

**Київський національний торговельно-економічний університет**

**Кафедра комп'ютерних наук та інформаційних систем**

**ВИПУСКНА КВАЛІФІКАЦІЙНА ПРОЕКТ**

на тему:

**«Розробка програмного засобу підтримки взаємодії з  
клієнтами медичної клініки»**

Студентки 4 курсу, 10 групи,  
спеціальності  
122 «Комп'ютерні науки»

Буракової  
Дар'ї  
Миколаївни

*підпис студента*

Науковий керівник  
кандидат технічних наук, доцент

Козлов Валерій  
Володимирович

*підпис керівника*

Гарант освітньої програми  
кандидат технічних наук, доцент

Демідов Павло  
Георгійович

*підпис керівника*

**Київ 2021**

**Київський національний торговельно-економічний університет**

Факультет інформаційних технологій  
Кафедра комп'ютерних наук та інформаційних систем  
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри \_\_\_\_\_

**Затверджую**  
Пурський О. І.

«    » грудня 2020р.

**Завдання**  
**на випускний кваліфікаційний проект студентки**

**Буракової Дар'ї Миколаївні**  
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)  
«Розробка програмного засобу підтримки взаємодії з клієнтами медичної клініки»  
Затверджена наказом ректора від «15» грудня 2020 р. № 3780
2. Строк здачі студентом закінченої роботи 28 травня 2021 року
3. Цільова установка та вихідні дані до роботи  
Мета роботи: обґрунтування та розробка чат-бота, з урахуванням сучасних тенденцій побудови функціональних організаційних структур підтримки взаємодії з клієнтами.  
Об'єкт дослідження: процес проектування чат-бота для підтримки взаємодії з клієнтами медичної клініки.  
Предмет дослідження: засоби створення чат-бота для підтримки взаємодії з клієнтами медичної клініки.
4. Перелік графічного матеріалу \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

| Розділ | Консультант<br>(прізвище, ініціали) | Підпис, дата   |                  |
|--------|-------------------------------------|----------------|------------------|
|        |                                     | Завдання видав | Завдання прийняв |
| 1      | Козлов В. В.                        | 15.12.2020 р.  | 15.12.2020 р.    |
| 2      | Козлов В. В.                        | 15.12.2021 р.  | 15.12.2021 р.    |
| 3      | Козлов В. В.                        | 15.12.2021 р.  | 15.12.2021 р.    |

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

### ВСТУП

#### РОЗДІЛ 1. Сучасні технології в розробці чат-бота.

##### 1.1 Концепція чат-бота. Архітектура сучасного чат-бота

##### 1.2. Огляд існуючих засобів проектування

##### Висновки до розділу

#### РОЗДІЛ 2. Організація розробки чат-бота для підтримки взаємодії з клієнтами медичної клініки

##### 2.1. Постановка задачі

##### 2.2 Вибір платформи для впровадження чат-бота

##### 2.3. Вибір мови програмування

##### 2.4. Вибір середовища програмування

##### 2.5. Вибір бази даних для чат-бота

##### 2.6. Функціональні вимоги до чат-бота

##### Висновки до розділу

#### РОЗДІЛ 3. Розробка чат-бота для підтримки взаємодії з клієнтами медичної клініки

##### 3.1. Створення чат-бота за допомогою BotFather

##### 3.2 Створення з'єднання з базою MySQL

##### 3.3. Розробка структури чат-бота

##### 3.4. Розробка структури бази даних



### 3.5. Оцінка отриманих результатів проектування

#### Висновки до розділу

#### ВИСНОВКИ

#### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

### 7. Календарний план виконання роботи

| №<br>Пор . | Назва етапів випускної кваліфікаційної роботи                                          | Строк виконання етапів роботи |            |
|------------|----------------------------------------------------------------------------------------|-------------------------------|------------|
|            |                                                                                        | За планом                     | фактично   |
| 1          | 2                                                                                      | 3                             | 4          |
| 1          | <i>Вибір теми випускної кваліфікаційної роботи</i>                                     | 01.10.2020                    | 01.10.2020 |
| 2          | <i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>             | 15.12.2019                    | 15.12.2020 |
| 3          | <i>Вступ</i>                                                                           | 03.02.2021                    | 01.02.2021 |
| 4          | <i>РОЗДІЛ 1. Сучасні технології в розробці чат-бота.</i>                               | 28.02.2021                    | 27.02.2021 |
| 5          | <i>РОЗДІЛ 2. Організація розробки чат-бота для підтримки клієнтів медичної клініки</i> | 06.04.2021                    | 04.04.2021 |
| 6          | <i>РОЗДІЛ 3 Розробка чат-бота для підтримки клієнтів медичної клініки.</i>             | 12.05.2021                    | 11.05.2021 |
| 7          | <i>Висновки</i>                                                                        | 15.05.2021                    | 14.05.2021 |
| 8          | <i>Здача випускної кваліфікаційної роботи на кафедру науковому керівнику</i>           | 20.05.2021                    | 20.05.21   |
| 9          | <i>Попередній захист випускної кваліфікаційної роботи</i>                              | 03.06.2021                    | 03.06.2021 |
| 11         | <i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>   | 04.06.2021                    | 04.06.2021 |
| 12         | <i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедру</i>        | 07.06.2021                    | 07.06.2021 |
| 13         | <i>Публічний захист випускної кваліфікаційної роботи</i>                               | За розкладом роботи ЕК        |            |

8. Дата видачі завдання «15» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Козлов В. В.

(прізвище, ініціали, підпис)

## 10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

### 11. Завдання прийняв до виконання студентка-дипломник

Буракова Д. М.

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

31.05.2021 p.

(*nidnuc*, *dam*)

### 13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента \_\_\_\_\_  
(прізвище, ініціали)  
може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми \_\_\_\_\_ Демідов П.Г.  
(підпис, прізвище, ініціали)

Завідувач кафедри \_\_\_\_\_ Пурський О.І.  
(підпис, прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

## АНОТАЦІЯ

Буракова Д.М. Розробка програмного засобу підтримки взаємодії з клієнтами медичної клініки.

Мета роботи: обґрунтування та розробка чат-бота, з урахуванням сучасних тенденцій побудови функціональних організаційних структур підтримки взаємодії з клієнтами.

Об'єкт дослідження: процес проектування чат-бота для підтримки взаємодії з клієнтами медичної клініки.

Предмет дослідження: засоби створення чат-бота для підтримки взаємодії з клієнтами медичної клініки.

Системами обміну миттєвими повідомленнями користуються мільйони людей у всьому світі. Вони еволюціонували від засобу спілкування між людьми до засобу отримання інформації та потужний маркетинговий інструмент. Це зумовлено в першу чергу можливістю впровадження чат-ботів на платформи. В кваліфікаційній роботі розглядаються основні принципи розробки чат-ботів та можливості їх використання для автоматизації підтримки клієнтів.

Результатом проведеного дослідження стало створення чат-бота для підтримки взаємодії з клієнтами медичної клініки у месенджері Telegram. Система є сучасною та зручною, задовольняє всі вимоги з точки зору функціональності, придатності використання, привабливості для користувача.

**КЛЮЧОВІ СЛОВА:** БОТ, ЧАТ-БОТ, ТЕЛЕГРАМ-БОТ, TELEGRAM, PYTHON, КЛІНІКА, ПІДТРИМКА КЛІЄНТІВ



## SUMMARY

Burakova D.M. Development of a software tool for support of interaction with clients in the medical clinic.

Objective: to develop a chat-bot, taking into account current trends in the creation of functional organizational structures for support of interaction with patients.

Object of study: the process of designing a chat-bot for support of interaction with patients in the medical clinic.

Subject of study: tools for creating a chat-bot for support of interactions with patients in the medical clinic.

Instant messaging systems are used by millions of people around the world. They have evolved from a means of communication between people to a means of receiving information and a powerful marketing tool. This is primarily due to the possibility of introducing chatbots on platforms. In this work the main principles of developing chat-bots and the possibilities of using them for automation of customer support are discussed.

The research resulted in the creation of a chat-bot for support of interaction with medical clinic clients via Telegram messenger. The system is good and user-friendly and meets all the requirements in terms of functionality, user friendliness, and user-friendliness.

**KEYWORDS:** BOT, CHATBOT, TELEGRAM BOT, TELEGRAM, PYTHON, CLIENT SUPPORT



## ЗМІСТ

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| ВСТУП.....                                                                           | 11 |
| РОЗДІЛ 1 Сучасні технології в розробці чат-бота.....                                 | 13 |
| 1.1 Концепція чат-бота. Архітектура сучасного чат-бота.....                          | 13 |
| 1.2 Огляд існуючих засобів проектування.....                                         | 20 |
| Висновки до розділу.....                                                             | 22 |
| РОЗДІЛ 2 Організація розробки чат-бота для підтримки взаємодії з клієнтами.....      | 23 |
| 2.1 Постановка задачі.....                                                           | 23 |
| 2.2 Вибір платформи для впровадження чат-бота.....                                   | 23 |
| 2.3. Вибір мови програмування.....                                                   | 27 |
| 2.4 Вибір середовища програмування.....                                              | 29 |
| 2.5 Вибір бази даних для чат-бота.....                                               | 30 |
| 2.6 Функціональні вимоги до чат-бота.....                                            | 32 |
| Висновки до розділу.....                                                             | 33 |
| РОЗДІЛ 3 Розробка чат-бота для підтримки взаємодії з клієнтами медичної клініки..... | 34 |
| 3.1 Створення чат-бота за допомогою BotFather.....                                   | 34 |
| 3.2 Створення з'єднання з базою MySQL.....                                           | 37 |
| 3.3 Розробка структури чат-бота.....                                                 | 38 |
| 3.4 Розробка структури бази даних.....                                               | 48 |
| 3.5 Оцінка отриманих результатів проектування.....                                   | 51 |

|                                 |    |
|---------------------------------|----|
|                                 | 10 |
| Висновки до розділу.....        | 59 |
| ВИСНОВКИ.....                   | 60 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 62 |
| ДОДАТКИ.....                    | 64 |
| Додаток А. Лістинг коду.....    | 64 |

## ВСТУП

У сучасному світі однією з основних рушійних сил розвитку інформаційних технологій є конкуренція. Швидке отримання інформації дає перевагу перед суперниками. Разом з цим виникає проблема, яка полягає в тому, що людина фізично не може обробляти занадто великий масив даних за заданий час і ефективно використовувати отриману інформацію. Отже, існує необхідність в автоматизації таких процесів, де людським фактором можна було б знехтувати.

В останні роки стрімко зростає популярність додатків для обміну миттєвими повідомленнями. Месенджери міцно увійшли в життя суспільства. Це стало основою для використання додатків для обміну миттєвими повідомленнями, як каналу продажів і взаємодії з клієнтами. Зараз месенджери все частіше використовуються не тільки як засіб спілкування між людьми, а й як інструмент маркетингу. Це обумовлено можливістю автоматизації процесів за допомогою чат-ботів. Чат-боти досить популярні в електронній комерції, але в останні роки вони поширилися на інші види бізнесу, такі як банки, нерухомість і, звичайно, приватні медичні клініки та постачальники медичних послуг.

Чат-боти для підтримки клієнтів медичної клініки відповідають за взаємодію з потенційними і поточними пацієнтами, допомагають їм знаходити лікарів, консультують про наявність послуг та їх ціни, про графік роботи клініки і фахівців, отримують доступ до інформації про методи лікування.

Актуальність теми. Автоматизація процесів шляхом створення чат-ботів зумовлена широким розповсюдженням сучасних месенджерів та використанням їх у різноманітних галузях. Чат-боти дозволяють у багато разів збільшити швидкість обробки інформації, спростити комунікацію між постачальником послуг та клієнтом, оптимізувати витрати та залучити нових споживачів. На українському ринку ця технологія лише починає розвиватися,

тому попит на фахівців для розробки чат-ботів досить великий і буде лише зростати.

Мета роботи: обґрунтування та розробка чат-бота, з урахуванням сучасних тенденцій побудови функціональних організаційних структур підтримки взаємодії з клієнтами.

Об'єкт дослідження: процес проектування чат-бота для підтримки взаємодії з клієнтами медичної клініки.

Предмет дослідження: засоби створення чат-бота для підтримки взаємодії з клієнтами медичної клініки.



## РОЗДІЛ 1 СУЧАСНІ ТЕХНОЛОГІЇ В РОЗРОБЦІ ЧАТ-БОТА

### 1.1 Концепція чат-бота. Архітектура сучасного чат-бота

Кожен хоч раз у своєму житті, якщо не користувався послугами чат-ботів, то хоча б стикався з ними у вигляді спливаючого вікна на якомусь сайті. "Здравствуйте. Чим можу вам допомогти?" питає чат-бот і пропонує можливі варіанти відповідей в залежності від тематики сайту.

Чат-бот – це додаток, віртуальний помічник, призначений для спрощення взаємодії між людиною та веб-сервісом. Концепція полягає в тому, що подібна програма імітує розмову з реальною людиною і здатна обмінюватися повідомленнями в чаті.

Чат-боти використовуються в багатьох сферах і виконують найрізноманітніші операції. Вони справляються з рутинними задачами, які можна звести до певного алгоритму, здійснюють пошук та агрегування даних, розповсюджують інформацію тощо. Компанія Drift, що спеціалізується на розробці чат-ботів, провела опитування(рис.1.1) [1] на тему того, які функції очікуються від віртуальних помічників у першу чергу:

- 37% - отримання швидких відповідей в екстрених випадках;
- 35% - рішення проблем;
- 35% - отримання детальних відповідей і пояснень;
- 33% - бронювання (столика в ресторані, номери в готелі та ін.);
- 29% - оплата рахунків;
- 27% - покупка предметів першої необхідності;
- 22% - ідеї для покупок;
- 22% - підписка на розсилку або новинний ресурс;
- 18% - комунікація з декількома брендами за допомогою однієї програми;
- 13% - допомога при покупці дорогих товарів.

### What do you predict you would use a chatbot for?

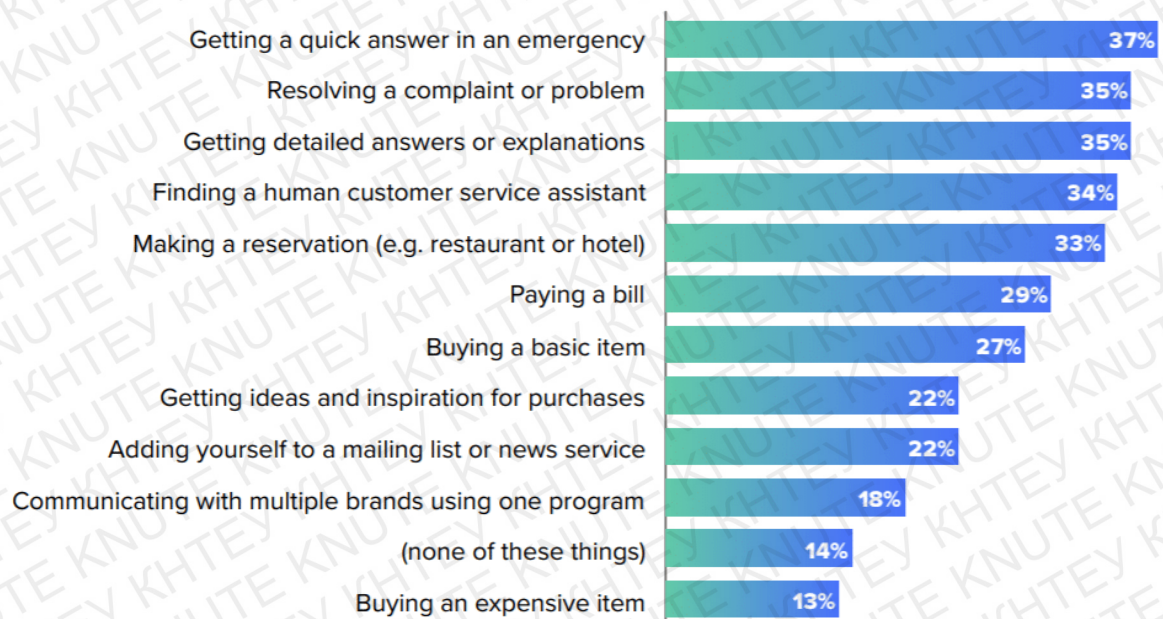


Рис. 1.1. Звіт про стан чат-ботів від компанії Drift

Існують різні підходи до класифікації чат-ботів. Однак можна виділити два основні параметри, за якими розподіляють чат-ботів: алгоритм праці та функціональність.

Класифікація чат-ботів за алгоритмом:

- Обмежений чат-бот – працює за заздалегідь визначеними командами, які створені на основі найбільш розповсюджених питань або ключових слів. Чат-боти такого типу не підтримують відкриті та розгорнуті відповіді на питання, але замість цього пропонують альтернативу у вигляді кнопок для взаємодії між користувачем і веб-сервісом. Практичне застосування таких ботів обмежено через те, що користувач може зіткнутися з відсутністю необхідного йому запиту в списку пропонованих рішень.
- Самонавчальний чат-бот - працює на основі штучних нейронних мереж, за допомогою методів розуміння природної мови (NLU, Natural Language Understanding) та методів обробки природної мови (NLP, Natural Language Processing).



Обробка природної мови (NLP) – це область штучного інтелекту, яка досліджує маніпулювання комп'ютерами з текстом на природній мові. Знання про розуміння та використання людської мови збираються для розробки методів, які допомагають комп'ютерам розуміти та маніпулювати природними виразами для виконання бажаних завдань[2]. Більшість методів НЛП засновані Machine Learning.

Розуміння природної мови (NLU) лежить в основі будь-якого завдання NLP. Це техніка, яка необхідна для реалізації природних користувацьких інтерфейсів, таких як чат-бот. NLU націлене на отримання контексту і сенсу з користувацьких повідомлень на природній мові, які можуть бути неструктурованими, і реагувати відповідним чином відповідно до намірів користувача[2]. Чат-бот, що навчається на базі NLU, визначає наміри користувача і витягує специфічні для даної області поняття. Набір(intent) - це відповідність між тим, що говорить користувач, і тим, які дії треба вжити чат-боту. Дії відповідають крокам, які чат-бот зробить, коли певні наміри будуть викликані користувацьким повідомленням, і можуть мати параметри для вказівки детальної інформації про них[3].

Одним з найбільших недоліків чат-ботів на основі штучного інтелекту можна назвати те, що вони вимагають великого набору даних для того, щоб натурально імітувати розмову з людиною і правильно визначати наміри користувачів. Але швидкий розвиток ІТ та Big Data робить самонавчальних чат-ботів однією з найперспективніших технологій.

- Гібридний чат-бот - поєднує в собі ознаки обмеженого та самонавчального чат-бота. Такі чат-боти працюють за заздалегідь визначеними командами, але при цьому використовують штучний інтелект для визначення користувацьких намірів і збору даних з інформації, що надається. Гібридний тип чат-ботів широко використовується в комерційній діяльності.

Класифікація чат-ботів за функціоналом:

- Розмовні чат-боти - застосовуються для комунікації з користувачем. Вони можуть надавати інформацію, відповідати на запитання або розмовляти з людиною, імітуючи діалог, без конкретно поставленої задачі.
- Функціональні чат-боти(боти-асистенти) – мають конкретну мету, для досягнення якої використовують користувацькі дані. Вони надають можливість виконання певних дій, наприклад бронювання квитків, переказ грошових коштів, оформлення іпотеки тощо.

Чат-ботів для комерційної діяльності класифікують здебільшого за сценарієм їх використання(use case).Відштовхуючись від очікувань користувачів, чат-ботів для бізнесу можна розділити на види в залежності від функцій, які вони виконують:

- Підтримка клієнтів. Автоматизація обробки простих, часто повторюваних запитів від клієнтів дозволяє зменшити навантаження на співробітників служби підтримки, оптимізувати роботу персоналу та підвищити якість і швидкість обслуговування клієнтів. Прикладом бота для підтримки клієнтів є Telegram-бот Арт-заводу «Платформа». Завдяки чат-боту можна подивитися розклад арт-виставок, сцен, карту фуд-кортів, задати питання службі підтримки, дізнатися про вартість оренди, отримати контакти керівництва, купити квиток на захід.
- Поширення контенту. Великі світові і українські компанії використовують ботів для поширення рекламних або інформаційних статей. Крім того, такі програми вміють відбирати матеріали відповідно до попередніх запитів користувачів. Прикладом такого бота є Viber-бот компанії «Розетка», який формує розсилку повідомлень на основі купівель споживачів, пропонує товари та інформує про акції. На рис. 1.2 зображені скріншоти повідомлень від «Розетки».



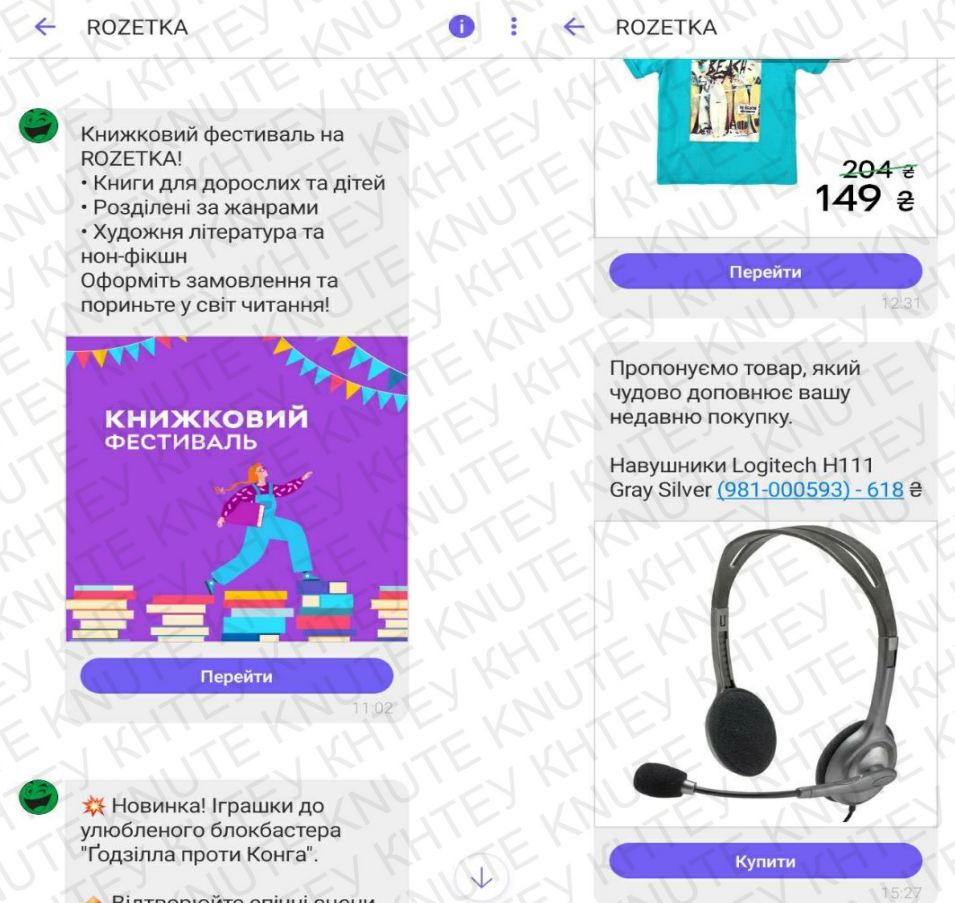


Рис. 1.2 Приклади повідомлень від чат-боту «ROZETKA»

- Пошук за параметрами. Алгоритм праці полягає в тому, що користувач встановлює параметри, а чат-бот здійснює пошук. Прикладом такої програми є Telegram-бот від світової компанії «Авіасейлс» (рис. 1.3) @AviaSalesBot, який обирає квитки на зворотні напрямки та дні.

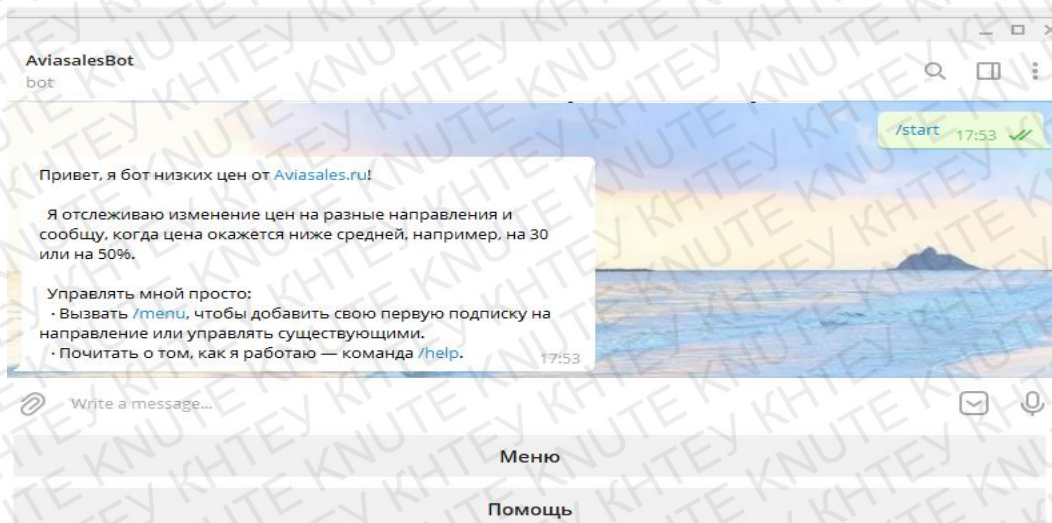


Рис. 1.3 Інтерфейс чат-бота від компанії «Авіасейлс»

- Збір інформації. Чат-боти збирають інформацію про потенційних покупців (адреса електронної пошти, номер телефону тощо). Інформація використовується для складання аналітичних звітів та значно полегшує роботу фахівців. Визначивши, що покупець готовий здійснити покупку, робот зв'язує його з «живим» менеджером з продажу.
- Прийом замовлень та продаж. Такі чат-боти оформлюють замовлення користувача та здійснюють продаж товарів чи послуг прямо у чаті. Поки що в Україні доступні чат-боти небагатьох компаній. Прикладом може слугувати Viber-бот від компанії «Фокстрот», завдяки якому користувач може замовляти та сплачувати товари безпосередньо в месенджері, без переходу на інші ресурси.

Проектування і розробка чат-ботів включає в себе безліч методик.

Розуміння того, які функції має виконувати чат-бот, допомагає розробникам вибрати алгоритми або платформи та інструменти для його створення. У той же час це допомагає кінцевим користувачам зрозуміти, чого очікувати.

Першим кроком в проектуванні будь-якої системи є поділ її на складові частини, щоб можна було слідувати модульному підходу до розробки [4]. На рис. 1.4 представлена загальна архітектура чат-бота.

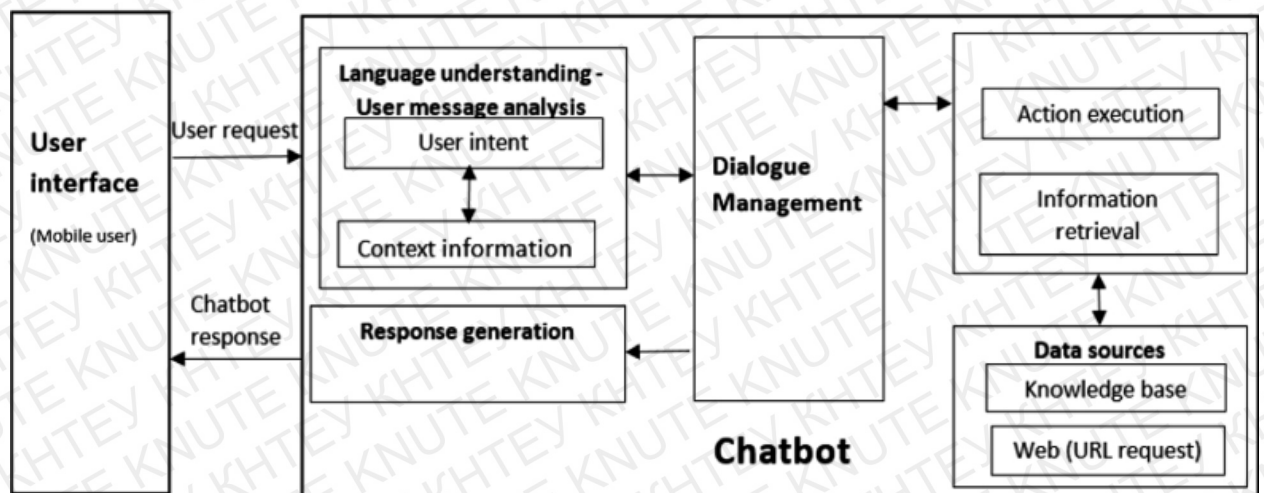


Рис. 1.4 Архітектура чат-бота



Процес починається з запиту користувача в один з доступних йому каналів, в якості якого можуть виступати месенджери, такі як Telegram, Facebook, WhatsApp, Viber або Skype. У запиті міститься певний намір(intent), бажання, яке користувач хоче задовольнити.

Після того як чат-бот отримує запит користувача, він аналізує його, щоб визначити намір користувача і пов'язану з ним інформацію. Запит надходить у платформу Dialogue Management. Її завдання - ефективно обробити запит користувача, віддавши результат. Для цього діалогові платформи використовують безліч технологій, таких як токенізація, морфологічний аналіз, аналіз семантичної подібності, ієрархізація гіпотез відповідно до поточного контексту, формування запитів на Machine Learning[4]. Чат-бот виконує дії, які були запитані, або витягує дані, які потрібні користувачу, з джерел, якими можуть бути база знань чат-бота, або зовнішні системи, доступ до яких здійснюється через виклик API[5]. Прикладом таких зовнішніх систем може бути 1C, SAP, CRM-системи.

Після отримання даних компонент генерації відповідей використовує генерацію природної мови (NLG)[6] для підготовки відповіді користувачу на природній мові на основі інформації про наміри та контексті, отриманого від аналізу повідомлень користувача.

Отже, основний цикл обробки запиту користувача складається з наступних подій і дій:

- 1) Запит користувача в один з доступних йому каналів.
- 2) Потрапляння запиту у Dialogue Management.
- 3) Dialogue Management знаходить необхідні дані у базі знань чат-бота або зовнішніх джерелах.
- 4) Запит користувача обробляється у NLU-модулі, де визначається намір користувача. Якщо чат-бот обмежений, не підтримує обробку текстових повідомлень та взаємодіє з користувачем шляхом використання кнопок чи їх аналогів, цей крок пропускається.



- 5) Виконання чат-ботом дій згідно заздалегідь визначеному сценарію(скрипту) по обробці клієнтського кейса.
- 6) Генерування відповіді з використанням природної мови(NLG).
- 7) Відправка відповіді користувачеві.

## 1.2 Огляд існуючих засобів проектування

Для розробки чат-бот існує безліч комерційних технологій і технологій з відкритим вихідним кодом. Кількість технологій, пов'язаних з чат-ботами, зростає з кожним днем [7].

Існує три підходи до розробки чат-ботів:

- 1) Проектування чат-бота з використанням мови програмування, за допомогою якої можна створити web API, наприклад Node.js, PHP, Java, Python, C ++, Ruby, Lisp.
- 2) Проектування чат-бота з використанням різноманітних фреймворків та спеціальних сервісів:
  - BotKit- open-source набір інструментів для створення ботів з докладною документацією;
  - Claudia- конструктор чат-ботів для роботи в AWS Lambda;
  - Bottr- фреймворк на Node.js з готовим додатком для тестування;
  - wit.ai- сервіс від Facebook, який приймає текст або голосові повідомлення і використовує NLP-методи для управління відповідями на запити;
  - motion.ai- сервіс для створення чат-ботів;
  - api.ai- сервіс, який використовує NLP-методи для створення ботів і визначення можливих сценаріїв їх спілкування з клієнтами;

Ці платформи підтримують машинне навчання. Вони мають деякі загальні стандартні функції (працюють в хмарі, підтримують різні мови програмування і природні мови), але значно відрізняються в інших аспектах[8].

- 3) Проектування чат-бота з використанням веб-сервісів для самостійного налаштування програми(конструкторів). Прикладами конструкторів для створення чат-ботів є SendPulse, Chatfuel, MobileMonkey, RASA ManyChat, Flow XO, Chatterbot, DialogFlow.

## Висновки до розділу

Чат-бот – це додаток, віртуальний помічник, призначений для спрощення взаємодії між людиною та веб-сервісом. Чат-боти використовуються в багатьох сферах і виконують найрізноманітніші операції. Вони справляються з рутинними задачами, які можна звести до певного алгоритму, здійснюють пошук та агрегування даних, розповсюджують інформацію тощо. Згідно з опитуванням від компанії Drift від віртуальних помічників очікується, в першу чергу, отримання швидких відповідей в екстрених випадках.

Існують різні підходи до класифікації чат-ботів. За алгоритмом праці чат-ботів поділяють на обмежених – тих, які працюють за чітко визначеними командами, та на самонавчальних – тих, які працюють на основі штучних нейронних мереж, за допомогою методів розуміння природної мови (NLU, Natural Language Understanding) та методів обробки природної мови (NLP, Natural Language Processing). Чат-ботів для комерційної діяльності класифікують здебільшого за сценарієм їх використання (use case). Серед них є: підтримка клієнтів, поширення контенту, пошук за параметрами, збір інформації, прийом замовлень та продаж. Використання чат-ботів у бізнес-процесах сприяє зростанню продажів, поширенню клієнтської бази, зменшенню навантаження на співробітників та пришвидшенню виконання операцій у декілька разів, що дає перевагу перед конкурентами.

Проектування і розробка чат-ботів включає в себе безліч методик. Чат-бот може бути розроблений з використанням мов програмування, різноманітних фреймворків, спеціальних сервісів, конструкторів – все залежить від того, який функціонал має підтримувати віртуальний помічник.



## РОЗДІЛ 2 ОРГАНІЗАЦІЯ РОЗРОБКИ ЧАТ-БОТА ДЛЯ ПІДТРИМКИ ВЗАЄМОДІЇ З КЛІЄНТАМИ МЕДИЧНОЇ КЛІНІКИ

### 2.1 Постановка задачі

Метою розробки є створення чат-бота для підтримки взаємодії з клієнтами медичної клініки.

Чат-бот має виконувати наступні функції:

- Здійснювати запис на прийом до певного лікаря;
- Надавати користувачеві інформацію про клініку;
- Відповідати на типові запитання про послуги, ціни, графік роботи клініки та спеціалістів, адресу, контакти та ін.

Можна виділити наступні цілі створення застосування:

- Спрощення взаємодії між пацієнтами та клінікою(автоматизація);
- Збільшення клієнтурної бази медичної клініки із числа користувачів месенджера через цілодобову доступність чат-боту.

В рамках цих цілей постають такі задачі розробки:

- розробити модуль роботи із месенджером;
- розробити модуль роботи із СКБД.

### 2.2 Вибір платформи для впровадження чат-бота

Месенджер – це клієнтська програма, яка необхідна для обміну миттєвими повідомленнями(Instant messaging). Сучасні месенджери дозволяють обмінюватися не тільки текстовими повідомленнями, але й фотографіями, відео, файлами тощо. Часто месенджери підтримують голосовий зв'язок і відеозв'язок, і для того, щоб зв'язатися з абонентом, потрібен лише доступ до Інтернету.

На рис. 2.1 представлена статистика кількості активних користувачів найпопулярніших месенджерів у всьому світі станом на січень 2021 року згідно з дослідженням, яке викладене на сайті [statista.com](https://www.statista.com) [9].

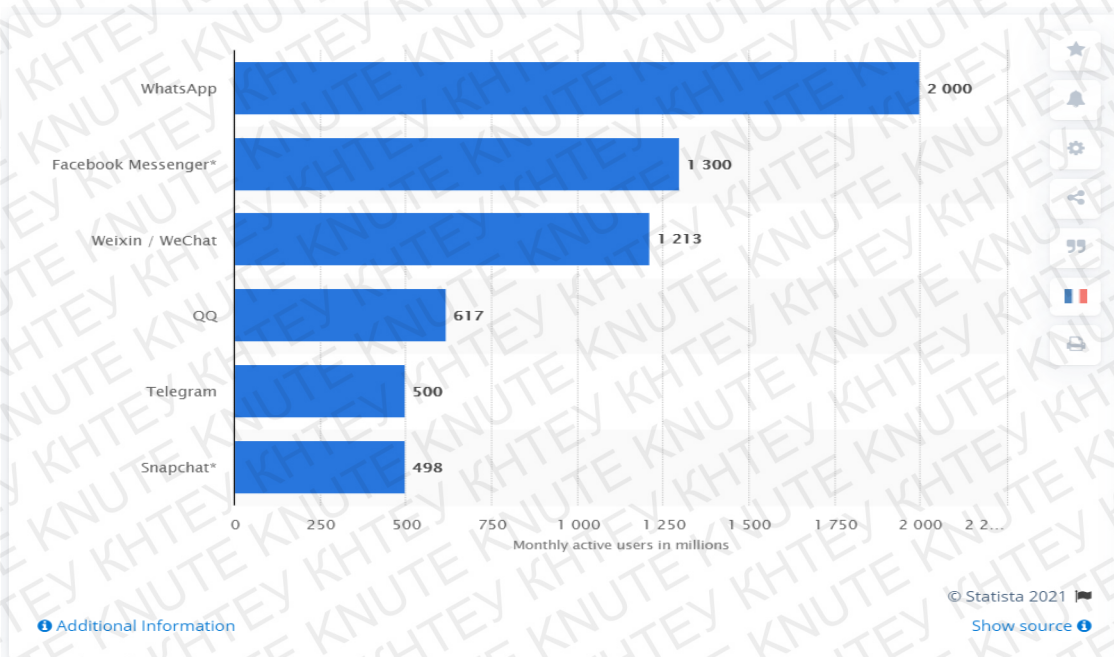


Рис. 2.1 Кількість активних користувачів у найпопулярніших месенджерах

Безумовним світовим лідером є месенджер WhatsApp з кількістю користувачів близько 2 млрд. Але варто сказати, що темпи зростання кількості користувачів у цього месенджера досить низькі, що можна побачити на наступному графіку [10] (рис. 2.2):

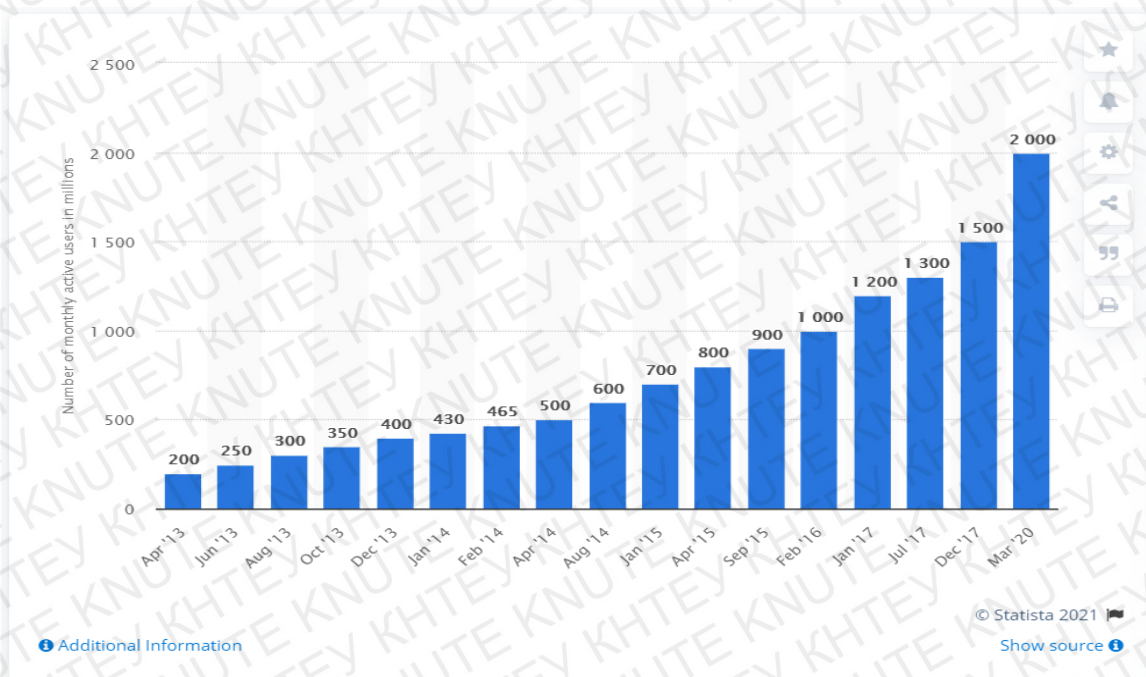


Рис. 2.2 Кількість активних користувачів у месенджері WhatsApp



Telegram – месенджер зі стрімко зростаючою аудиторією. З 2016 до 2021 року(рис.2.3) кількість активних користувачів Telegram збільшилася приблизно в 5 разів, в той час як кількість активних користувачів WhatsApp збільшилася всього в 2 рази.

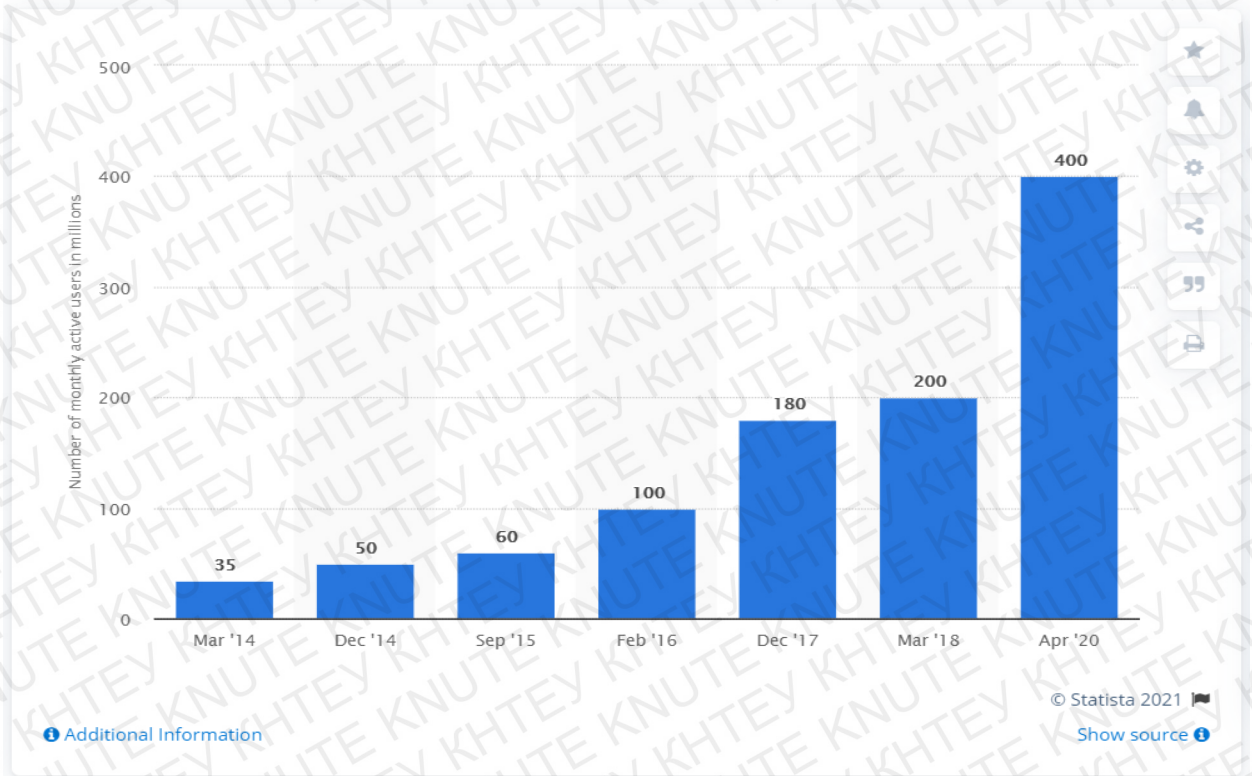


Рис. 2.3 Кількість активних користувачів у месенджері Telegram [11]

Українська компанія Kantar провела дослідження на тему використання мобільних додатків. Рейтинг, складений в березні 2020 року [12], демонструє, що в Україні найпопулярнішим месенджером є Viber(97,9%), друге місце займає Facebook Messenger(73,6%), а третє - Telegram(70,6%) (рис. 2.4). Але вже в січні 2021 року [13] Telegram виривається на друге місце з показником 76,8%(зростання на 6,2%), в той час як кількість активних користувачів Viber та Facebook Messenger зменшилися на 1,1% та 4,3% відповідно (рис. 2.5).



## Рейтинг мобільних додатків Березень 2020

|   |                                                                                   |              |       |    |                                                                                    |            |       |
|---|-----------------------------------------------------------------------------------|--------------|-------|----|------------------------------------------------------------------------------------|------------|-------|
| 1 |  | Viber        | 97,9% | 9  |  | Instagram  | 71,3% |
| 2 |  | Chrome       | 96,4% | 10 |  | Telegram   | 70,6% |
| 3 |  | YouTube      | 96,0% | 11 |  | OLX.ua     | 47,6% |
| 4 |  | Gmail        | 91,8% | 12 |  | Нова пошта | 39,0% |
| 5 |  | Facebook     | 87,2% | 13 |  | AliExpress | 38,3% |
| 6 |  | Google Maps  | 82,9% | 14 |  | WhatsApp   | 37,7% |
| 7 |  | Приват24     | 78,7% | 15 |  | monobank   | 37,4% |
| 8 |  | Fb Messenger | 73,6% |    |                                                                                    |            |       |

CMeter Mobile: Охоплення в %, березень 2020, мобільні користувачі смартфонів Android 16-55 років, міста 50K+

**KANTAR**

Рис. 2.4 Рейтинг мобільних додатків в Україні станом на березень 2020 року

## Рейтинг мобільних додатків Січень 2021

|    |                                                                                     |              |       |    |                                                                                      |              |       |
|----|-------------------------------------------------------------------------------------|--------------|-------|----|--------------------------------------------------------------------------------------|--------------|-------|
| 1  |  | Chrome       | 98,2% | 11 |  | OLX.ua       | 50,8% |
| 2  |  | YouTube      | 97,2% | 12 |  | Нова пошта   | 50,5% |
| 3  |  | Viber        | 96,8% | 13 |  | My Vodafone  | 44,4% |
| 4  |  | Gmail        | 95,8% | 14 |  | WhatsApp     | 42,2% |
| 5  |  | Google Maps  | 86,0% | 15 |  | AliExpress   | 42,1% |
| 6  |  | Facebook     | 85,7% | 16 |  | Мой Киевстар | 38,4% |
| 7  |  | Приват24     | 77,5% | 17 |  | monobank     | 34,5% |
| 8  |  | Telegram     | 76,8% | 18 |  | Дія          | 29,7% |
| 9  |  | Fb Messenger | 72,3% | 19 |  | Rozetka      | 27,7% |
| 10 |  | Instagram    | 69,0% | 20 |  | Shazam       | 26,7% |

CMeter Mobile: Охоплення в %, січень 2021, мобільні користувачі смартфонів Android 16-55 років, міста 50K+

**KANTAR**

Рис. 2.5 Рейтинг мобільних додатків в Україні станом на січень 2021 року

Отже, Telegram є найперспективнішим месенджером для впровадження чат-бота, адже кількість його користувачів постійно зростає. Інші переваги Telegram:

- Наявність Telegram Bot API – HTTP-інтерфейсу для роботи з ботами;
- Просте створення і адміністрування бота за допомогою BotFather;
- Можливість використання polling;
- Можливість створення інтерактивних меню(або кнопок) в чаті;
- Швидкість доставки повідомлень;
- Перенесення облікового запису на різні пристрої(можна одночасно використовувати додаток з телефона та комп'ютера);
- Безпека;
- Зберігання всіх файлів у хмарі.

### 2.3 Вибір мови програмування

Існує багато реалізацій Telegram Bot API на різних мовах програмування.

Список бібліотек та фреймворків, які використовуються для написання чат-ботів у Telegram[14]:

Для PHP:

- Bot API PHP SDK
- NovaGram

Для Python:

- python-telegram-bot
- PyTelegramBotAPI
- AIOGram

Для Java:

- TelegramBots
- Java API

Для C#:

- Telegram.Bot

Для JavaScript:

- node-telegram-bot-api

З перерахованих варіантів була обрана PyTelegramBotAPI, оскільки дана бібліотека дозволить вдало реалізувати весь функціонал чат-бота для підтримки клієнтів медичної клініки. Відтак, обрана мова програмування – Python.

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією [15]. Python підтримує різні парадигми програмування: структурну, об'єктно-орієнтовану, функціональну, імперативну і аспектно-орієнтовану. Мова включає в себе динамічну типізацію, автоматичне керування пам'яттю, повне самоспостереження, механізм обробки винятків, підтримку багатопотокових обчислень і практичні структури даних високого рівня [16].

Переваги Python[17]:

- відкрита розробка;
- досить простий у вивченні, особливо на початковому етапі;
- особливості синтаксису стимулюють програміста писати код, який легко читати;
- надає кошти швидкого прототипування і динамічної семантики;
- безліч корисних бібліотек і розширень мови можна легко використовувати в своїх проектах завдяки гранично уніфікованого механізму імпорту та програмним інтерфейсами;
- механізми модульності добре продумані і можуть бути легко використані.

Недоліки Python:

- не дуже вдала підтримка багатопоточності;
- на Python створено не так вже й багато якісних програмних проектів у порівнянні з іншими універсальними мовами програмування, наприклад, з Java;



- відсутність комерційної підтримки засобів розробки (хоча ця ситуація з часом змінюється);
- початкова обмеженість коштів для роботи з базами даних

Перевагами створення чат-бота на мові програмування Python є:

- широкий спектр можливостей;
- відносна простота написання коду;
- наявність великої кількості документації;
- великий вибір бібліотек, які допоможуть створити хороший сервіс

## 2.4 Вибір середовища програмування

Для написання коду потрібна інтегроване середовище розробки – комплексне програмне рішення для розробки програмного забезпечення, яке зазвичай складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм [18].

Для проектування чат-бота для підтримки клієнтів медичної клініки середовищем розробки був обраний PyCharm Community Edition.

PyCharm має зручний редактор коду з усіма корисними функціями: підсвічуванням синтаксису, автоматичним форматуванням, доповненням і відступами. PyCharm дозволяє перевіряти версії інтерпретатора мови на сумісність, а також використовувати шаблони коду [19].

Переваги PyCharm [20]:

- автодоповнення коду;
- гарні можливості для налагодження програми;
- зручна робота з серверами контролю версії (СКВ);
- можливість роботи з базами даних (автодоповнення SQL);
- можливість роботи з кодом, який знаходиться на віддаленій машині (та його налагодження);
- автоматичне виправлення залежностей та ін.

Недоліки PyCharm:

- бувають різні дрібні баги при налагодженні;
- мова інтерфейсу – англійська.

## 2.5 Вибір бази даних для чат-бота

Для зберігання даних, які необхідні для коректної роботи чат-бота, була обрана база даних MySQL.

MySQL – вільна реляційна система керування базами даних (СКБД), яка працює як сервер для забезпечення багатокористувацького доступу до великої кількості баз даних [21].

База MySQL організована у вигляді пов'язаних між собою таблиць. Відкритий вихідний код можна змінювати. Ви можете самостійно встановити програму. Якщо вивчити вільний код, можна налаштувати його під свої потреби. Коли знадобиться розширена підтримка, потрібно придбати ліцензовану версію.

Комп'ютери, які запускають програму СКБД, називаються клієнтами. Для отримання даних вони підключаються до сервера. MySQL сервер - це багатопоточний обслуговуючий пристрій для роботи з базами даних. Така система називається «клієнт-сервер» [22] (рис. 2.6).

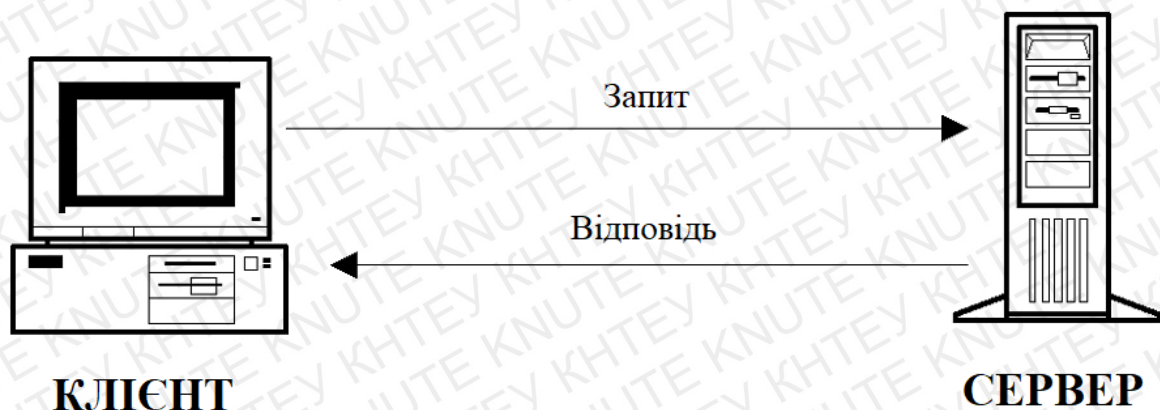


Рис. 2.6 Базова структура системи «клієнт-сервер»

Спочатку система створювалася для обмеженого використання. Сьогодні вона сумісна з такими платформами, як Microsoft Windows, Linux, macOS і Ubuntu. Великі додатки (Google, Facebook, YouTube, Twitter і Yahoo!) використовують для зберігання своїх даних MySQL.

Перевагами MySQL є:

- Простота у встановленні та використанні;
- Можливість змінювати вихідний код;
- Кількість рядків у таблицях може досягати 50 млн;
- Продуктивність – система підтримує безліч моделей кластерних серверів;
- Висока швидкість виконання команд;
- Стандартизація;
- Безпека – система доступу до облікових записів дозволяє гарантувати збереження баз даних і високий клас їх безпеки. Доступно шифрування пароля і перевірка на основі хоста.

Принцип роботи MySQL. Клієнт підключається до сервера по мережі. Він робить запит з інтерфейсу користувача (GUI). Якщо серверу зрозумілі інструкції, він видає інформацію.

У середовищі MySQL відбуваються такі процеси:

- створюється база даних MySQL, визначається ставлення таблиці;
- клієнти роблять запити, використовуючи команди SQL;
- сервер відповідає запитаній інформацією, яка відправляється клієнтові.

Популярними інтерфейсами є SequelPro, DBVisualizer, MySQL WorkBench і Navicat DB Admin Tool.

Для проектування бази даних для чат-бота був обраний інтерфейс MySQL WorkBench.

Для спрощення роботи з базою даних був обраний клієнт HeidiSQ – інструмент управління базами даних з відкритим вихідним кодом, який підтримує MariaDB, MySQL, MS SQL, PostgreSQL і SQLite [23]. Він дозволяє



переглядати і редагувати дані, створювати і змінювати таблиці, уявлення, процедури, тригери і події в розкладі. Він також надає можливість експортувати структури і дані в SQL-файли, буфер обміну або переносити на інші сервера [24].

## 2.6 Функціональні вимоги до чат-бота

Чат-бот для підтримки клієнтів медичної клініки має два модулі:

- 1) Модуль роботи з Telegram
- 2) Модуль роботи з базою даних MySQL

Функціональні вимоги до модулів чат-бота для підтримки клієнтів медичної клініки наведені у табл. 2.1.

Таблиця 2.1

### Функціональні вимоги до модулів чат-бота для підтримки клієнтів медичної клініки

| Модуль                            | Функціональна вимога                                                                                                             | Пріоритет |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------|-----------|
| Модуль роботи з Telegram          | Система є зв'язком між користувачем та системними даними                                                                         | Високий   |
|                                   | Система збирає дані від запитів користувачів за допомогою сервісу Telegram Bot API та обробляє їх                                | Високий   |
|                                   | Система використовує дані, надані сервісом Telegram Bot API, і на основі цього робить запит до модуля роботи з базою даних MySQL | Високий   |
|                                   | Система отримує дані від модуля роботи з базою даних MySQL та опрацьовує їх                                                      | Низький   |
| Модуль роботи з базою даних MySQL | Система отримує дані від модуля роботи з Telegram                                                                                | Високий   |
|                                   | Система опрацьовує дані отримані від модуля роботи з Telegram                                                                    | Високий   |
|                                   | Система зупиняє запис даних у разі виникнення помилки                                                                            | Високий   |
|                                   | Система робить запис у базу даних                                                                                                | Високий   |

## Висновки до розділу

Чат-бот медичної клініки є ботом для підтримки клієнтів. Сучасний користувач надає перевагу використанню програм-месенджерів над традиційними дзвінками. При цьому невід'ємною частиною взаємодії із месенджерами є використання ботів. Оскільки ключовим для розробника є кінцевий досвід експлуатації користувачем продукту, слід при розробці бота врахувати сучасні тенденції та максимізувати при цьому позитивний досвід.

Telegram є найперспективнішим месенджером для впровадження чат-бота, адже кількість його користувачів в останні роки постійно зростає. Крім цього, Telegram має багато інших переваг, серед яких наявність Telegram Bot API, можливість використання polling, швидкість доставки повідомлень та безпека. Telegram пропонує розробникам чат-ботів велику кількість корисних бібліотек на різних мовах програмування, зокрема на Python. Python, у порівнянні з іншими мовами, дозволяє спростити код програми, зробити його зрозумілим та доступним. Отже, використання мови програмування Python є вдалим рішенням для розробки чат-бота.

Для проектування чат-бота для підтримки клієнтів медичної клініки середовищем розробки був обраний PyCharm Community Edition. PyCharm має зручний редактор коду з усіма корисними функціями: підсвічуванням синтаксису, автоматичним форматуванням, доповненням і відступами.

Для зберігання даних, які необхідні для коректної роботи чат-бота, була обрана база даних MySQL. Сьогодні вона сумісна з такими платформами, як Microsoft Windows, Linux, macOS і Ubuntu. Великі додатки (Google, Facebook, YouTube, Twitter і Yahoo!) використовують для зберігання своїх даних MySQL, через зручність її використання, продуктивність, швидкість обробки даних та безпеку.



## РОЗДІЛ 3 РОЗРОБКА ЧАТ-БОТА ДЛЯ ПІДТРИМКИ ВЗАЄМОДІЇ З КЛІЄНТАМИ МЕДИЧНОЇ КЛІНІКИ

### 3.1. Створення чат-бота за допомогою BotFather

BotFather – це Telegram-бот, за допомогою якого здійснюється створення чат-ботів на платформі. Для того, щоб зареєструвати власного бота, необхідно зробити декілька простих кроків, результатом яких є отримання унікального токена авторизації, який дозволяє серверам Telegram ідентифікувати бота.

Для початку роботи з BotFather необхідно використати команду «/start». Бот висилає список команд, необхідних для створення та адміністрування чат-ботів (рис. 3.1).

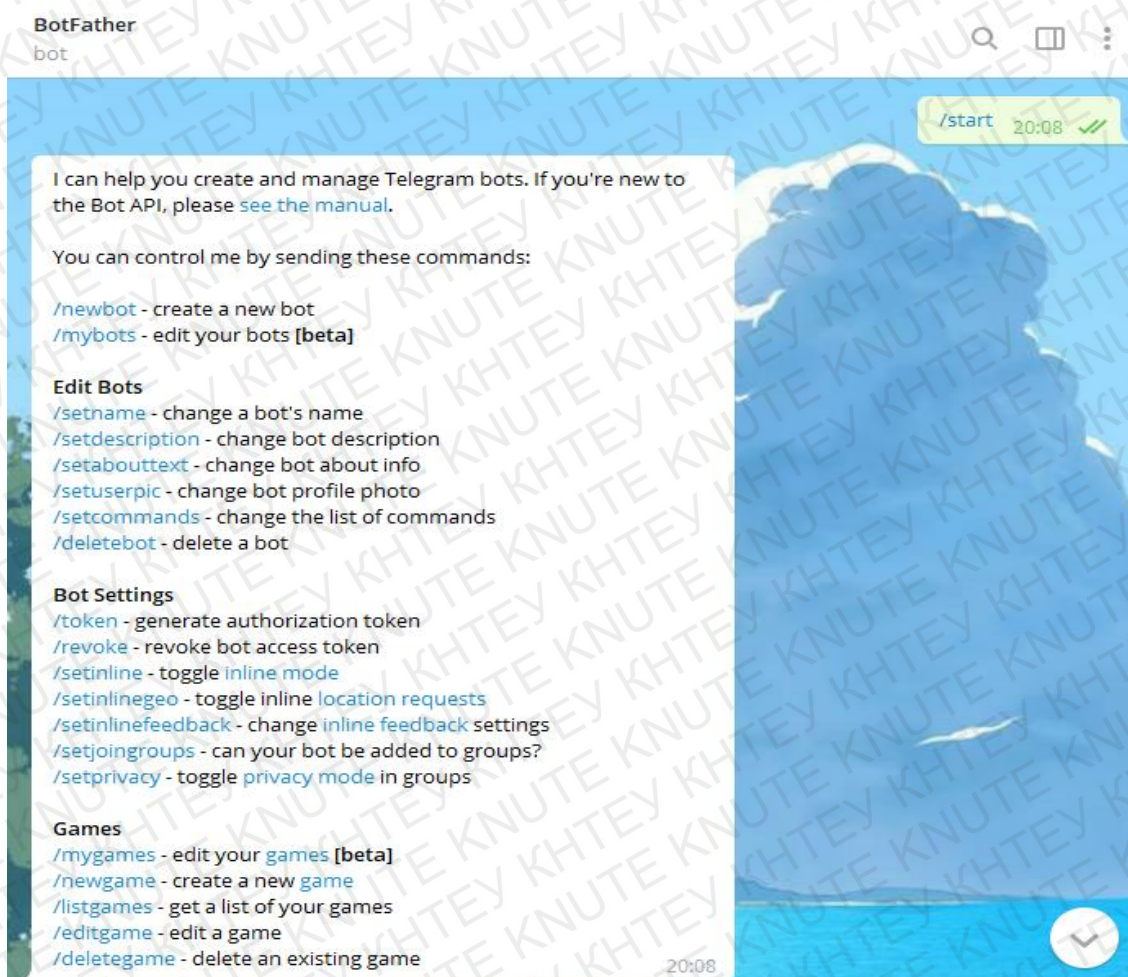


Рис. 3.1 Початок роботи з BotFather



Для того, щоб створити власного чат-бота, необхідно виконати команду «/newbot». BotFather запросить вказати:

- Ім'я бота – ім'я, яке відображатиметься зверху над вікном діалогу. Користувач постійно бачить ім'я бота, тому повинно розкривати сутність програми. Найкращий варіант – вписати назву клініки та додати «бот», щоб користувач мав змогу одразу зрозуміти, з ким ведеться чат.
- Ім'я користувача – ім'я, яке потрібне для формування посилання на чат-бота. Воно повинно містити від 5 до 32 символів(допускаються тільки латинські символи, цифри та підкреслення). Ім'я користувача обов'язково повинно закінчуватись на «bot». Має сенс зробити однакові(або схожі) імена для бота та користувача. Незважаючи на те, що клієнт не бачить ім'я користувача постійно, при посиланні на чат-бота у нього будуть виникати певні асоціації.

На рис 3.2. представлений фрагмент діалогу з BotFather про отримання токена.

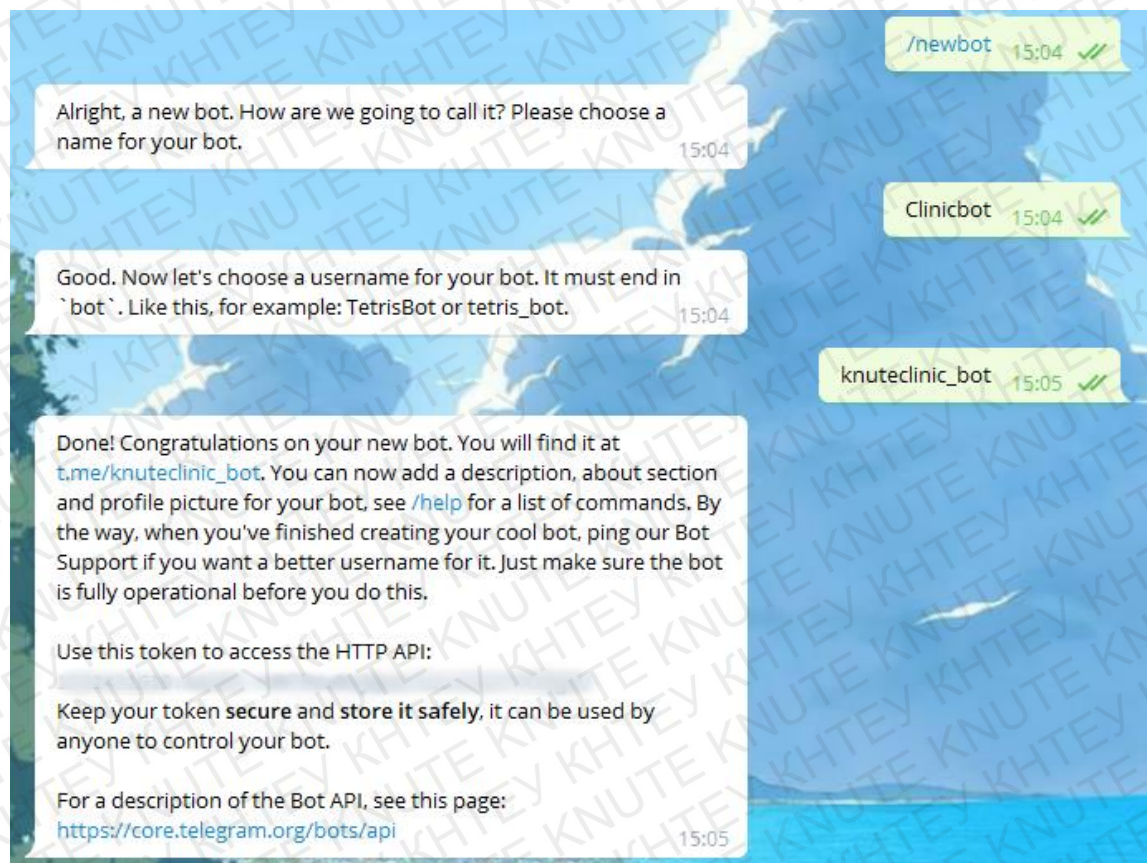


Рис. 3.2 Процес отримання токена від BotFather

BotFather попереджає: “Забезпечуйте безпеку свого токена та зберігайте його в надійному місці, бо він може бути використаний ким завгодно для управління вашим ботом”.

Botfather використовується не лише для створення ботів, але й для їх адміністрування. Список команд, які потрібні для змін налаштувань бота, наведений у табл. 3.1.

Таблиця 3.1

| Команда            | Призначення                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------|
| /mybots            | Надає список створених ботів і дозволяє редагувати їх налаштування                                   |
| /setname           | Змінює ім'я бота                                                                                     |
| /setdescription    | Змінює опис бота – текст, який користувач бачить перед початком роботи з чат-ботом                   |
| /setabouttext      | Змінює інформацію про бота – текст, який можна побачити в строчці About в профілі чат-бота           |
| /setuserpic        | Змінює зображення профілю бота                                                                       |
| /setcommands       | Змінює список команд бота. Користувачі бачать ці команди як пропозиції, коли вони вводять «/» в чаті |
| /deletebot         | Видаляє бота                                                                                         |
| /token             | Генерує маркер авторизації(токен)                                                                    |
| /revoke            | Анулює токен                                                                                         |
| /setinline         | Вмикає чи вимикає режим “інлайн-бота”                                                                |
| /setinlinegeo      | Дозволяє боту використовувати дані про місцезнаходження користувача                                  |
| /setinlinefeedback | Змінює налаштування вбудованого зворотного зв'язку                                                   |
| /setjoiningroups   | Змінює налаштування під'єднання чат-бота до груп                                                     |
| /setprivacy        | Вмикає чи вимикає режим приватності – визначає, які повідомлення бот буде отримувати в групах        |

На рис. 3.3 зображено вікно інформації про Clinicbot після налаштувань бота за допомогою Botfather.



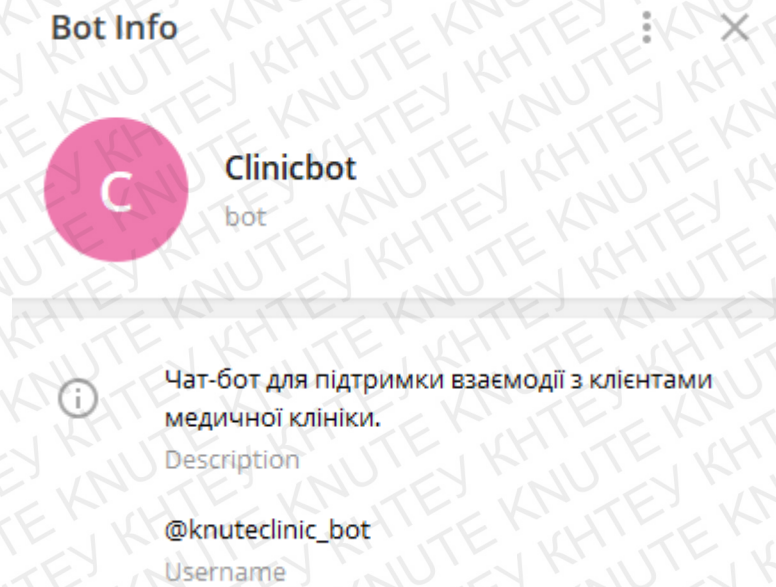


Рис. 3.3 Вікно інформації про Clinicbot після налаштувань

### 3.2 Створення з'єднання з базою MySQL

Першим кроком у створенні з'єднання з базою MySQL є підключення пакету `mysql-connector-python` – драйвера MySQL, написаного на Python, який не залежить від клієнтських бібліотек MySQL C та реалізує специфікацію DB API v2.0. Після цього потрібно імпортувати бібліотеку `mysql.connector`.

У середовищі MySQL Workbench необхідно створити MySQL Connection, у якому потрібно вказати:

- назву з'єднання;
- метод з'єднання;
- ім'я хосту та порту;
- ім'я користувача;
- пароль.

На рис. 3.4 зображене налаштування MySQL Connection у MySQL Workbench.



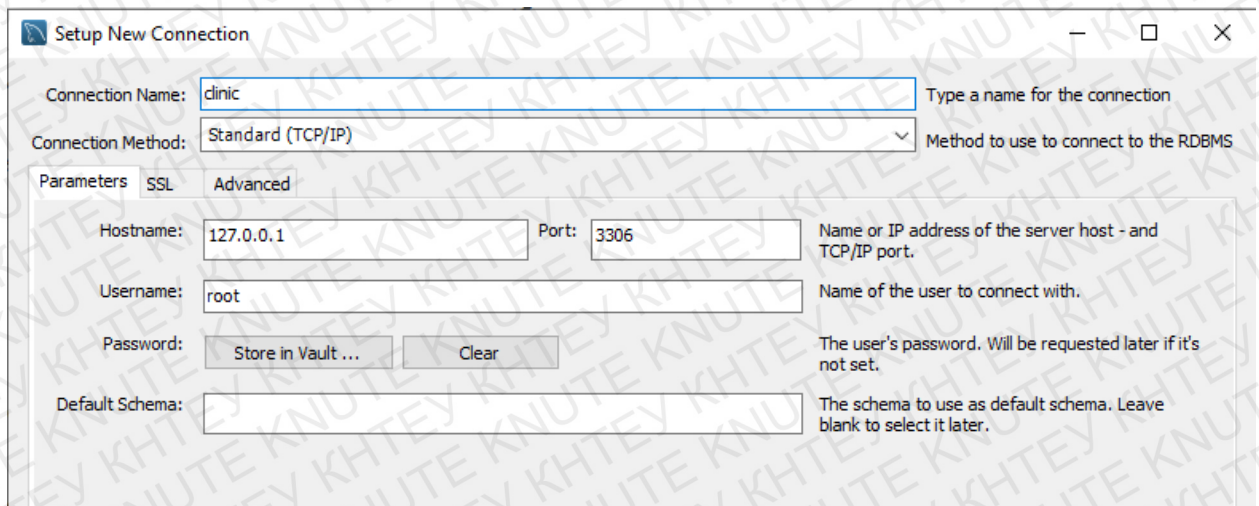


Рис. 3.4 Налаштування MySQL Connection у MySQL Workbench

Для створення з'єднання необхідно скористатися функцією `mysql.connector.connect()`. Потрібно вказати хост, ім'я користувача та пароль. При необхідності можливо вказати інший порт. На рис. 3.5 показано з'єднання з базою MySQL у PyCharm.

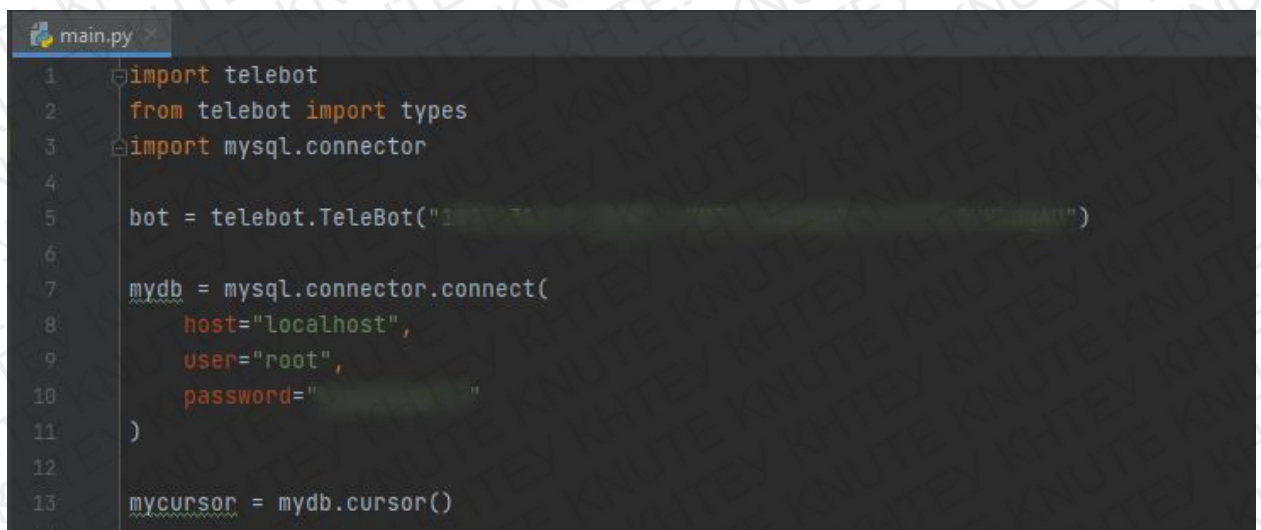


Рис. 3.5 З'єднання з базою MySQL

### 3.3 Розробка структури чат-бота

Розроблюваний чат-бот базується на заздалегідь визначених і запрограмованих алгоритмах спілкування з користувачами системи, на відміну від чат-ботів, що базуються на методах штучного інтелекту. Отже, проектування вдалої структури грає дуже велику роль в розробці чат-бота,

оскільки від неї цілком залежить майбутній функціонал додатка, який повинен повністю задовольняти користувачів.

Виходячи з функцій, які виконує чат-бот, його структуру можна розділити на два основні блоки:

- 1) Інформаційний блок;
- 2) Блок запису на прийом.

Інформаційний блок реалізований у вигляді багаторівневого кнопкового меню, яке визивається командою /start. Головними перевагами кнопкової системи є:

- Зручність користування;
- Скорочення часу взаємодії користувача з системою;
- Збільшення кількості запитів, як наслідок скорочення часу взаємодії з системою

Інформаційний блок надає користувачеві наступні дані:

- 1) Загальна інформація про клініку;
- 2) Інформація про лікарів клініки, їх графік роботи;
- 3) Інформація про послуги, що надаються, і ціни на них;
- 4) Адреси клінік, контактні номери, графік роботи;
- 5) Посилання на команди, які здійснюють запис на прийом до певного лікаря.

На рис. 3.6 зображена загальна структура інформаційного блоку чат-бота для підтримки клієнтів медичної клініки.



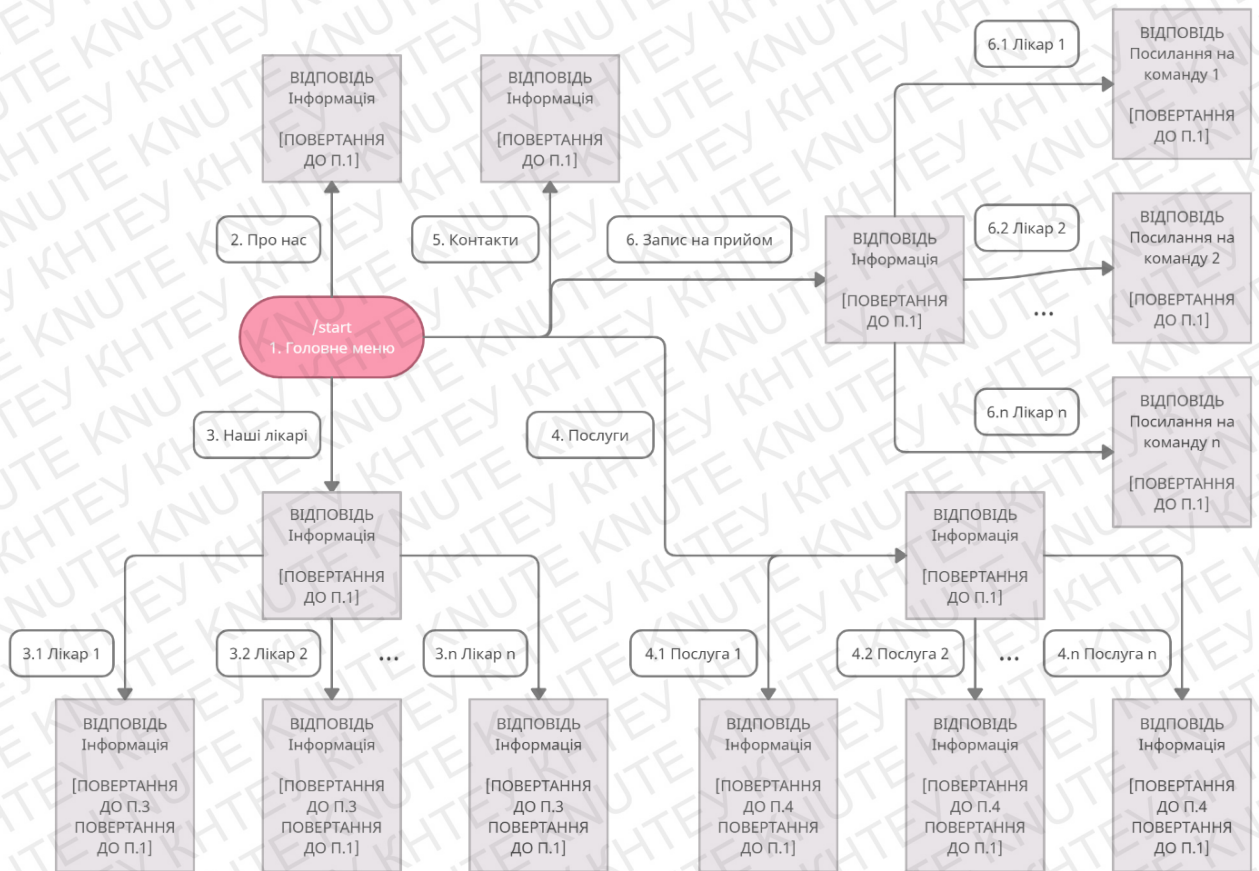


Рис. 3.6 Структура інформаційного блоку чат-бота

Команда /start викликає повідомлення «Вас вітає клініка Н!», до якого прикріплена інлайн-клавіатура. Інлайн-клавіатура являє собою об'єкт `InlineKeyboardMarkup`, а кожна інлайн-кнопка - це об'єкт `InlineKeyboardButton`. Існує три типи інлайн-кнопок: URL-кнопки, Callback-кнопки і Switch-кнопки. Для зручної навігації по меню було використано Callback-кнопки. Вони дозволяють виконувати довільні дії при їх натисканні. Все залежить від того, які параметри кожна кнопка в собі несе. Відповідно, всі натискання будуть приводити до відправки боту об'єкта `CallbackQuery`, який містить поле `data`, в якому написаний деякий рядок, закладений в кнопку. Зміст інлайн-клавіатури першого рівня описаний у табл. 3.2.



Таблиця 3.2

## Інлайн-клавіатура першого рівня

| Кнопка        | text                   | callback_data | Призначення                                                                                               |
|---------------|------------------------|---------------|-----------------------------------------------------------------------------------------------------------|
| first_button  | «Про нас»              | first         | При натисканні відкриває повідомлення з інформацією про клініку                                           |
| second_button | «Наші лікарі»          | second        | При натисканні відкриває список лікарів(інлайн-клавіатуру другого рівня)                                  |
| third_button  | «Послуги»              | third         | При натисканні відкриває список послуг(інлайн-клавіатуру другого рівня)                                   |
| fourth_button | «Контакти»             | fourth        | При натисканні відкриває повідомлення з інформацією про адресу клініки, контактні номери та графік роботи |
| fifth_button  | «Записатися на прийом» | fifth         | При натисканні відкриває список лікарів для запису на прийом(інлайн-клавіатуру другого рівня)             |

На рис. 3.7 зображений фрагмент коду з реалізацією першого рівня меню чат-бота.

```
@bot.message_handler(commands=['start'])
def any_msg(message):
    keyboardmain = types.InlineKeyboardMarkup(row_width=2)
    first_button = types.InlineKeyboardButton(text="Про нас", callback_data="first")
    second_button = types.InlineKeyboardButton(text="Наші лікарі", callback_data="second")
    third_button = types.InlineKeyboardButton(text="Послуги", callback_data="third")
    fourth_button = types.InlineKeyboardButton(text="Контакти", callback_data="fourth")
    fifth_button = types.InlineKeyboardButton(text="Записатися на прийом", callback_data="fifth")
    keyboardmain.add(first_button, second_button, third_button, fourth_button, fifth_button)
    bot.send_message(message.chat.id, "Вас вітає клініка Н!", reply_markup=keyboardmain)
```

Рис. 3.7 Фрагмент коду з реалізацією першого рівня меню чат-бота

Другий та третій рівні інлайн-меню реалізовані за допомогою операторів if-elif. Зміст інлайн-клавіатури другого рівня описаний у табл. 3.3.

Таблиця 3.3

## Інлайн-клавіатура другого рівня

| if-elif<br>callback_data | Кнопка     | text               | callback_data | Призначення                                                                  |
|--------------------------|------------|--------------------|---------------|------------------------------------------------------------------------------|
| first                    | backbutton | «Назад до<br>меню» | mainmenu      | Повертає до<br>головного меню                                                |
| second                   | rele1      | Лікар1             | likar1        | При натисканні<br>відкриває<br>повідомлення з<br>інформацію про<br>лікаря 1  |
|                          | rele2      | Лікар2             | likar2        | При натисканні<br>відкриває<br>повідомлення з<br>інформацію про<br>лікаря 2  |
|                          | ...        |                    |               |                                                                              |
|                          | relen      | Лікарn             | likarn        | При натисканні<br>відкриває<br>повідомлення з<br>інформацію про<br>лікаря n  |
|                          | backbutton | «Назад до<br>меню» | mainmenu      | Повертає до<br>головного меню                                                |
| third                    | rele1      | Послуга1           | posluga1      | При натисканні<br>відкриває<br>повідомлення з<br>інформацію про<br>послугу 1 |

Продовження табл. 3.3

|        |            |                 |          |                                                                                      |
|--------|------------|-----------------|----------|--------------------------------------------------------------------------------------|
|        | rele2      | Послуга2        | posluga2 | При натисканні відкриває повідомлення з інформацію про послугу 2                     |
|        | rele3      | Послуга3        | posluga3 | При натисканні відкриває повідомлення з інформацію про послугу 3                     |
|        | ...        |                 |          |                                                                                      |
|        | relen      | Послуган        | poslugan | При натисканні відкриває повідомлення з інформацію про послугу n                     |
|        | backbutton | «Назад до меню» | mainmenu | Повертає до головного меню                                                           |
| fourth | backbutton | «Назад до меню» | mainmenu | Повертає до головного меню                                                           |
| fifth  | rele1      | Лікар1          | app1     | При натисканні відкриває повідомлення з командою, що потрібна для запису до лікаря 1 |



Продовження табл. 3.3

|  |            |                 |          |                                                                                      |
|--|------------|-----------------|----------|--------------------------------------------------------------------------------------|
|  | rele2      | Лікар2          | app2     | При натисканні відкриває повідомлення з командою, що потрібна для запису до лікаря 2 |
|  | ...        |                 |          |                                                                                      |
|  | relen      | Лікарn          | appn     | При натисканні відкриває повідомлення з командою, що потрібна для запису до лікаря n |
|  | backbutton | «Назад до меню» | mainmenu | Повертає до головного меню                                                           |

На рис. 3.8 зображений фрагмент коду з реалізацією другого рівня меню чат-бота.

```

if call.data == "first":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад до меню", callback_data="mainmenu")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id, text="Опис лікарні", reply_markup=keyboard)

elif call.data == "second":
    keyboard = types.InlineKeyboardMarkup(row_width=1)
    rele1 = types.InlineKeyboardButton(text="Лікар1", callback_data="likar1")
    rele2 = types.InlineKeyboardButton(text="Лікар2", callback_data="likar2")
    backbutton = types.InlineKeyboardButton(text="Назад до меню", callback_data="mainmenu")
    keyboard.add(rele1, rele2, backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id, text="Наші лікарі", reply_markup=keyboard)

```

Рис. 3.8 Фрагмент коду з реалізацією другого рівня меню чат-бота

Зміст інлайн-клавіатури третього рівня описаний у табл. 3.4.

Таблиця 3.4

## Інлайн-клавіатура третього рівня

| if-elif<br>callback_data                             | Кнопка     | text    | callback_data | Призначення                                    |
|------------------------------------------------------|------------|---------|---------------|------------------------------------------------|
| likar1<br>likar2<br>...<br>likar n                   | backbutton | «Назад» | second        | Повертає до<br>списку<br>лікарів               |
| posluga1<br>posluga2<br>posluga3<br>...<br>posluga n | backbutton | «Назад» | third         | Повертає до<br>списку<br>послуг                |
| app1<br>app2<br>...<br>appn                          | backbutton | «Назад» | fifth         | Повертає до<br>списку<br>лікарів для<br>запису |

На рис. 3.8 зображений фрагмент коду з реалізацією другого рівня меню чат-бота.

```

elif call.data == "posluga1":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад", callback_data="third")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id, text="Опис послуги\n\nЦіна послуги", reply_markup=keyboard)

elif call.data == "posluga2":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад", callback_data="third")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id, text="Опис послуги\n\nЦіна послуги", reply_markup=keyboard)

elif call.data == "posluga3":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад", callback_data="third")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id, text="Опис послуги\n\nЦіна послуги", reply_markup=keyboard)

```

Рис. 3.9 Фрагмент коду з реалізацією третього рівня меню чат-бота



Для запису на прийом користувачеві будуть запропоновані спеціальні команди, посилання на які він отримає у ході взаємодії з інформаційним блоком. Кожному лікарю присвоєна своя власна окрема команда: це гарантує зручний запис даних у базу.

Після виконання команди «/app\_líkar»(яка буде змінюватися в залежності від того, якого лікаря обрав користувач для запису) чат-бот починає формування клієнтської заявки, яка містить наступні дані про користувача:

- ПІБ пацієнта;
- Дата народження пацієнта;
- Номер телефону пацієнта;
- Дата та час прийому пацієнта;
- Короткий опис скарги.

Після формування клієнтської заявки, чат-бот інформує користувача про успішність реєстрації(користувач зареєстрований/не зареєстрований, з вказанням причини помилки) та пропонує команду «/start» для повернення до головного меню.

На рис. 3.10 зображена блох-схема клієнтського запису на прийом до лікаря.



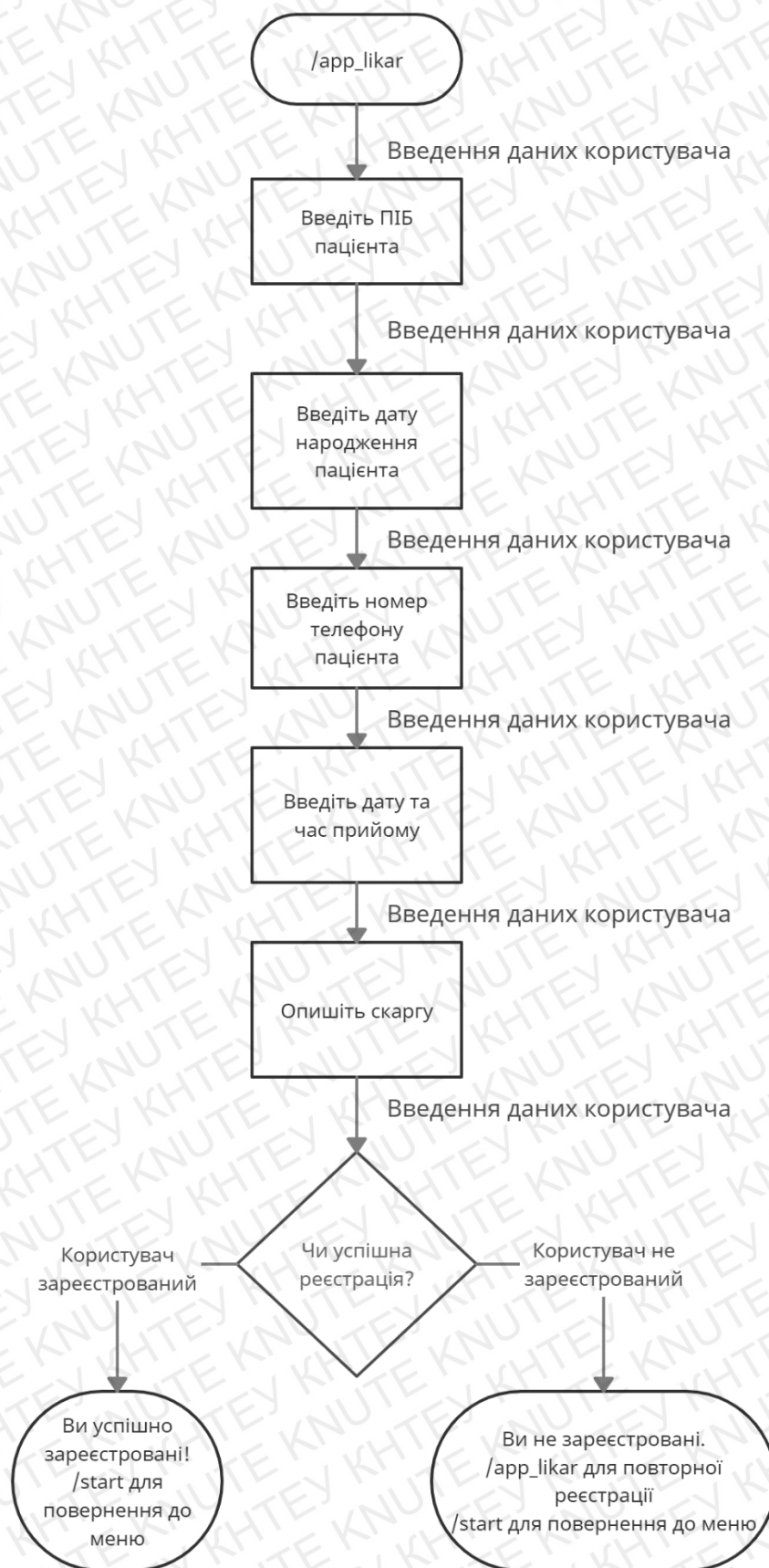


Рис. 3.10 Блох-схема запису на прийом

### 3.4 Розробка структури бази даних

При реалізації функціональних можливостей чат-боту програма буде оперувати інформацією, яка буде надіслана йому серверами Telegram.

MySQL зберігає дані у вигляді таблиць. Для автоматизації запису на прийом таблиці, які зберігають інформації про користувача, повинні бути окремими для кожного лікаря. Кількість таблиць, їх назви і зміст можуть змінюватися згідно з потреб клініки. Загальна структура таблиці для запису до лікаря представлена у табл. 3.5.

Таблиця 3.5

| Назва поля    | Тип                                  | Призначення                                           |
|---------------|--------------------------------------|-------------------------------------------------------|
| id            | INT<br>AUTO_INCREMENT<br>PRIMARY KEY | Ідентифікатор,<br>автоматично<br>згенерований MySQL   |
| pat_name      | VARCHAR(255)                         | ПІБ пацієнта                                          |
| date_of_birth | VARCHAR(255)                         | Дата народження<br>пацієнта                           |
| phone_number  | VARCHAR(255)                         | Номер телефону<br>пацієнта                            |
| receipt_time  | VARCHAR(15) NOT<br>NULL UNIQUE       | Дата та час прийому                                   |
| complaint     | VARCHAR(255)                         | Короткий опис скарги                                  |
| user_id       | INT(15)                              | Унікальний<br>ідентифікатор<br>користувача в Telegram |

Дата та час прийому(receipt\_time) мають тип VARCHAR(15) NOT NULL UNIQUE – це означає, що ці користувацькі дані є унікальними, отже можуть бути записані у таблицю лише в одному екземплярі. Це забезпечує те, що користувачі не зможуть зробити запис до одного й того ж самого лікаря на один й той же самий час. Якщо користувач спробує зробити запис на таку дату та час, які вже присутні в таблиці, йому прийде повідомлення про помилку при реєстрації. Чат-бот попросить користувача обрати іншу дату або час прийому для успішної реєстрації клієнтського запиту.

Для збереження інформації про користувача створена таблиця «користувачі». Вона потрібна для фіксації кількості активних юзерів, що користуються чат-ботом у Telegram. Користувач зі свого аккаунта може записувати на прийом безліч пацієнтів необмежену кількість разів. Це може бути зручним, наприклад, у ситуації, коли один з членів родини реєструє всіх своїх родичів(особливо неповнолітніх та нездатних зареєструватися самостійно з різних причин). Інформація про кожного нового користувача буде зберігатися в базі даних у таблиці «користувачі». Дані про кількість активних юзерів можуть бути використані для дослідження динаміки розповсюдження чат-бота та проведення певних заходів для підвищення частоти застосування продукту. У табл. 3.6 наведена структура таблиці «користувачі».

Таблиця 3.6

| Назва поля | Тип                                  | Призначення                                         |
|------------|--------------------------------------|-----------------------------------------------------|
| id         | INT<br>AUTO_INCREMENT<br>PRIMARY KEY | Ідентифікатор,<br>автоматично<br>згенерований MySQL |
| first_name | VARCHAR(255)                         | Ім'я користувача в<br>Telegram                      |
| last_name  | VARCHAR(255)                         | Фамілія користувача в<br>Telegram                   |



Продовження табл. 3.6

|                  |                |                                                       |
|------------------|----------------|-------------------------------------------------------|
| telegram_user_id | INT(15) UNIQUE | Унікальний<br>ідентифікатор<br>користувача в Telegram |
|------------------|----------------|-------------------------------------------------------|

На рис. 3.11 представлений фрагмент коду зі створення таблиць у базі даних clinic.

```

mycursor.execute("CREATE DATABASE clinic")

mycursor.execute("CREATE TABLE лікар1 (pat_name VARCHAR(255), date_of_birth VARCHAR(255), phone_number VARCHAR(255),
        receipt_time VARCHAR(15) NOT NULL UNIQUE, complaint VARCHAR(255))")

mycursor.execute("ALTER TABLE лікар1 ADD COLUMN (id INT AUTO_INCREMENT PRIMARY KEY)")

mycursor.execute("ALTER TABLE лікар1 ADD COLUMN (user_id INT(11))")

mycursor.execute("CREATE TABLE користувачі (id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(255),
        last_name VARCHAR(255), telegram_user_id INT(11) UNIQUE)")

```

Рис. 3.11 Створення таблиць у базі даних clinic

На рис. 3.12 представлена таблиця «лікар1» у базі даних clinic.

| # | Имя           | Тип данных | Длина/Знач... | Беззна...                | Разреш...                           | Zero fill                | По умолчанию       | Комментарий |
|---|---------------|------------|---------------|--------------------------|-------------------------------------|--------------------------|--------------------|-------------|
| 1 | id            | INT        | 10            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | AUTO_INCREME...    |             |
| 2 | pat_name      | VARCHAR    | 255           | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL               |             |
| 3 | date_of_birth | VARCHAR    | 255           | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL               |             |
| 4 | phone_number  | VARCHAR    | 255           | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL               |             |
| 5 | receipt_time  | VARCHAR    | 15            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | Нет значения по... |             |
| 6 | complaint     | VARCHAR    | 255           | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL               |             |
| 7 | user_id       | INT        | 10            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL               |             |

Рис. 3.12 Таблица «лікар1» у базі даних clinic

На рис. 3.13 представлена таблиця «користувачі» у базі даних clinic.

| # | Имя              | Тип данных | Длина/Знач... | Беззна...                | Разреш...                           | Zero fill                | По умолчанию    | Комментарий |
|---|------------------|------------|---------------|--------------------------|-------------------------------------|--------------------------|-----------------|-------------|
| 1 | id               | INT        | 10            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | AUTO_INCREME... |             |
| 2 | first_name       | VARCHAR    | 255           | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL            |             |
| 3 | last_name        | VARCHAR    | 255           | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL            |             |
| 4 | telegram_user... | INT        | 10            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL            |             |

Рис 3.13 Таблица «користувачі» у базі даних clinic

### 3.5 Оцінка отриманих результатів проектування

Для того, щоб оцінити отримані результати проектування, необхідно здійснити тестування можливих сценаріїв.

Першим буде виконаний тест запуску головного меню чат-бота. Він описаний у табл. 3.7.

Таблиця 3.7

| Тест                 |                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------|
| Стан системи         | Відкритий діалог с чат-ботом                                                                 |
| Дії                  | Відправлення повідомлення з командою «/start»                                                |
| Очікуваний результат | Привітання від чат-бота. Відкривається головне меню у вигляді прикріпленої інлайн-клавіатури |

Результат тесту продемонстрований на рис. 3.14

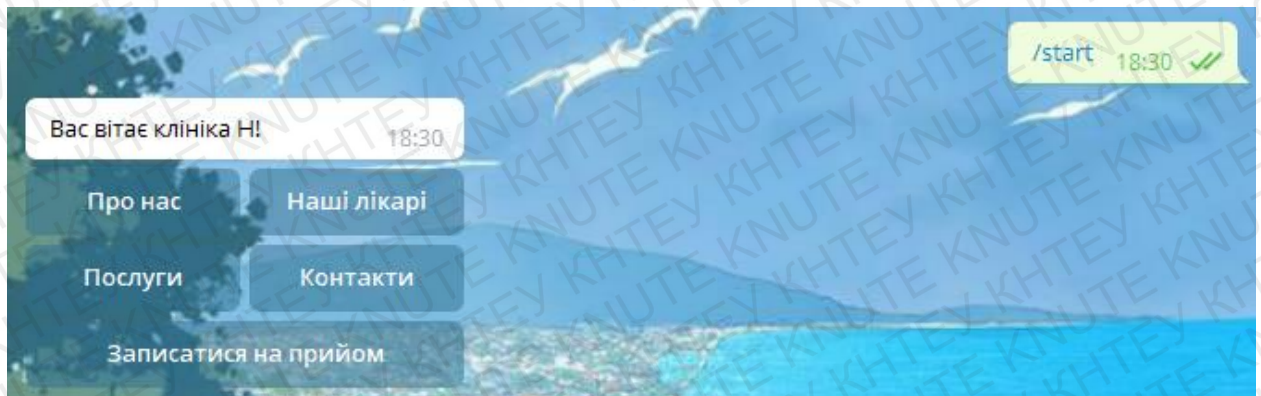


Рис. 3.14 Результат виконання команди «/start»

Наступний тест – отримання загальної інформації про клініку(табл.3.8).

Таблиця 3.8

| Тест         |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| Стан системи | Розпочато діалог с чат-ботом. Відкрите головне меню у вигляді прикріпленої інлайн-клавіатури |
| Дії          | Натискання кнопки «Про нас»                                                                  |



|                               |                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------|
| Очікуваний результат          | Чат-бот надіслає загальну інформацію про клініку                                                    |
| Стан системи після тестування | Відкривається другий рівень інлайн-клатвіатури. З'являється кнопка для повернення до головного меню |

Результат тесту продемонстрований на рис. 3.15



Рис. 3.15 Результат натискання кнопки «Про нас»

Наступним кроком є перевірка того, як чат-бот надає інформацію про послуги(табл. 3.9)

Таблиця 3.9

| Тест                          |                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------|
| Стан системи                  | Відкрите головне меню у вигляді прикріпленої інлайн-клатвіатури                                     |
| Дії                           | Натискання кнопки «Послуги»                                                                         |
| Очікуваний результат          | Чат-бот надіслає список послуг, що надаються                                                        |
| Стан системи після тестування | Відкривається другий рівень інлайн-клатвіатури. З'являється кнопка для повернення до головного меню |

Результат тесту продемонстрований на рис. 3.16





Рис. 3.16 Результат натискання кнопки «Послуги»

Тестування запису на прийом є одним з найважливіших пунктів тестування системи. Для того, щоб здійснити запис, спочатку необхідно отримати спеціальну команду, яка починає опитування. Після того, як користувач натискає кнопку «Записатися на прийом», він отримує список можливих для запису лікарів. У табл. 3.10 описаний тест отримання команди для запису на прийом.

Таблиця 3.10

| Тест                          |                                                                                                  |
|-------------------------------|--------------------------------------------------------------------------------------------------|
| Стан системи                  | Відкритий другий рівень інлайн-меню. Відкритий список лікарів, які доступні для запису на прийом |
| Дії                           | Натискання кнопки «Лікар1»                                                                       |
| Очікуваний результат          | Отримання команди для початку запису до Лікаря1                                                  |
| Стан системи після тестування | Відкривається третій рівень інлайн-меню.<br>Доступна кнопка для повернення назад до меню         |

Результат тесту продемонстрований на рис. 3.17

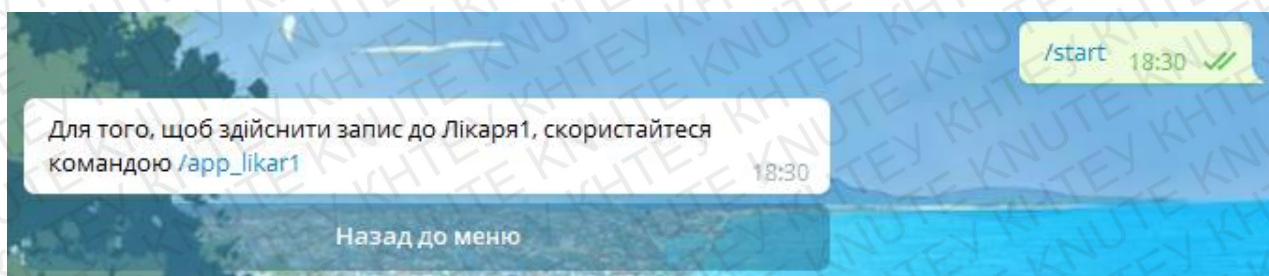


Рис. 3.17 Отримання команди для запису на прийом до Лікаря1

Після отримання необхідної команди, користувач може починати реєстрацію. У табл. 3.11 описаний тест реєстрації користувача з введеним коректних даних(дата та час прийому не зайняті).

Таблиця 3.11

| Тест                          |                                                                                                                         |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Стан системи                  | Відкритий третій рівень інлайн-меню.<br>Користувачеві запропонована команда для запису на прийом до Лікаря1             |
| Вихідні дані                  | ПІБ пацієнта, дата народження пацієнта, номер телефону пацієнта, дата та час прийому, короткий опис скарги              |
| Дії                           | Натискання команди «/app_likar1»                                                                                        |
| Очікуваний результат          | Чат-бот збирає дані, необхідні для реєстрації пацієнта. По завершенню присилає повідомлення «Ви успішно зареєстровані!» |
| Стан системи після тестування | Відкрите повідомлення з командою для переходу до головного меню                                                         |

Діалог із користувачем, проведений у тесті, показаний на рис. 3.18.



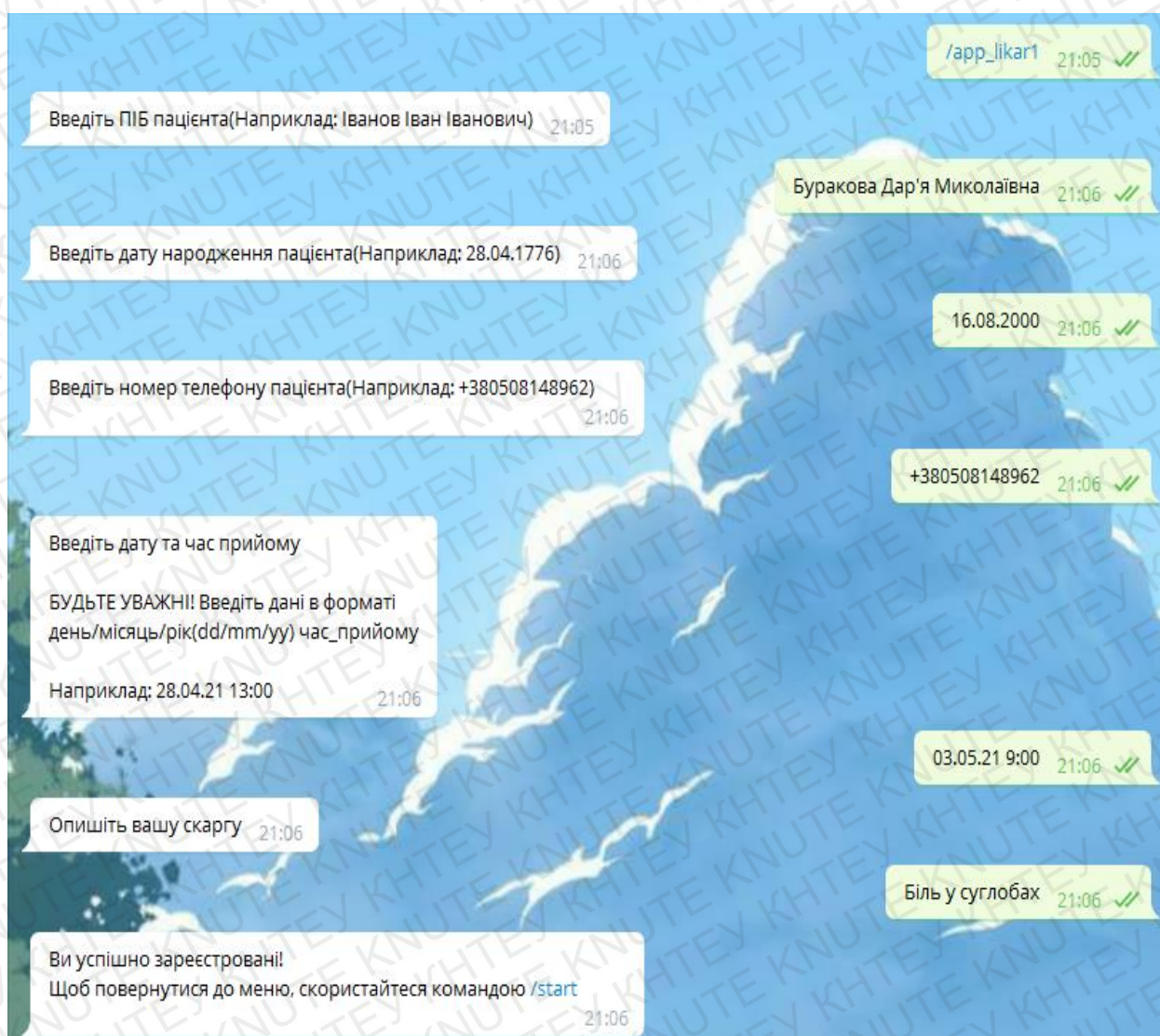


Рис. 3.18 Формування заявки пацієнта

Після того, як пацієнт надав свої дані чат-боту, вони записуються та зберігаються у базі даних MySQL. На рис. 3.19 продемонстрований запис запиту користувача у таблицю «лікар1».

clinic.likar1: 1 строк (приблизительно) >> Далее Показать все | Сортировка (1) Столбцы (7/7) Фильтр

| id | pat_name                  | date_of_birth | phone_number  | receipt_time  | complaint       | user_id     |
|----|---------------------------|---------------|---------------|---------------|-----------------|-------------|
| 1  | Буракова Дар'я Миколаївна | 16.08.2000    | +380972790021 | 03.05.21 9:00 | Біль у суглобах | 427 717 304 |

Рис. 3.19 Збереження запиту користувача у базі даних

Необхідно протестувати, як система буде реагувати на спробу користувача обрати вже зайняті дату та час прийому(табл. 3.12).



Таблиця 3.12

| Тест                          |                                                                                                                                              |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Стан системи                  | Відкритий третій рівень інлайн-меню.<br>Користувачеві запропонована команда для запису на прийом до Лікаря1                                  |
| Вихідні дані                  | ПІБ пацієнта, дата народження пацієнта, номер телефону пацієнта, дата та час прийому(які вже є в системі), короткий опис скарги              |
| Дії                           | Натискання команди «/app_likar1»                                                                                                             |
| Очікуваний результат          | Чат-бот збирає дані, необхідні для реєстрації пацієнта. По завершенню присилає повідомлення про відміну реєстрації та вказує причину помилки |
| Стан системи після тестування | Відкрите повідомлення з командами для повторної реєстрації та для переходу до головного меню                                                 |

Тестування такого сценарію є вкрай необхідним через те, що правильна реєстрація пацієнта забезпечує вправну роботу автоматизованої системи. Перевірка коректного введення даних є обов'язковою складовою програми. Якщо пацієнт зможе зробити запис до лікаря на один й той ж самий час, на який вже записаний інший пацієнт, то це призведе до помилок роботи клініки і затримок прийому, незадоволеність користувачів та як наслідок – зменшення кількості клієнтів. На рис. 3.20 показано, як чат-бот відхиляє заявку пацієнта з вказанням причини та пропонує команду для повторної реєстрації.

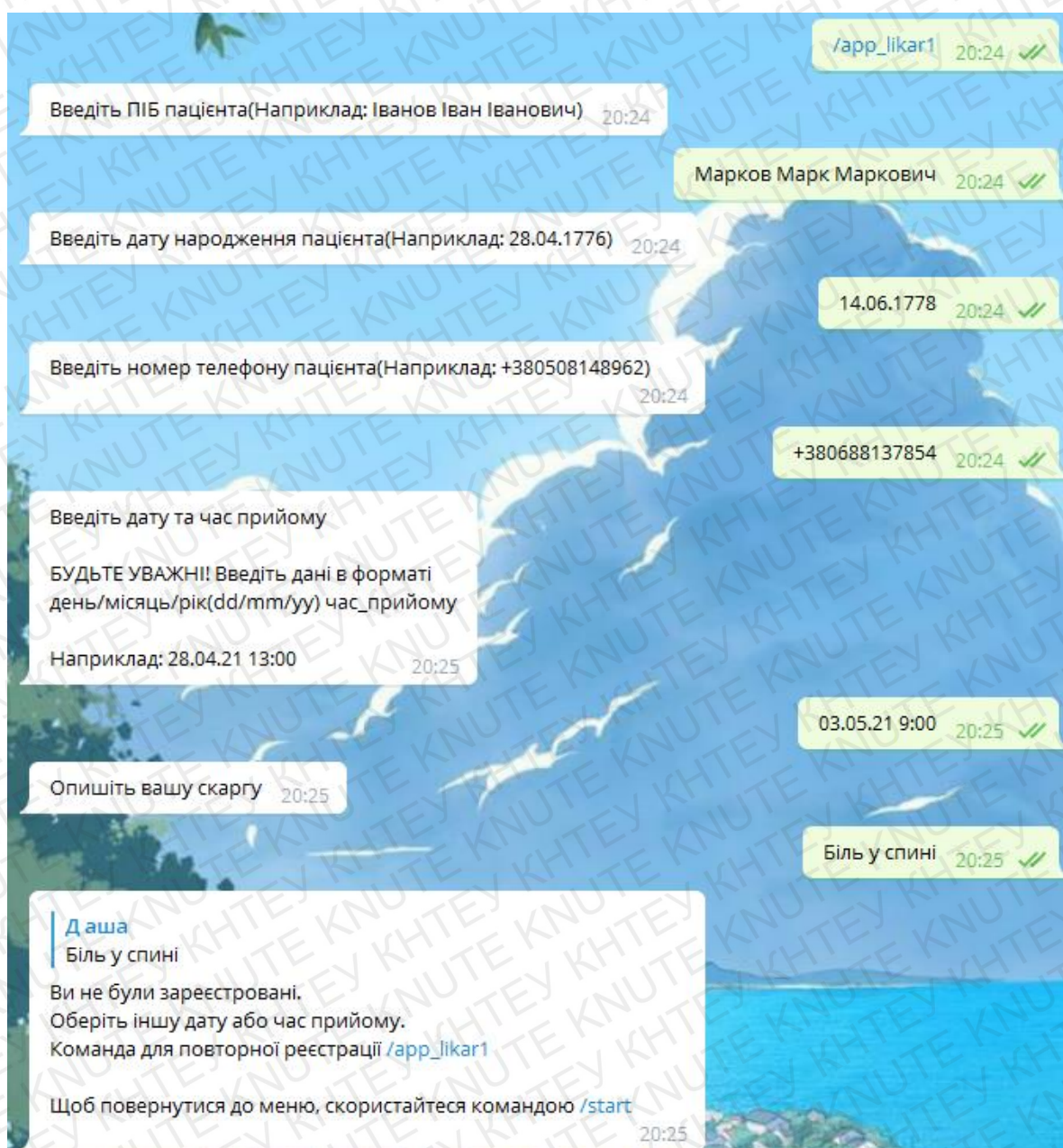


Рис. 3.20 Відхилення заявки пацієнта

Також необхідно перевірити коректність запису даних у таблицю «користувачі». З одного аккаунту можливо зареєструвати необмежену кількість пацієнтів, але система буде розуміти, що всі запити здійснив один й той самий користувач. Для тестування була здійснена реєстрація декількох пацієнтів з одного аккаунту(однаковий user\_id). На рис. 3.21 можна побачити, як виглядає таблиця «лікар1» після реєстрації декількох пацієнтів з одного мобільного пристрою.



Хост: 127.0.0.1 База данных: clinic Таблица: лікар1 Данные Запрос

clinic.лікар1: 3 строк (приблизительно) >> Далее Показывать все Сортировка (1) Столбцы (7/7) Фильтр

| id | pat_name                  | date_of_birth | phone_number  | receipt_time   | complaint       | user_id     |
|----|---------------------------|---------------|---------------|----------------|-----------------|-------------|
| 1  | Буракова Дар'я Миколаївна | 16.08.2000    | +380508148962 | 03.05.21 9:00  | Біль у суглобах | 427 717 304 |
| 2  | Гладченко Сергій Олегович | 28.04.1776    | +380502144964 | 03.05.21 11:00 | Болять коліна   | 427 717 304 |
| 3  | Жмих Соломія Пилипівна    | 13.12.1995    | +380972790021 | 03.05.21 10:00 | Біль у спині    | 427 717 304 |

Рис. 3.21 Реєстрація пацієнтів з одного аккаунту

Таблиця «користувачі» зберігає дані про користувача, з аккаунту якого була здійснена реєстрація пацієнтів(рис. 3.22)

Хост: 127.0.0.1 База данных: clinic Таблица: користувачі Данные Запрос

clinic.користувачі: 1 строк (приблизительно) >> Далее Показывать все Сортировка

| id | first_name | last_name | telegram_user_id |
|----|------------|-----------|------------------|
| 1  | Дар'я      | Буракова  | 427 717 304      |

Рис. 3.22 Результат реєстрації пацієнтів з одного аккаунту у таблиці «користувачі»

## Висновки до розділу

Розуміння того, які функції має виконувати чат-бот, допомагає вибрати алгоритми та інструменти для його створення, розробити вдалу структуру системи. Чат-бот для підтримки клієнтів медичної клініки має здійснювати запис на прийом до певного лікаря, надавати користувачеві інформацію про клініку, відповідати на типові запитання.

Месенджер Telegram пропонує зручний спосіб налаштування чат-ботів на платформі за допомогою BotFather. Створення з'єднання з базою MySQL з підключенням пакету `mysql-connector-python` надає можливість зберігання даних користувача у вигляді спеціальних таблиць, що є окремими для кожного лікаря-спеціаліста.

Чат-бот має бути зручним та інтуїтивно зрозумілим у використанні. Отже, було вирішено створити багаторівневе меню інформаційного блока у вигляді інлайн-клавіатури для спрощення взаємодії бота та користувача. Інформаційний блок містить дані про послуги, ціни, графік роботи клініки та спеціалістів, адресу, контакти та ін. Це повністю автоматизує отримання інформації про клініку.

Тестування системи показало, що програма виконує свої функції відповідно до поставленої задачі, як у загальних випадках, так і за винятковими сценаріями.



## ВИСНОВКИ

Діджиталізація економіки і нова тенденція орієнтації систем підтримки прийняття рішень на використання мобільних пристроїв ставлять задачу розробки нового покоління чат-ботів.

Виходячи з аналізу предметної області, а саме дослідження актуальності та необхідності використання чат-ботів у сфері медицини, зокрема для підтримки взаємодії з клієнтами, було вирішено розробити бота, який зможе автоматизувати процеси отримання інформації та запису на прийом, причому в цілодобовій доступності для користувача.

Дана розробка буде безкоштовна і буде у відкритому доступі для всіх верств населення, у яких є смартфон, підключення до мережі Інтернет, а також месенджер Telegram.

Предметом дослідження кваліфікаційної роботи є засоби створення чат-бота для підтримки взаємодії з клієнтами медичної клініки. Було представлено огляд і порівняльний аналіз технічних засобів для розробки чат-бота, а саме розглянуто загальні поняття чат-бота і переваги його використання для автоматизації бізнес-процесів. Розглянуто існуючі засоби проектування чат-ботів, сучасні технології, пов'язані з розробкою систем. Також була представлена статистика й аналіз сучасних платформ для запровадження програми. Обґрунтовано вибір програмного забезпечення для реалізації чат-бота: месенджеру Telegram, мови програмування Python, середовища розробки PyCharm та бази даних MySQL.

При проектуванні продукту було описане призначення, цілі та задачі розробки, визначено функціональні вимоги до програмного забезпечення, на основі яких було створено функціонал, що має набір системних функцій. Була розроблена детальна структура чат-бота, структура бази даних та наведено формат збереження інформації в ній. Описаний процес створення та налаштування чат-бота у месенджері Telegram за допомогою Botfather та процес з'єднання серверної частини з базою даних MySQL. Була розглянута практична реалізація проекту, показані фрагменти коду, представлений

алгоритм роботи програми. Для перевірки працездатності та високої якості програмного забезпечення здійснено тестування системи, показано як працює програма в загальних випадках та за винятковими сценаріями.

Результатом проведеного дослідження стало створення чат-бота для підтримки взаємодії з клієнтами медичної клініки у месенджері Telegram. Система є сучасною та зручною, задовольняє всі вимоги з точки зору функціональності, придатності використання, привабливості для користувача.

Отже, використання чат-ботів для підтримки взаємодії з клієнтами медичної клініки має переваги як для пацієнтів, так і для співробітників, маркетологів і є вигідним для бізнесу.

Переваги чат-ботів для клієнтів медичної клініки:

- Можливість отримання відповіді на запитання у будь-який зручний для користувача час;
- Можливість запису на прийом у будь-який зручний для користувача час;
- Отримання допомоги в навігації.

Переваги чат-ботів для співробітників медичної клініки:

- Отримання детальної необхідної інформації про пацієнта ще до прийому;
- Економія часу на отримання відповідей на стандартні запитання.

Переваги для маркетологів:

- Відстеження ефективних рекламних джерел залучення клієнтів до клініки;
- Генерування саме тих заявок, які потрібні.

Вигода для бізнесу:

- Підвищення конверсії та отримання більше клієнтів онлайн;
- Розподілення навантаження певних пацієнтів в клініці;
- Контроль якості роботи на всіх етапах від залучення пацієнта онлайн до надання йому медичної допомоги фахівцем;
- Підвищення якості обслуговування пацієнтів.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The State of Chatbots Report: How Chatbots Are Reshaping Online Experiences [Електронний ресурс]. – Режим доступу: <https://www.drift.com/blog/Chatbots-report/>
2. Jung, S.: Semantic vector learning for natural language understanding. Comput. Speech Lang. 56, 130–145(2019) – Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S0885230817303595?via%3Dihub>
3. Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K.: A survey of design techniques for conversational agents. In: Kaushik, S., Gupta, D., Kharb, L., Chahal, D. (eds.) ICICCT 2017. CCIS, vol. 750, pp. 336–350. Springer, Singapore (2017) – Режим доступу: [https://link.springer.com/chapter/10.1007%2F978-981-10-6544-6\\_31](https://link.springer.com/chapter/10.1007%2F978-981-10-6544-6_31)
4. Nimavat, K., Champaneria, T.: Chatbots: an overview types, architecture, tools and future possibilities. Int. J. Sci. Res. Dev. 5, 1019–1024 (2017)
5. Fern, A., et al.: The Best Open Source Chatbot Platforms in 2019 (2019). - Режим доступу: <https://verloop.io/blog/the-best-open-source-chatbot-platforms/>
6. Singh, S., Darbari, H., Bhattacharjee, K., Verma, S.: Open source NLG systems: a survey with a vision to design a true NLG system. 9, 4409–4421 (2016)
7. Nayyar, D.A.: Chatbots and the Open Source Tools You Can Use to Develop Them (2019). – Режим доступу: <https://www.opensourceforu.com/2019/01/chatbots-and-the-open-source-tools-you-can-use-to-develop-them/>
8. Canonico, M., Russis, L.D.: A comparison and critique of natural language understanding tools. Presented at the (2018)
9. Most popular global mobile messenger apps as of January 2021, based on number of monthly active users [Електронний ресурс]. – Режим доступу: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>
10. Number of monthly active WhatsApp users worldwide from April 2013 to March 2020 [Електронний ресурс]. – Режим доступу: <https://www.statista.com/statistics/260819/number-of-monthly-active-whatsapp-users/>
11. Number of monthly active Telegram users worldwide from March 2014 to April 2020 [Електронний ресурс]. – Режим доступу: <https://www.statista.com/statistics/234038/telegram-messenger-mau-users/>
12. Рейтинг мобільних додатків за березень 2020 [Електронний ресурс]. – Режим доступу: <https://tns-ua.com/news/rejting-mobilnih-dodatviv-za-berezen-2020>



13. Рейтинг мобільних додатків за січень 2021 [Електронний ресурс]. – Режим доступу: <https://tns-ua.com/news/rejting-mobilnih-dodatkov-za-sichen-2021>
14. Telegram Bots. Bot Code Examples [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/bots/samples>
15. Python. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Python>
16. Язык программирования Python [Електронний ресурс]. – Режим доступу: <https://web-creator.ru/articles/python>
17. PHP, Ruby, Python – краткая характеристика трёх языков программирования. [Електронний ресурс]. – Режим доступу: <https://www.internet-technologies.ru/articles/php-ruby-python-harakteristika-yazykov-programmirovaniya.html>
18. Інтегроване середовище розробки. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B5%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8>
19. PyCharm: IDE для Python. [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/pycharm-ide-dlya-python-1>
20. PyCharm. [Електронний ресурс]. – Режим доступу: <https://webhamster.ru/mytetrashare/index/mtb172/14838149631rsxf559k9>
21. MySQL. Вікі-Клік- енциклопедія кафедри інформатики и інформаційно-комунікаційних технологій. [Електронний ресурс]. – Режим доступу: <http://wikifizmat.udpu.edu.ua/index.php?title=MySQL>
22. ЧТО ТАКОЕ MYSQL? [Електронний ресурс]. – Режим доступу: <https://freehost.com.ua/faq/wiki/chto-takoe-mysql/>
23. HeidiSQL — это легкий инструмент для управления базами данных с открытым исходным кодом, встроенный в Delphi. [Електронний ресурс]. – Режим доступу: <https://blogs.embarcadero.com/ru/heidisql-is-a-lightweight-open-source-database-management-tool-built-in-delphi/>
24. HeidiSQL. Национальная библиотека им. Н. Э. Баумана Bauman National Library. [Електронний ресурс]. – Режим доступу: <https://ru.bmstu.wiki/HeidiSQL>



## ДОДАТКИ

Додаток А

## Лістинг коду

```
import telebot
from telebot import types
import mysql.connector

bot =
telebot.TeleBot("1582431619:AAHN9rUNT9Uu0mjQx*****YXlDgAU")

@bot.message_handler(commands=['start'])
def any_msg(message):
    keyboardmain = types.InlineKeyboardMarkup(row_width = 2)
    first_button = types.InlineKeyboardButton(text="Про нас",
callback_data="first")
    second_button = types.InlineKeyboardButton(text="Наші лікарі",
callback_data="second")
    third_button = types.InlineKeyboardButton(text="Послуги",
callback_data="third")
    fourth_button = types.InlineKeyboardButton(text="Контакти",
callback_data="fourth")
    fifth_button = types.InlineKeyboardButton(text="Записатися на прийом",
callback_data="fifth")
    keyboardmain.add(first_button, second_button, third_button, fourth_button,
fifth_button)
    bot.send_message(message.chat.id, "Вас вітає клініка Н!",
reply_markup=keyboardmain)

@bot.callback_query_handler(func=lambda call:True)
def callback_inline(call):
    if call.data == "mainmenu":

        keyboardmain = types.InlineKeyboardMarkup(row_width=2)
        first_button = types.InlineKeyboardButton(text="Про нас",
callback_data="first")
        second_button = types.InlineKeyboardButton(text="Наші лікарі",
callback_data="second")
        third_button = types.InlineKeyboardButton(text="Послуги",
callback_data="third")
        fourth_button = types.InlineKeyboardButton(text="Контакти",
callback_data="fourth")
```

```

    fifth_button = types.InlineKeyboardButton(text="Записатися на прийом",
callback_data="fifth")
    keyboardmain.add(first_button, second_button, third_button, fourth_button,
fifth_button)

bot.edit_message_text(chat_id=call.message.chat.id,message_id=call.message.mes
sage_id, text="Вас вітає клініка Н!",reply_markup=keyboardmain)

    if call.data == "first":
        keyboard = types.InlineKeyboardMarkup()
        backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
        keyboard.add(backbutton)

bot.edit_message_text(chat_id=call.message.chat.id,message_id=call.message.mes
sage_id, text="Опис лікарні",reply_markup=keyboard)

    elif call.data == "second":
        keyboard = types.InlineKeyboardMarkup(row_width=1)
        rele1 = types.InlineKeyboardButton(text="Лікар1", callback_data="likar1")
        rele2 = types.InlineKeyboardButton(text="Лікар2", callback_data="likar2")
        backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
        keyboard.add(rele1, rele2, backbutton)

bot.edit_message_text(chat_id=call.message.chat.id,message_id=call.message.mes
sage_id, text="Наші лікарі",reply_markup=keyboard)

    elif call.data == "likar1":
        keyboard = types.InlineKeyboardMarkup()
        backbutton = types.InlineKeyboardButton(text="Назад",
callback_data="second")
        keyboard.add(backbutton)
        bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text="Лікар1 - опис",
reply_markup=keyboard)

    elif call.data == "likar2":
        keyboard = types.InlineKeyboardMarkup()
        backbutton = types.InlineKeyboardButton(text="Назад",
callback_data="second")
        keyboard.add(backbutton)

```



```
bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text="Лікар2 - опис",
reply_markup=keyboard)
```

```
elif call.data == "third":
    keyboard = types.InlineKeyboardMarkup(row_width=1)
    rele1 = types.InlineKeyboardButton(text="Послуга1",
callback_data="posluga1")
    rele2 = types.InlineKeyboardButton(text="Послуга2",
callback_data="posluga2")
    rele3 = types.InlineKeyboardButton(text="Послуга3",
callback_data="posluga3")
    backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
    keyboard.add(rele1, rele2, rele3, backbutton)
```

```
bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.mes
sage_id, text="Наші послуги", reply_markup=keyboard)
```

```
elif call.data == "posluga1":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад",
callback_data="third")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text="Опис послуги\n\nЦіна послуги",
reply_markup=keyboard)
```

```
elif call.data == "posluga2":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад",
callback_data="third")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text="Опис послуги\n\nЦіна послуги",
reply_markup=keyboard)
```

```
elif call.data == "posluga3":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад",
callback_data="third")
    keyboard.add(backbutton)
```

```
bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text="Опис послуги\n\nЦіна послуги",
reply_markup=keyboard)
```

```
elif call.data == "fourth":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
    keyboard.add(backbutton)
```

```
bot.edit_message_text(chat_id=call.message.chat.id,message_id=call.message.message_id, text="Адреса\n\nГрафік роботи\n\nКонтактні номери",reply_markup=keyboard)
```

```
elif call.data == "fifth":
    keyboard = types.InlineKeyboardMarkup(row_width=1)
    rele1 = types.InlineKeyboardButton(text="Лікар1", callback_data="app1")
    rele2 = types.InlineKeyboardButton(text="Лікар2", callback_data="app2")
    backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
    keyboard.add(rele1, rele2, backbutton)
```

```
bot.edit_message_text(chat_id=call.message.chat.id,message_id=call.message.message_id, text="Оберіть лікаря",reply_markup=keyboard)
```

```
elif call.data == "app1":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
text="Для того, щоб здійснити запис до Лікаря1, скористайтесь командою /app_likar1 ", reply_markup=keyboard)
```

```
elif call.data == "app2":
    keyboard = types.InlineKeyboardMarkup()
    backbutton = types.InlineKeyboardButton(text="Назад до меню",
callback_data="mainmenu")
    keyboard.add(backbutton)
    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
text="Для того, щоб здійснити запис до Лікаря2, скористайтесь командою /app_likar2 ",
```



reply\_markup=keyboard)

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="*****",
    database="clinic"
)
```

```
mycursor = mydb.cursor()
```

```
# mycursor.execute("CREATE DATABASE clinic")
```

```
# mycursor.execute("CREATE TABLE лікар1 (pat_name VARCHAR(255),
date_of_birth VARCHAR(255), phone_number VARCHAR(255), receipt_time
VARCHAR(15) NOT NULL UNIQUE, complaint VARCHAR(255))")
```

```
# mycursor.execute("ALTER TABLE лікар1 ADD COLUMN (id INT
AUTO_INCREMENT PRIMARY KEY)")
```

```
# mycursor.execute("ALTER TABLE лікар1 ADD COLUMN (user_id INT(11))")
```

```
#mycursor.execute("CREATE TABLE користувачі (id INT
AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(255), last_name
VARCHAR(255), telegram_user_id INT(11) UNIQUE)")
```

```
#mycursor.execute("CREATE TABLE лікар2 (pat_name VARCHAR(255),
date_of_birth VARCHAR(255), phone_number VARCHAR(255),receipt_time
VARCHAR(15) NOT NULL UNIQUE, complaint VARCHAR(255))")
```

```
#mycursor.execute("ALTER TABLE лікар2 ADD COLUMN (id INT
AUTO_INCREMENT PRIMARY KEY)")
```

```
user_data = {}
```

```
class User:
```

```
    def __init__(self, pat_name):
        self.pat_name = pat_name
        self.date_of_birth = "
```

```

self.phone_number = "
self.receipt_time = "
self.complaint = "

@bot.message_handler(commands=['app_likar1'])
def start_message(message):
    msg = bot.send_message(message.chat.id, "Введіть ПІБ пацієнта(Наприклад:
Іванов Іван Іванович)")
    bot.register_next_step_handler(msg, process_patname_step)

def process_patname_step(message):
    try:
        user_id = message.from_user.id
        user_data[user_id] = User(message.text)

        msg = bot.send_message(message.chat.id, "Введіть дату народження
пацієнта(Наприклад: 28.04.1776)")
        bot.register_next_step_handler(msg, process_dateofbirth_step)
    except Exception as e:
        bot.reply_to(message, 'Помилка')

def process_dateofbirth_step(message):
    try:
        user_id = message.from_user.id
        user = user_data[user_id]
        user.date_of_birth = message.text

        msg = bot.send_message(message.chat.id, "Введіть номер телефону
пацієнта(Наприклад: +380508148962)")
        bot.register_next_step_handler(msg, process_phonenumber_step)
    except Exception as e:
        bot.reply_to(message, 'Помилка')

def process_phonenumber_step(message):
    try:
        user_id = message.from_user.id
        user = user_data[user_id]
        user.phone_number = message.text

        msg = bot.send_message(message.chat.id, "Введіть дату та час
прийому\n\nБУДЬТЕ УВАЖНІ! Введіть дані в

```



форматі\день/місяць/рік(dd/mm/yy) час\_прийому\n\nНаприклад: 28.04.21 13:00")

```

    bot.register_next_step_handler(msg, process_receipttime_step)
except Exception as e:
    bot.reply_to(message, 'Помилка')

def process_receipttime_step(message):
    try:
        user_id = message.from_user.id
        user = user_data[user_id]
        user.receipt_time = message.text

        msg = bot.send_message(message.chat.id, "Опишіть вашу скаргу")
        bot.register_next_step_handler(msg, process_complaint_step)
    except Exception as e:
        bot.reply_to(message, 'Помилка')

def process_complaint_step(message):
    try:
        user_id = message.from_user.id
        user = user_data[user_id]
        user.complaint = message.text

        sql = "SELECT * FROM користувачі WHERE telegram_user_id = {0}".format(user_id)
        mycursor.execute(sql)
        existsUser = mycursor.fetchone()

        if (existsUser == None):
            sql = "INSERT INTO користувачі (first_name, last_name, telegram_user_id) VALUES (%s, %s, %s)"
            val = (message.from_user.first_name, message.from_user.last_name, user_id)
            mycursor.execute(sql, val)

            sql = "INSERT INTO лікар1 (pat_name, date_of_birth, phone_number, receipt_time, complaint, user_id) VALUES (%s, %s, %s, %s, %s, %s)"
            val = (user.pat_name, user.date_of_birth, user.phone_number, user.receipt_time, user.complaint, user_id)
            mycursor.execute(sql, val)
            mydb.commit()

            bot.send_message(message.chat.id, "Ви успішно зареєстровані!\nЩоб повернутися до меню, скористайтесь командою /start")

```

except Exception as e:

```
bot.reply_to(message, 'Ви не були зареєстровані.\nОберіть іншу дату або час прийому.\nКоманда для повторної реєстрації /app_likar1\n\nЩоб повернутися до меню, скористайтесь командою /start')
```

```
@bot.message_handler(content_types=['text'])
```

```
def get_text_messages(message):
```

```
    bot.send_message(message.from_user.id, "Щоб відкрити меню, скористайтесь командою /start.")
```

```
# Enable saving next step handlers to file "./handlers-saves/step.save".
```

```
# Delay=2 means that after any change in next step handlers (e.g. calling register_next_step_handler())
```

```
# saving will happen after delay 2 seconds.
```

```
bot.enable_save_next_step_handlers(delay=2)
```

```
# Load next_step_handlers from save file (default "./handlers-saves/step.save")
```

```
# WARNING It will work only if enable_save_next_step_handlers was called!
```

```
bot.load_next_step_handlers()
```

```
bot.polling()
```