ключових спостережень. По-перше, відзначається зростаючий тренд популярності месенджера Telegram, що може бути пов'язано з його функціональними можливостями. По-друге, інтерес до Viber має тенденцію до зниження, особливо після піку у попередні роки, що може свідчити про зміну уподобань користувачів. WhatsApp, незважаючи на коливання, зберігає стабільний інтерес серед користувачів, що підкреслює його постійну популярність. Discord, хоч і використовується менш широко в порівнянні з іншими месенджерами, демонструє помітне зростання інтересу, особливо серед геймерської спільноти. Нарешті, Signal, який наголошує на приватності, показує менший, але стабільний рівень інтересу.

Динамічний характер інтересу до месенджерів в Україні, який можна побачити на графіку, свідчить про зміни в популярності цих сервісів. Такі коливання можуть бути обумовлені різними факторами, включаючи нові функції, зміни у політиці конфіденційності, загальні тенденції в технологіях та суспільстві. Ці спостереження надають цінну інформацію для розуміння

поведінки споживачів та впливу технологічних інновацій на вибір комунікаційних інструментів.

1.3 Технічне завдання до розробки модуля взаємодії клієнтів та менеджерів

На основі зростаючого тренду популярності месенджера Telegram основним завданням кваліфікаційної роботи є розробка чат-боту в зазначеній мережі, з урахуванням специфіки взаємодії користувачів та операторами служби підтримки.

Вимога зосереджена на оптимізації взаємодії з клієнтами та на підвищенні ефективності роботи служби підтримки. Процес розробки охоплює планування, яке передбачає визначення вимог, вибір інструментів та мови програмування. Далі йде етап кодування бота, створення бази даних, а також розробка інтерфейсів для користувачів та адміністраторів. Таким чином, для створення чат-боту нам потрібно:

- платформа Telegram;
- мова програмування Python із бібліотекою aiogram яка є повністю асинхронний фреймворк для розробки чат-ботів.
- база даних SQLite для зберігання інформації про зареєстрованих користувачів, графік роботи та список днів.

На початку роботи з чат-ботом, користувача повинно зустрічати привітальне повідомлення з пропозицією зареєструватись. Для реєстрацію потрібно введення даних ПІП та адресу електронної пошти. Зазначимо, що електронна пошта має будити унікальною і не повторюватись.

Наступним етапом, після реєстрації нас зустрічає інтерфейс в залежності від рівня доступу (користувач, адміністратор).

Кнопки навігації інтерфейсу користувача:

- задати питання, починає пошук вільних операторів та живий чат з оператором;

- графік роботи дає можливість переглянути роботи дні та часи компанії;

- завершити діалог, з'являється лише тоді, коли почалась жива розмова між користувачем та операторами служби підтримки.

Кнопки навігації інтерфейсу адміністратора:

- почати працювати / завершити робочий день, динамічна кнопка яка залежить від того, чи працює зараз оператор;

- призначити адміністратора надає можливість змінювати рівень доступу будь-кому із списку зареєстрованих користувачів;

- змінити графік роботи дає можливість додавати та редагувати робочі часи для кожного дня тижня, також призначати дні вихідними;

- бан / разбан користувача надає можливість заблокувати або розблокувати будь-кого із списку зареєстрованих користувачів;

- завершити розмову, з'являється лише тоді, коли почалась жива розмова між користувачем та операторами служби підтримки.

Після завершення розробки бота, йде розгортання бота, налаштування взаємодії з базою даних, перевірка його функціональності, а також моніторинг його роботи. На цих етапах також здійснюється внесення необхідних оновлень та покращень, а також виявлення та усунення будь-яких помилок.

Цей бот може бути удосконалений у майбутньому шляхом додавання додаткового функціоналу. Представлений сервіс може виявитися корисним для середнього та малого бізнесу, що прагне покращити обслуговування клієнтів та оптимізувати роботу своїх служб підтримки.

2 РОЗРОБКА МОДУЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ КЛІЄНТІВ ТА МЕНЕДЖЕРІВ

2.1 Стек технологій які використовувався при розробці модуля автоматизація взаємодії клієнтів та менеджерів

В епоху постійних інновацій, ІТ-технології стають ключовими для бізнес-середовища, дозволяючи компаніям ефективно використовувати різноманітні програми для покращення якості роботи. Розвиток світу та змінювані потреби ринку вимагають від бізнесів адаптації та інтеграції новітніх технологій у всі сфери діяльності. Особливо це важливо для середнього та малого бізнесу, де автоматизація взаємодії з клієнтами через сучасні канали, такі як месенджери, може значно підвищити ефективність та рівень задоволення клієнтів.

Розробка Telegram-бота для автоматизації взаємодії між клієнтами та менеджерами відіграє важливу роль у забезпеченні конкурентних переваг для бізнесу. Використання Telegram надає переваги не тільки через його популярність та доступність.

У даній роботі обрано Python як основну мову програмування з огляду на її високий рівень, інтерпретованість, об'єктно-орієнтованість та широку популярність, яка обумовлена простотою читання коду та гнучкістю. Створена Гвідо ван Россумом у 1991 році, Python став значним інструментом у таких сферах, як веб-розробка, штучний інтелект, автоматизація тощо [23]. Інтерпретована природа Python дозволяє виконувати програми без необхідності попередньої компіляції, спрощуючи тим самим процес розробки та тестування. Підтримка різноманітних стилів програмування, включаючи процедурний, об'єктно-орієнтований та функціональний підходи, розширює можливості розробників у виборі оптимального способу рішення поставлених задач.

Однією з ключових переваг Python є висока читабельність його коду, обумовлена чітким синтаксисом. Ця характеристика робить мову доступною навіть для початківців, сприяючи ефективній розробці проектів. Крім того, зрозумілість коду на Python значно полегшує співпрацю між розробниками, що є важливим аспектом у командній роботі та при великих програмних проектах. Основні характеристики мови, такі як динамічна типізація, автоматичне керування пам'яттю, наявність великої стандартної бібліотеки та розвинена спільнота розробників, роблять Python вельми привабливим для широкого спектра застосувань. Додатково, Python вирізняється наявністю численних сторонніх бібліотек і фреймворків, що полегшують розробку різноманітного програмного забезпечення.

Об'єктно-орієнтована природа Python, при якій усе в мові є об'єктом і весь код написаний за допомогою класів та об'єктів, сприяє створенню модульного та легко розширювального коду, що полегшує управління складністю програм. Python також забезпечує високу ефективність та підтримку для різноманітних операцій завдяки вбудованим структурам даних та підтримці роботи з файлами, мережами, регулярними виразами. Для веб-розробки використовуються фреймворки, такі як Django і Flask, де Django пропонує повноцінний стек для розробки веб-додатків, а Flask виступає як більш легкий та гнучкий варіант.

Python відіграє важливу роль у сфері науки про дані та штучного інтелекту, де його багата екосистема бібліотек, включаючи NumPy, Pandas, Matplotlib, SciPy для аналізу даних, а також TensorFlow та PyTorch для розробки нейронних мереж, є незамінною [21]. Ці інструменти перетворюють Python на основний інструмент для машинного навчання та розробки інтелектуальних систем. Завдяки простоті та гнучкості Python, він користується популярністю не лише серед новачків, а й серед досвідчених розробників. Бібліотеки та фреймворки, які постійно з'являються, дозволяють застосовувати Python у різних галузях, зокрема в веб-розробці, науці про дані, штучному інтелекті.

Динаміка популярності мов програмування, відображена на Рисунку 2.1, підтверджує значний інтерес до Python, який не лише вперше очолив індекс Tiobe у жовтні 2021 року, а й продовжує бути лідером станом на березень 2024 року, випереджаючи C та Java [24].

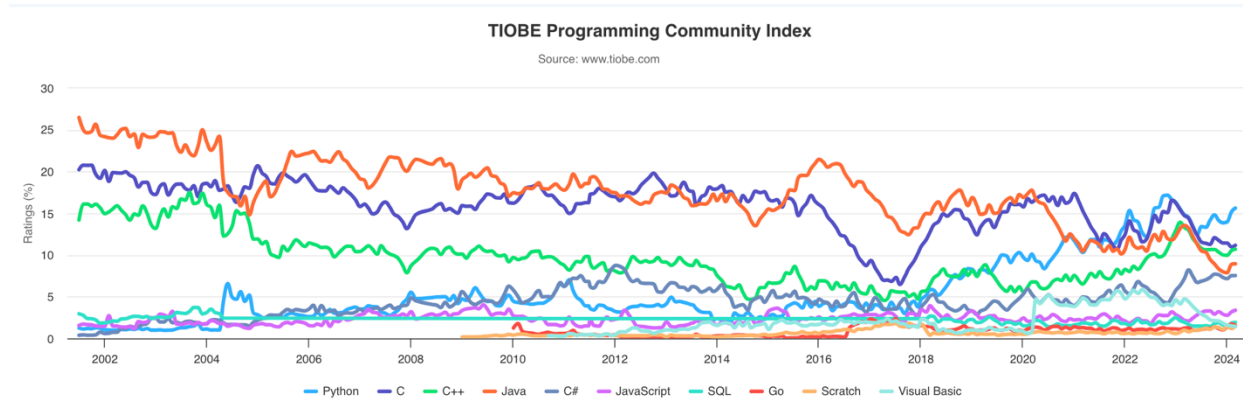


Рисунок 2.1 – Інтерес до мов програмування в світі з часом [29]

Індекс Tiobe ранжує мови на основі кількості інженерів, курсів і сторонніх постачальників, з використанням ресурсів Google, Amazon, Вікіпедія та Bing. Аналогічно, індекс PYPL оцінює популярність мов за частотою пошуків їх навчальних матеріалів у Google. За даними PYPL на березень 2024 року, Python займає лідируючу позицію з часткою ринку 15,63% [29].

Зміни в рейтингу мов програмування за індексом Tiobe відображають не тільки зміну уподобань у галузі, а й адаптацію до новітніх технічних можливостей та вимог ринку. Python зберігає позиції лідера, демонструючи свою універсальність, що відображає ключові напрямки розвитку у сфері програмного забезпечення.

Для розробки Telegram-бота було обрано середовище розробки PyCharm від JetBrains, яке користується великою популярністю серед розробників Python. PyCharm вирізняється своїм комплексним набором інструментів, які підвищують продуктивність розробки: від оптимізації процесу написання коду до ефективного виявлення помилок, автоматичного доповнення коду та

управління версіями проекту. Ці особливості значно спрощують і прискорюють процес створення чат-ботів, зокрема Telegram-бота [22].

Aiogram є асинхронним фреймворком для створення Telegram-ботів на Python, який дозволяє розробляти високопродуктивні асинхронні боти, здатні одночасно обробляти велику кількість запитів. Цей фреймворк, який використовує API Telegram Bot, сприяє швидкому реагуванню ботів на запити користувачів, забезпечуючи при цьому гнучкість і масштабованість бізнес-проектів. Основними перевагами aiogram є його простота використання та асинхронність, підтримувана стандартною бібліотекою Python asyncio та асинхронним веб-фреймворком aiohttp [7; 9; 10]. Важливою особливістю aiogram є його здатність ефективно працювати з великими навантаженнями при мінімальному використанні ресурсів, а також наявність вбудованого кінцевого автомата для розробки більш складних чат-ботів і підтримка вебхуків Telegram API для відповіді на оновлення [28].

Технологія Aiogram вважається найкращою для розробки ботів на Python, завдяки своїй простоті і повній асинхронності, що дозволяє створювати високопродуктивні рішення з мінімальними вимогами до ресурсів [8]. Розробники високо цінують aiogram за здатність витримувати великі навантаження при мінімальному споживанні ресурсів, що робить його оптимальним вибором для розробки чат-ботів [8].

Наступним кроком у процесі створення бота є розробка структури бази даних. Цей етап є критично важливим для зберігання та організації інформації, необхідної для функціонування бота.

Для бази даних цього Telegram-бота обрано SQLite, пакет для якого вже інтегровано у Python. SQLite у контексті Python пропонує обмежений набір типів даних порівняно з іншими реалізаціями SQL, що, з одного боку, ставить певні обмеження, але з іншого – спрощує багато аспектів роботи з базою даних [26].

База даних є систематизованим набором даних з певними характеристиками та зв'язками. Інтеграція даних в єдину базу дозволяє

ефективно організувати інформацію. У веб-розробці часто використовуються реляційні бази даних, де інформація організована у вигляді таблиць та зв'язків між ними. Мова SQL застосовується для взаємодії з базою даних, забезпечуючи можливості для читання, оновлення, видалення та структурування інформації [19].

Система управління базами даних (СУБД) виконує ключові завдання: управління даними, забезпечення їх цілісності, резервне копіювання та відновлення, мультикористувацький доступ, а також захист інформації. SQLite як реляційна СУБД, реалізована на мові C, інтегрується безпосередньо в програму, пропонуючи ряд переваг: повну підтримку стандарту SQL-92, сумісність з багатьма мовами програмування, включно з Node.js, зберігання даних у міжплатформенному файлі з максимальним розміром до 281 ТБ, простоту запуску без потреби в первинній конфігурації, відсутність зовнішніх залежностей із компактним кодом до 350 КБ [26].

SQLite, як легковагова система управління базами даних, не потребує встановлення окремого серверу. Це робить її оптимальним вибором для зберігання даних про користувачів, їх запити та взаємодію з чат-ботом, оскільки вона забезпечує швидкий доступ до інформації завдяки підтримці SQL і надійному зберіганню даних.

На сервері Telegram вхідні оновлення зберігаються до моменту обробки, проте не більше 24 годин. Для отримання оновлень використовуються дві ключові технології: Long Polling та Webhook (рис. 2.2). При використанні Long Polling бот періодично опитує сервери Telegram, встановлюючи короткотривале з'єднання для негайної передачі всіх оновлень. Цей метод, хоч і простий, але не вважається найбільш надійним [28; 11; 18].



Рисунок 2.2 – Механізм Long Polling та Webhook [18]

Webhook працює за іншим принципом. Суть його роботи полягає в тому, що Telegram автоматично надсилає сповіщення, коли повідомлення надходить у чат. Це усуває потребу в постійному опитуванні сервера, що значно знижує ризик помилок. Однак, використання Webhook вимагає наявності на обладнанні користувача повноцінного веб-сервера, який має відповідати наступним вимогам [28; 11; 18]:

- підтримка IPv4 (IPv6 наразі не підтримується для Webhook);
- прийом вхідних POST-повідомлень з підмереж 149.154.160.0/20 та 91.108.4.0/22 на порт 443, 80, 88, або 8443;
- обробка HTTPS-трафіку з підтримкою TLS1.2 або вище;
- наявність підтримуваного, перевіреного або самопідписаного сертифікату;
- використання CN або SAN, що відповідає домену, зазначеному під час налаштування;
- надання всіх проміжних сертифікатів для завершення ланцюжка перевірки.

Ці вимоги мають бути виконані для ефективної роботи з Webhook. Також рекомендується регулярно перевіряти офіційну документацію Telegram на предмет можливих змін у діапазоні IP-адрес, щоб уникнути проблем з підключенням [28].

2.2 Описання структури і функціоналу модуля автоматизація взаємодії клієнтів та менеджерів

На початковому етапі розробки Telegram-бота була створена база даних «database.db». Основний код взаємодії з базою даних розміщено у Додатку Б. База даних для Telegram-бота включає три таблиці, як показано на рисунку 2.3.

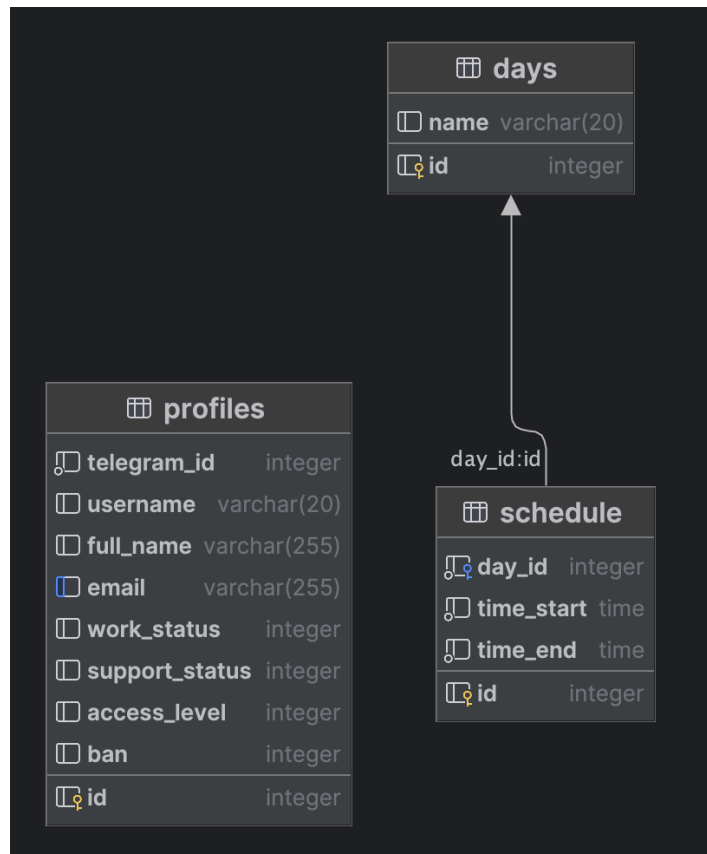


Рисунок 2.3 – Структура бази даних

Таблиця «profiles» містить кілька елементів, призначених для зберігання інформації про користувачів:

- id, первинний ключ, що унікально ідентифікує кожен профіль
- telegram_id, унікальний ідентифікатор кожного користувача в Telegram;
- username, ім'я користувача в Telegram;
- full_name, повне ім'я користувача (вказується при реєстрації);
- email, електронна адреса користувача (вказується при реєстрації);

- `work_status`, `support_status`, `access_level`, `ban`, різні статуси для управління привілеями та обмеженнями користувачів.

Таблиця «`schedule`» включає деталі, пов'язані з управлінням часом:

- `id`, первинний ключ, що унікально ідентифікує кожен запис у розкладі.
- `day_id`, зовнішній ключ, який зв'язується з таблицею «`days`», вказуючи на день для розкладу;
- `time_start`, час початку роботи компанії;
- `time_end`, час завершення роботи компанії;

Таблиця «`days`» є простою структурою, яка містить інформацію про дні:

- `id`, первинний ключ, що унікально ідентифікує кожен день, а також служить зв'язком для таблиці «`schedule`»;
- `name`, назва дня тижня.

У Telegram існує спеціалізований бот, відомий як BotFather, який дозволяє користувачам створювати власних ботів із заданими параметрами та функціоналом, що є основним завданням у процесі створення бота. BotFather допомагає присвоїти ім'я боту та налаштувати список команд за запитом користувача. Процес створення бота починається з відправлення команди «`/newbot`» до BotFather, після чого необхідно слідувати інструкціям (рис. 2.4).

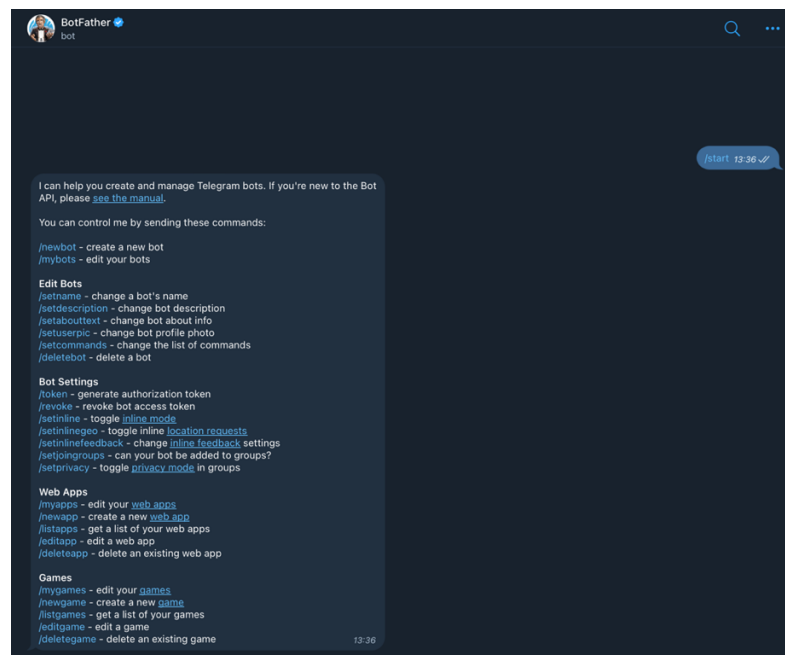


Рисунок 2.4 – Скріншот BotFather

Під час першої взаємодії з ботом користувачам пропонується зареєструватися, вводячи своє повне ім'я та електронну пошту. Після реєстрації користувачам стають доступні функції, такі як перегляд графіка роботи, отримання інформації про бота та можливість задати питання менеджеру (рис. 2.5).

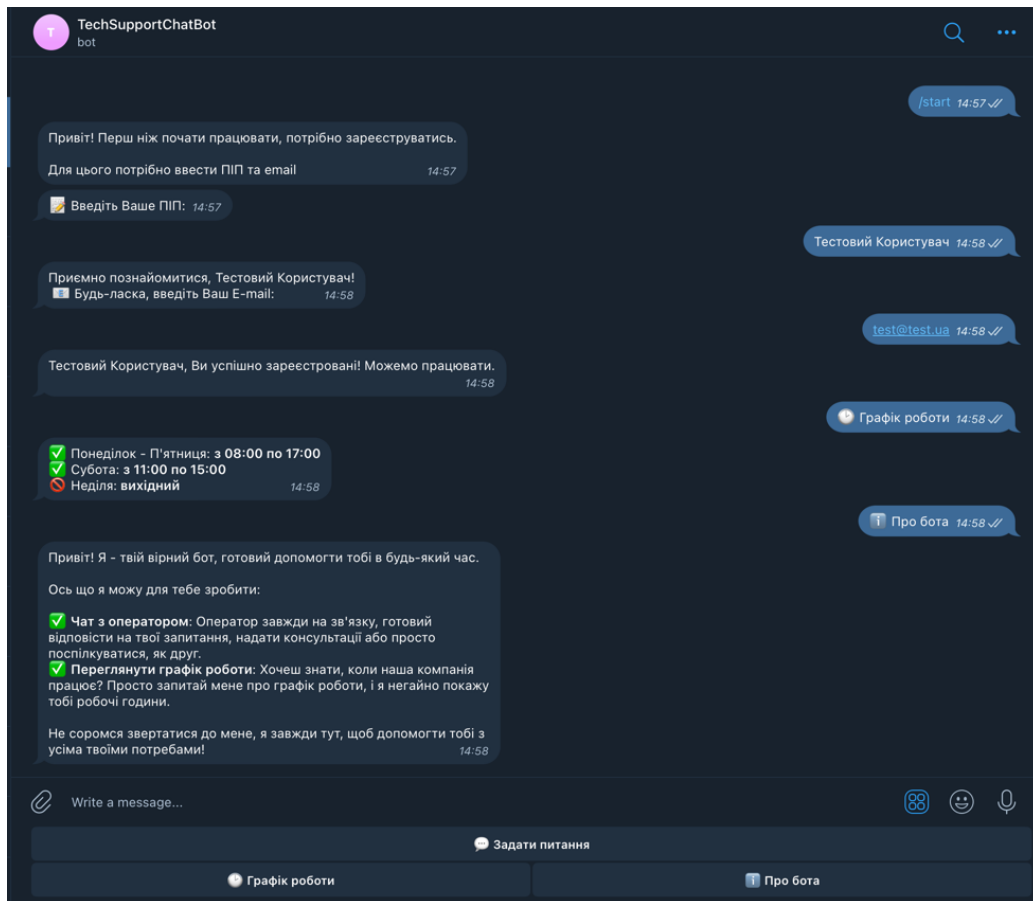



Рисунок 2.5 – Скріншот розробленого бота зі сторони користувача

Коли користувач натискає кнопку « Задати питання», бот спробує знайти вільного менеджера. У випадку, якщо вільний менеджер не буде знайдений, користувач отримає повідомлення про те, що всі оператори зайняті. Якщо ж менеджер доступний, йому надсилається запит на спілкування з користувачем, де вказано ім'я користувача та його Telegram ID, а користувач переходить у режим очікування взаємодії (рис. 2.6).

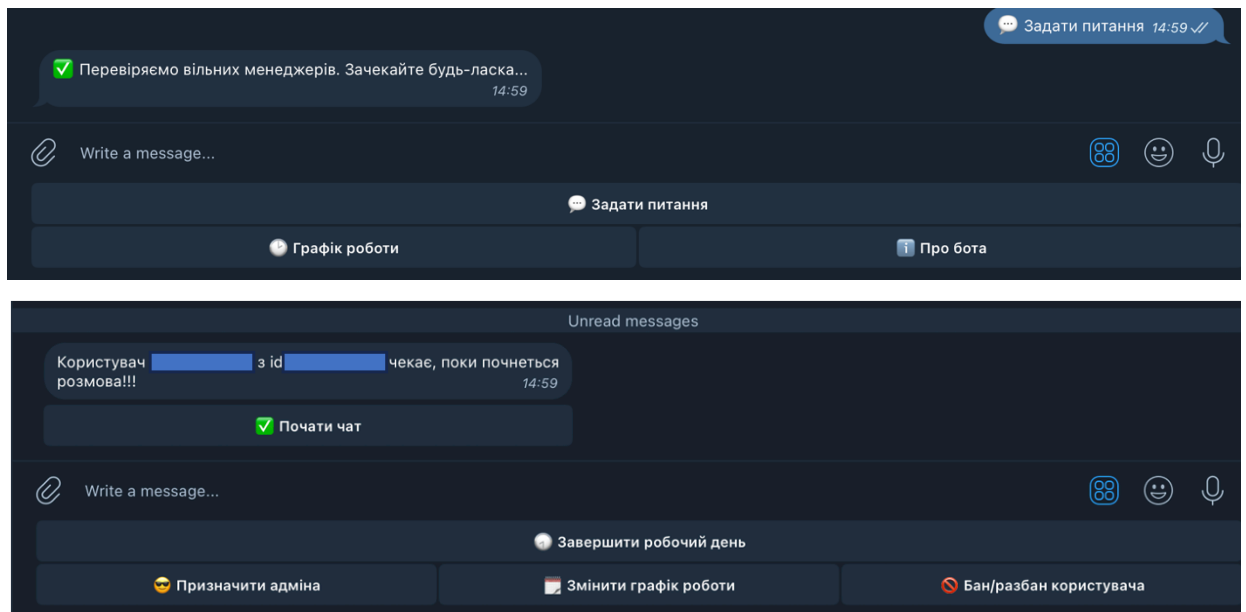


Рисунок 2.6 – Початок діалогу

Коли менеджер підтверджує початок чату, користувач отримує повідомлення про підключення до менеджера, що відкриває можливість для обох сторін обмінюватися повідомленнями, файлами, фотографіями, та відео в режимі реального часу, що можна побачити на рисунках 2.7 та 2.8. Для запобігання одночасному спілкуванню менеджера з кількома користувачами, статус менеджера змінюється на «в процесі спілкування». Така система гарантує, що кожен діалог з менеджером є індивідуальним і не переривається іншими запитами.

Як користувач, так і менеджер мають право в будь-який момент завершити розмову, при цьому обидві сторони отримують повідомлення про завершення діалогу. Після завершення діалогу, статус менеджера автоматично змінюється на «доступний для спілкування», що робить його готовим до нових звернень користувачів.

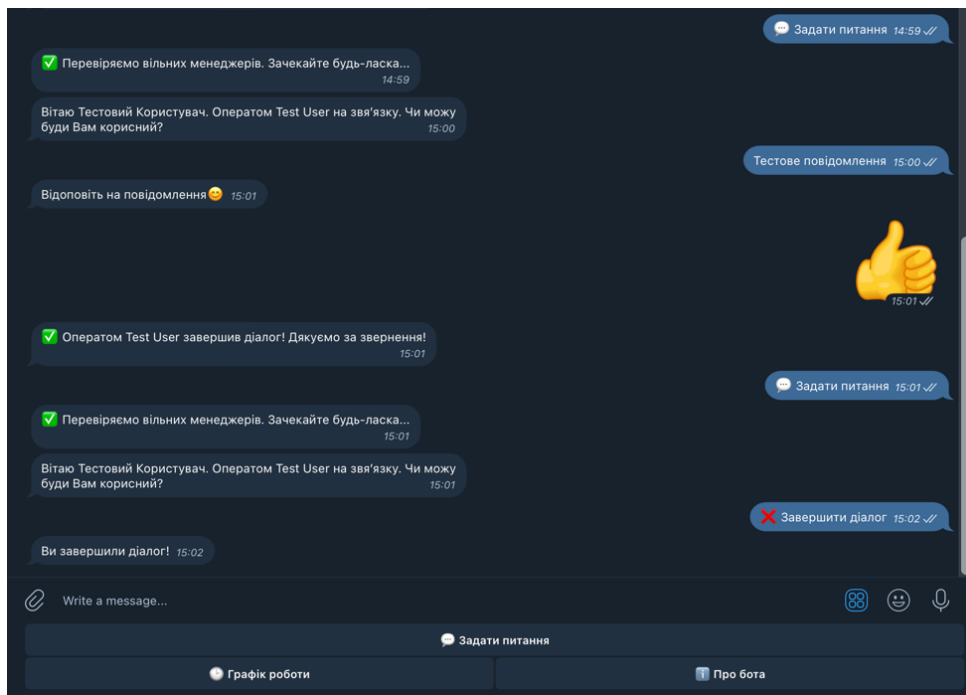


Рисунок 2.7 – Скріншот діалог

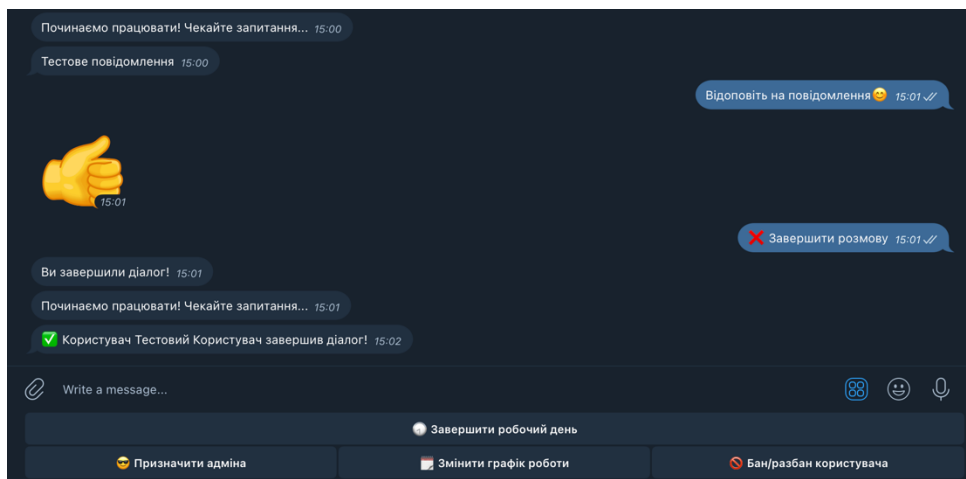





Рисунок 2.8 – Скріншот діалог

Функції менеджера обмежені безпосереднім спілкуванням з користувачами та можливістю змінювати свій робочий статус. Менеджер може переводити свій статус між « Почати працювати» та « Завершити робочий день». Залежно від обраного статусу, менеджер отримує або не отримує запити на спілкування від користувачів. Коли менеджер вибирає статус « Завершити робочий день», йому більше не надходять нові

запити на діалог, що дозволяє ефективно управляти часом роботи та взаємодією з користувачами (рис. 2.9).

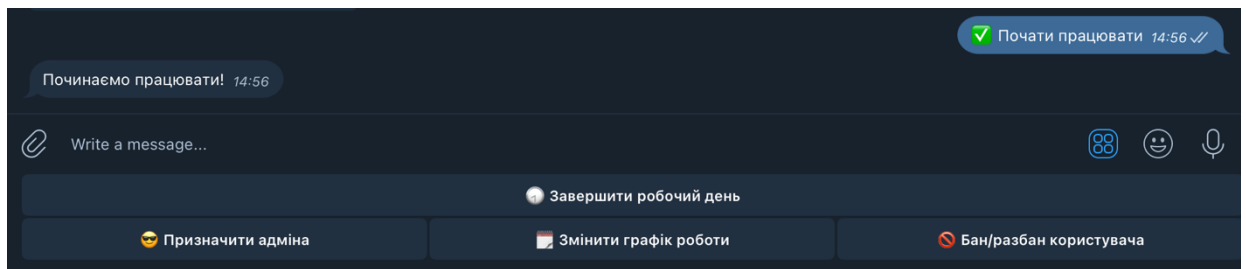


Рисунок 2.9 – Скріншот зміни статусу

Адміністратор відіграє ключову роль у керуванні доступом та налаштуваннями бота, забезпечуючи його ефективну адаптацію до потреб бізнесу та вимог клієнтів. Повноважень адміністратора охоплює кілька критичних аспектів.

Управління правами користувачів, адміністратор має можливість змінювати права та привілеї користувачів. Це дозволяє гнучко налаштовувати рівні доступу до бота, відповідно до ролі кожного користувача у бізнес-процесах (рис. 2.10).

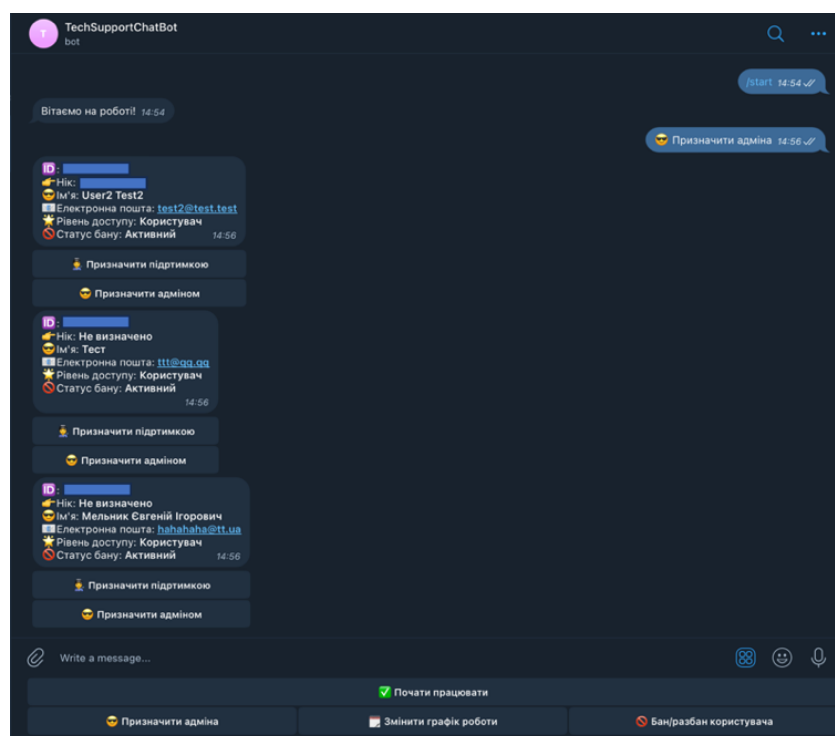


Рисунок 2.10 – Змінювати права та привілеї користувачів

Блокування та розблокування користувачів, адміністратор може управляти доступом користувачів шляхом їх блокування або розблокування. Таке управління є важливим інструментом для забезпечення безпеки та відповідності політикам компанії (рис 2.11);

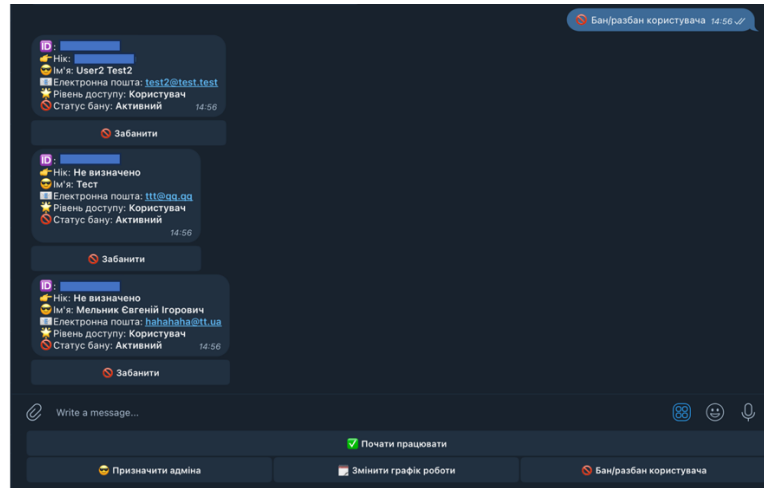


Рисунок 2.11 – Блокувати та розблокувати користувачів

Редагування графіку роботи, відповідальність адміністратора також включає можливість редагування графіку роботи команди, що дозволяє оптимізувати робочі процеси та забезпечити найвищу ефективність взаємодії з клієнтами (рис. 2.12).

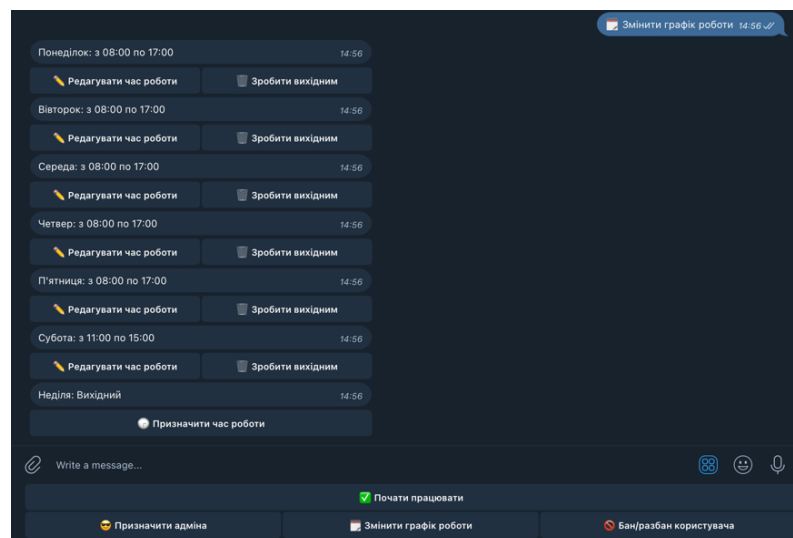


Рисунок 2.12 – Редагувати графік роботи

Завдяки цим можливостям, адміністратор сприяє створенню ефективного середовища для роботи бота, що включає у себе як технічні налаштування, так і управління людськими ресурсами. Детальніше про взаємодію між інтерфейсами користувача, менеджера та адміністратора можна дізнатись в Додатку Г, де представлена загальна блок-схема.

2.3 Оцінка очікуваного ефекту від впровадження модуля автоматизація взаємодії клієнтів та менеджерів

Автоматизація взаємодії між клієнтами та менеджерами представляє собою важливий елемент підвищення ефективності бізнес-операцій, який вимагає детального аналізу економічної доцільності. Економічна ефективність, яка характеризується співвідношенням отриманих результатів до затрачених ресурсів, є ключовим показником для оцінки вартості впровадження автоматизації. Часто підприємці поспішають впровадити автоматизацію, не враховуючи повного обсягу її потенційної користі, що включає можливість зекономити кошти та інші ресурси.

Залежно від специфіки сектору, типу бізнесу та характеристик продукції, підходи до оцінки економічної ефективності можуть відрізнятися. Однак загальним для них є те, що автоматизація спрямована на мінімізацію витрат, а не на пряме створення прибутку. Основні переваги автоматизації полягають у зменшенні використання ресурсів завдяки оптимізації управління та зниженню витрат на оплату праці [3; 4; 1].

Одним з найбільш доступних інструментів для автоматизації взаємодії з клієнтами є розробка Telegram-бот, який вирізняється своєю безкоштовністю та відкритістю для широкого кола користувачів. Важливо порівняти це з платними альтернативами на ринку, які пропонують подібні послуги, але за підписку. Аналіз вартості платних сервісів показує, що ціни можуть суттєво коливатися в залежності від обсягу запитів, з вартістю від 702 грн на місяць за

менше ніж 1000 підключень до 1326 грн за обсяг від 5000 до 20000 підключень [6].

Розглядаючи ціни на розробку чат-ботів (рис. 2.13), стає зрозумілим, що впровадження власного Telegram-бота може бути значно вигіднішим для середнього та малого бізнесу, якому важливо швидко реагувати на запити клієнтів. Також варто відзначити, що розробка такого бота не вимагає великих витрат або особливих ресурсів з боку розробників, що робить його доступним варіантом для широкого кола бізнесів.

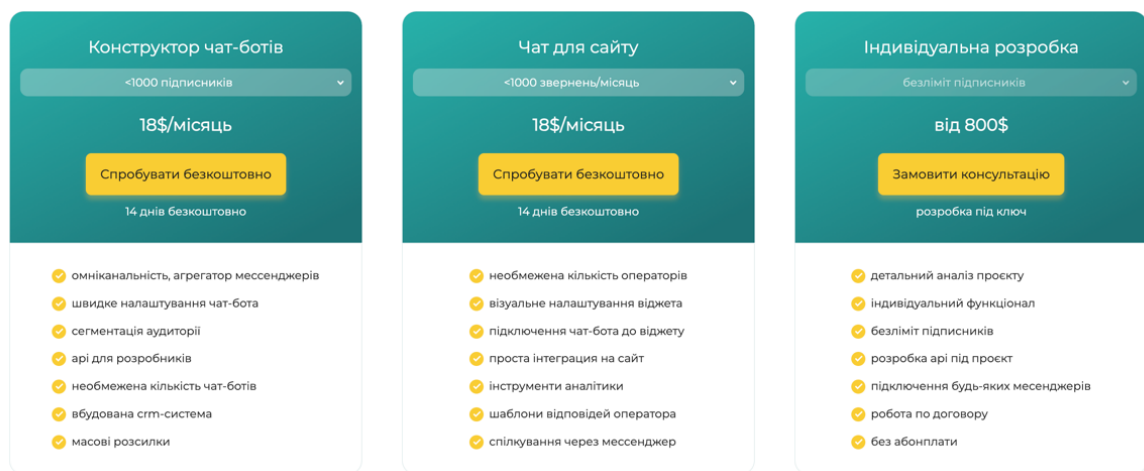


Рисунок 2.13 – Цінова політика сайту Gerabot [6]

Капітальні витрати на розробку Telegram-бота можуть бути розраховані за формулою, яка включає витрати на проектування, програмування, впровадження, налагодження, програмне та апаратне забезпечення [3; 4; 1].

$$K = K_1 + K_2 + K_3 + K_4 + K_5 + K_6 \quad (2.1)$$

де $K_1 - K_4$ вказують на заробітну плату програмістів, що охоплює різні аспекти роботи над системою: включаючи витрати на розробку системи, програмування, її впровадження, а також витрати, пов'язані з її налаштуванням та перевіркою;

K_4 – витрати, що виникають під час налаштування та тестування системи;

K_5 – витрати на програмне забезпечення;

K_6 – витрати на обладнання;

При оцінці витрат на розробку та впровадження програмного забезпечення, ключовим компонентом є аналіз витрат на працю програмістів, а також врахування витрат на інші необхідні ресурси. Загальна кількість годин робочого часу, потрібного для розробки, оцінюється у 80 годин. Виходячи з середньої місячної заробітної плати програміста у розмірі 101 400 грн [15], враховуючи 40 годинний робочий тиждень, годинна ставка складає 633,75 грн. До витрат також входить оплата хостингу, яка становить 2340 грн на рік за тарифом Nanode 1 GB [13]. Використання інструменту PyCharm Community, який є безкоштовним, дозволяє знизити загальні капітальні витрати [22].

Отже, розрахунок загальних капітальних витрат виглядає наступним чином:

$$K = 80 \cdot 633,75 + 0 + 2340 = 53\,040 \text{ (грн)}$$

Платні сервіси часто надають широкий спектр функціональних можливостей, пропонуючи розширений набір інструментів для бізнесу. Однак, варто врахувати, що не всі ці можливості можуть бути активно використані, що робить частину функціоналу зайвою. У таких випадках, безкоштовні альтернативи, наприклад, розробка власних Telegram-ботів, стають особливо привабливими. Вони дозволяють суттєво скоротити витрати, при цьому забезпечуючи користувачам необхідний функціонал без зайвих переplat за невикористані сервіси. Заощадження, що виникають в результаті такого підходу, можуть становити до 53 040 грн, забезпечуючи ефективне рішення, яке задовольняє потреби користувачів у зручності та доступності, дозволяючи використовувати лише потрібний функціонал.

ВИСНОВКИ

В умовах сучасної економіки, яка характеризується швидкими темпами цифровізації, бізнес-моделі та вимоги споживачів постійно змінюються, автоматизація взаємодії між клієнтами та менеджерами набуває стратегічної важливості. Це стає не лише інструментом оптимізації, але й ключовим елементом стратегії, особливо для малого та середнього бізнесу, що стикається з обмеженими ресурсами та високим рівнем конкуренції. Ефективні рішення для підтримки високоякісного обслуговування стають критично важливими.

У межах проведеного дослідження було розроблено Telegram-бота, який спрощує процеси взаємодії з клієнтами та збільшує продуктивність служби підтримки, відкриваючи нові можливості для підвищення задоволеності клієнтів, скорочення часу реагування на запити та покращення загальної ефективності бізнесу.

Головна мета цієї кваліфікаційної роботи – розробка Telegram-бота для підтримки клієнтів, зі специфічним функціоналом для адміністраторів та користувачів, була успішно досягнута.

В рамках роботи було досягнуто наступних завдань:

- описано існуючі рішення для автоматизації взаємодії між клієнтами та службою підтримки;
- розроблено технічне завдання для створення Telegram-бота, враховуючи специфіку взаємодії користувачів та операторів служби підтримки;
- досліджено оптимальний стек технологій та бібліотек для розробки Telegram-бота;
- розроблено та протестовано Telegram-бот «TechSupportChatBot».

Таким чином, автоматизація взаємодій з клієнтами через Telegram-боти відіграє важливу роль у стратегії цифровізації бізнесу, дозволяючи підвищити якість обслуговування та ефективність бізнес-процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грек К. А. Автоматизований клієнтський сервіс трекінгу поштових відправлень : робота на здобуття кваліфікаційного ступеня магістра за спеціальністю 051 - економіка / К. А. Грек; наук. кер. К. Г. Гриценко. – Суми : Сумський державний університет, 2022. – 56 с.
2. Ковальов Б.Л., Павлик А.В., Федина С.М. Бізнес-комунікації: конспект лекцій для студ. спец. 051 "Економіка" освітньої програми "Економіка і бізнес" і 076 "Підприємництво, торгівля та біржова діяльність" освітньої програми "Підприємництво, торгівля та біржова діяльність" денної форми навчання. Суми: СумДУ, 2020. 113 с. [Електронний ресурс]. Електронне видання каф. економіки, підприємництва та бізнес-адміністрування ННІ ФЕМ ім. О.Балацького. Доступно: <https://essuir.sumdu.edu.ua/handle/123456789/77603>.
3. Методи оцінки ефективності інвестиційного проекту [Електронний ресурс]. Доступно: <https://buklib.net/books/35308/>
4. Півень А. В. Розробка модуля для автоматизованого аналізу фінансових звітів компаній з використанням машинного навчання : робота на здобуття кваліфікаційного ступеня магістра за спеціальністю 051 - економіка / А. В. Півень; наук. кер. В. В. Койбічук. – Суми : Сумський державний університет, 2023. – 74 с.
5. Рикова В. ТОП-10 месенджерів для повідомлень та дзвінків у 2024 р. [Електронний ресурс]. Доступно: <https://vlada-rykova.com/ua/top-messendzherov/>
6. Розробка та створення чат бота telegram, viber, fb, instagram – компанія Gerabot [Електронний ресурс]. Доступно: <https://gerabot.com/>
7. Створюємо Telegram бота на Python. Частина 1 [Електронний ресурс]. Режим доступу: <https://codeguida.com/post/410>
8. Aiogram Documentation [Електронний ресурс]. Доступно: <https://docs.aiogram.dev/en/latest/>

9. Async HTTP client/server for asyncio and Python [Електронний ресурс]. Режим доступу: <https://docs.aiohttp.org/en/stable/>
10. Asynchronous I/O [Електронний ресурс]. Режим доступу: <https://docs.python.org/3/library/asyncio.html>
11. Building A Simple Telegram Bot With AWS Lambda [Електронний ресурс]. Режим доступу: <https://phatdangx.medium.com/building-a-simple-telegram-bot-with-aws-lambda-c3143e596b3b>
12. Cebrián E., Domenech J. Is Google Trends a quality data source? "Applied Economics Letters", 2023, vol. 30, no. 6, p. 811–815. <https://doi.org/10.1080/13504851.2021.2023088>.
13. Cloud Computing Services [Електронний ресурс]. Доступно: <https://www.linode.com/pricing/>
14. Discord [Електронний ресурс]. Доступно: <https://discord.com/>
15. DOU: Спільнота програмістів [Електронний ресурс]. Доступно: <https://jobs.dou.ua/salaries/?period=2023-12&position=Middle%20SE&technology=Python>
16. Fraccastoro S., Gabrielsson M., Pullins E. B. The integrated use of social media, digital, and traditional communication tools in the B2B sales process of international SMEs. "International Business Review", 2021, vol. 30, no. 4, 101776. <https://doi.org/10.1016/j.ibusrev.2020.101776>.
17. Google Trends. [Електронний ресурс]. Доступно: <https://trends.google.com.ua/trends/explore?date=2014-01-01%202024-03-01&geo=UA&q=Telegram,Viber,WhatsApp,Discord,Signal&hl=uk>.
18. How we did it: Webhooks at Dev Summit 2019 [Електронний ресурс]. Режим доступу: <https://www.esri.com/arcgis-blog/products/arcgis-enterprise/administration/webhooks-dev-summit-2019/>
19. Khan W., Kumar T., Zhang C., Raj K., Roy A.M., Luo B. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review // Big Data and Cognitive Computing. – 2023.

- Vol. 7, No. 2. – Article 97. – Режим доступа:
<https://doi.org/10.3390/bdcc7020097>
20. Melović B., Jocović M., Dabić M., Vulić T. B., Dudic B. The impact of digital transformation and digital marketing on the brand promotion, positioning and electronic business in Montenegro. "Technology in Society", 2020, vol. 63, 101425. <https://doi.org/10.1016/j.techsoc.2020.101425>.
21. Packages: A Comprehensive Review // International Journal Papier Advance and Scientific Review. – 2024. – Vol. 5, No. 1. – P. 1-9. – Режим доступа до журн.: <https://doi.org/10.47667/ijpasr.v5i1.283>
22. PyCharm [Электронный ресурс]. Доступно: <https://www.jetbrains.com/pycharm/>
23. Python Documentation [Электронный ресурс]. Доступно: <https://docs.python.org/3/>
24. Python stretches lead in language popularity index [Электронный ресурс]. Доступно: <https://www.infoworld.com/article/3714284/python-stretches-lead-in-language-popularity-index.html>
25. Signal [Электронный ресурс]. Доступно: <https://signal.org/uk/>
26. SQLite Documentation [Электронный ресурс]. Доступно: <https://www.sqlite.org/docs.html>
27. Telegram [Электронный ресурс]. Доступно: <https://telegram.org/?setln=uk>
28. Telegram APIs [Электронный ресурс]. Режим доступа: <https://core.telegram.org/>
29. TIOBE Programming Community index [Электронный ресурс]. Доступно: <https://www.tiobe.com/tiobe-index/>
30. Verma A. K. et al. The Role of Communication for Business Growth: A Comprehensive Review. "World Journal of English Language", 2022, vol. 12, no. 3, p. 164. <https://doi.org/10.5430/wjel.v12n3p164>.
31. Viber [Электронный ресурс]. Доступно: <https://www.viber.com/ua/>
32. WhatsApp [Электронный ресурс]. Доступно: https://www.whatsapp.com/?lang=uk_UA

ДОДАТКИ

ДОДАТОК А

SUMMARY

Kostiushko I. O. Automation of Interaction between Clients and Managers in the Segment of Medium and Small Business – Bachelor's thesis. Sumy State University, Sumy, 2024.

This qualification work is dedicated to the development of an automated solution that simplifies interaction between clients and managers in small and medium-sized businesses through a Telegram bot. The research analyzes existing communication methods, identifies key requirements for the new system, and discusses the selection of technologies for development. The work includes designing the architecture and functionality of the chatbot, as well as assessing the potential benefits of implementing such a solution. In the development process, special attention is paid to choosing the technology stack, developing automation modules, and planning their integration with existing business processes. An overview of the main system requirements is provided, and the choice of development tools is justified, including the Python programming language and the aiogram framework.

Keywords automation, Telegram bot, customer interaction, communication, chatbot, Python, aiogram.

АНОТАЦІЯ

Костюшко І. О. Автоматизація взаємодії клієнтів та менеджерів у сегменті середнього та малого бізнесу – Кваліфікаційна робота бакалавра. Сумський державний університет, Суми, 2024 р.

Ця кваліфікаційна робота присвячена розробці автоматизованого рішення, що спрощує взаємодію між клієнтами та менеджерами в середньому та малому бізнесі через Telegram-бота. У дослідженні аналізуються наявні методи комунікації, визначаються ключові вимоги до нової системи та обговорюється вибір технологій для розробки. Робота включає проектування архітектури та функціональності чат-бота, а також оцінює потенційні переваги впровадження такого рішення. В процесі розробки особлива увага приділяється вибору стеку технологій, розробці модуля автоматизації та плануванню його інтеграції з існуючими бізнес-процесами. Подано огляд основних вимог до системи та обґрунтовано вибір інструментів для розробки, включаючи мову програмування Python та фреймворк aiogram.

Ключові слова: автоматизація, Telegram-бот, взаємодія з клієнтами, комунікація, чат-бот, Python, aiogram.

ДОДАТОК Б

Структура проєкту

Таблиця Б.1 – Опис структури проєкту

Назва файлу	Опис
bot.py	Головний файл проєкту, що містить основну логіку та обробники для бота
config.py	Файл конфігурації, який зберігає стандартні повідомлення, Telegram ID розробника та токен бота
loader.py	Відповідає за імпортування необхідних бібліотек, ініціалізацію бота та налаштування Dispatcher для обробки оновлень
keyboards.py	Визначає інтерактивні клавіатури бота
.env	Конфігураційний файл, що містить чутливі дані, такі як токен Telegram (TG_TOKEN).
database.py	Налаштування з'єднання з базою даних та включає функції для маніпуляції з даними бази даних
admin.py	Містить функціональні можливості, доступні адміністраторам бота
support.py	Реалізує функціонал менеджера підтримки, сприяючи операціям з обслуговування клієнтів

Продовження таблиці Б.1

Назва файлу	Опис
client.py	Охоплює функціонал, який використовується користувачами бота
is_support_middleware.py	Проміжне програмне забезпечення для перевірки наявності менеджерів підтримки та призначення підтримки клієнту
admin_schedule_list.py	Додаткова функція для відображення та зміни робочого графіка адміністратором
get_current_schedule.py	Функція для отримання та відображення поточного робочого графіку компанії
get_users_list.py	Функціональність для генерації списку всіх користувачів, які взаємодіють з ботом

ДОДАТОК В

Фрагмент коду файлу database.py

```
import random
import sqlite3
import os

current_directory = os.path.dirname(os.path.realpath(__file__))
DATABASE = os.path.join(current_directory, 'database.db')

def get_db_connection():
    conn = sqlite3.connect(DATABASE)
    conn.row_factory = sqlite3.Row
    return conn

def get_schedule():
    conn = get_db_connection()
    cursor = conn.execute(
        'SELECT days.*, schedule.* FROM days LEFT OUTER JOIN
schedule ON days.id = schedule.day_id')
    items = cursor.fetchall()
    conn.close()
    return items

def get_day_by_id(day_id):
    conn = get_db_connection()
    cursor = conn.execute(
        'SELECT * FROM days WHERE id = ?', (day_id,))
    items = cursor.fetchall()
    conn.close()
    return items

def add_schedule_to_db(day, time_start, time_end):
    conn = get_db_connection()
    try:
        conn.execute('INSERT INTO schedule (day_id, time_start,
time_end) VALUES (?, ?, ?)',
                    (day, time_start, time_end))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error adding schedule to database: {str(e)}")
        conn.close()
        return False
```

Продовження додатка В

```
def update_schedule_in_db(schedule_id, time_start, time_end):
    conn = get_db_connection()
    try:
        conn.execute('UPDATE schedule SET time_start = ?,
time_end = ? WHERE day_id = ?',
                    (time_start, time_end, schedule_id))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error updating schedule in the database:
{str(e)}")
        conn.close()
        return False

def delete_schedule_by_id(schedule_id):
    conn = get_db_connection()
    try:
        conn.execute('DELETE FROM schedule WHERE day_id = ?',
(schedule_id,))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error deleting schedule in the database:
{str(e)}")
        conn.close()
        return False

def get_all_days():
    conn = get_db_connection()
    cursor = conn.execute('SELECT * FROM days')
    items = cursor.fetchall()
    conn.close()
    return items

def db_profile_exist(uid):
    conn = get_db_connection()
    cursor = conn.execute('SELECT * FROM profiles WHERE
telegram_id = ?', (uid,))
    items = cursor.fetchall()
    conn.commit()
    conn.close()
    return items
```

Продовження додатка В

```

def db_profile_add(telegram_id, username, full_name, email):
    conn = get_db_connection()
    try:
        conn.execute('INSERT INTO profiles (telegram_id,
username, full_name, email) VALUES (?, ?, ?, ?)',
                    (telegram_id, username, full_name, email))
        conn.commit()
        conn.close()
        return True
    except sqlite3.IntegrityError:
        print(f"Error adding profile to database: Email
'{email}' already exists.")
        conn.close()
        return False
    except Exception as e:
        print(f"Error adding profile to database: {str(e)}")
        conn.close()
        return False

def db_profile_access():
    conn = get_db_connection()
    cursor = conn.execute("SELECT telegram_id FROM profiles
WHERE access_level IN (1, 2)")
    items = cursor.fetchall()
    telegram_ids = [item[0] for item in items]
    conn.close()
    return telegram_ids

def db_profile_free_for_support():
    conn = get_db_connection()
    cursor = conn.execute(
        "SELECT telegram_id FROM profiles WHERE access_level IN
(1, 2) AND support_status = 0 AND work_status = 1")
    items = cursor.fetchall()
    telegram_ids = [item[0] for item in items]
    random.shuffle(telegram_ids)
    conn.close()
    return telegram_ids

def db_profile_banned(uid):
    conn = get_db_connection()
    cursor = conn.execute('SELECT ban FROM profiles WHERE
telegram_id = ?', (uid,))
    result = cursor.fetchone()
    if result['ban'] == 1:
        conn.close()
        return True
    else:
        conn.close()
        return False

```

Продовження додатка В

```
def db_profile_update(access, uid):
    conn = get_db_connection()
    try:
        conn.execute("UPDATE profiles SET access = ? WHERE id =
?", (access, uid))
        conn.commit()
        return True
    except Exception as e:
        print(f"Error adding profiles to database: {str(e)}")
        conn.close()
        return False

def change_work_status(status, telegram_id):
    conn = get_db_connection()
    try:
        conn.execute('UPDATE profiles SET work_status = ? WHERE
telegram_id = ?', (status, telegram_id))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False

def set_support_status(user_id, support_id):
    conn = get_db_connection()
    try:
        conn.execute('UPDATE profiles SET support_status = ?
WHERE telegram_id = ?', (user_id, support_id))
        conn.execute('UPDATE profiles SET support_status = ?
WHERE telegram_id = ?', (support_id, user_id))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False
```


Продовження додатка В

```

def reset_support_status(telegram_id):
    conn = get_db_connection()
    try:
        cursor = conn.execute('SELECT support_status FROM
profiles WHERE telegram_id = ?', (telegram_id,))
        support_status_id = cursor.fetchone()[0]
        conn.execute('UPDATE profiles SET support_status = 0
WHERE telegram_id = ?', (telegram_id,))
        conn.execute('UPDATE profiles SET support_status = 0
WHERE telegram_id = ?', (support_status_id,))
        conn.commit()
        conn.close()
        return support_status_id
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False

def get_name(telegram_id):
    conn = get_db_connection()
    try:
        cursor = conn.execute('SELECT full_name FROM profiles
WHERE telegram_id = ?', (telegram_id,))
        support_name = cursor.fetchone()[0]

        conn.commit()
        conn.close()
        return support_name
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False

def get_work_status(telegram_id):
    conn = get_db_connection()
    try:
        cursor = conn.execute('SELECT work_status FROM profiles
WHERE telegram_id = ?', (telegram_id,))
        work_status = cursor.fetchone()[0]

        conn.commit()
        conn.close()
        return work_status
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False

```

Продовження додатка В

```
def get_session(u_id):
    conn = get_db_connection()
    try:
        cursor = conn.execute("SELECT support_status FROM
profiles WHERE telegram_id = ?", (u_id,))
        result = cursor.fetchone()
        conn.close()
        return result[0] if result else None
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False

def check_admin_access(u_id):
    conn = get_db_connection()
    try:
        cursor = conn.execute("SELECT access_level FROM profiles
WHERE telegram_id = ?", (u_id,))
        result = cursor.fetchone()
        conn.close()
        if result:
            access_level = result[0]
            return access_level == 2
        else:
            return False
    except Exception as e:
        print(f"Error checking admin access in the database:
{str(e)}")
        conn.close()
        return False

def get_profiles_list(u_id):
    conn = get_db_connection()
    cursor = conn.execute('SELECT * FROM profiles WHERE
telegram_id != ?', (u_id,))
    items = cursor.fetchall()
    conn.commit()
    conn.close()
    return items
```

Продовження додатка В

```
def set_ban_status(telegram_id, status):
    if status == "ban":
        is_ban = 1
    else:
        is_ban = 0
    conn = get_db_connection()
    try:
        conn.execute('UPDATE profiles SET ban = ? WHERE
telegram_id = ?', (is_ban, telegram_id))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False

def set_access_status(telegram_id, level):
    conn = get_db_connection()
    try:
        conn.execute('UPDATE profiles SET access_level = ? WHERE
telegram_id = ?', (level, telegram_id))
        conn.commit()
        conn.close()
        return True
    except Exception as e:
        print(f"Error updating profiles in the database:
{str(e)}")
        conn.close()
        return False
```

ДОДАТОК Г

Загальна блок-схема з інтерфейсами користувача, менеджера та адміністратора

