

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
бакалавр
(ступінь вищої освіти)

на тему Програмна реалізація Телеграм-бота для закладу вищої освіти
Software Implementation of a Telegram Bot for a Higher Education Institution

Виконав: студент(ка) 4 курсу, групи КНТ-117
Спеціальності 121 Інженерія програмного
забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Суцєвських О.В.

(прізвище та ініціали)

Керівник Каплієнко Т.І.

(прізвище та ініціали)

Рецензент Шитікова О.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20__ року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Суцєвських Олександр Віталійович
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація Телеграм-бота для закладу вищої освіти Software Implementation of a Telegram Bot for a Higher Education Institution

керівник проєкту (роботи) Каплієнко Тетяна Ігорівна, к.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 28 травня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

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

1. Аналіз предметної області. 2. Аналіз програмних засобів. 3. Основні рішення з реалізації компонентів системи. 4. Експлуатація та тестування програми.

5. Текст програми. 6. Технічне завдання.

5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Каплієнко Т.І., доцент		
Нормоконтроль	Камінська Ж.К., асистент		

7. Дата видачі завдання 15 лютого 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка архітектури програми.	4 тиждень	Розділ 2
4	Розробка програми.	5–6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми.	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи.	9 тиждень	

Студент (ка) _____ Суцєвських О.В.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи) _____ Каплієнко Т.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
51 с., 3 табл., 28 рис., 3 дод., 11 джерел.

PYTHON, TELEGRAM, ЧАТ-БОТ, БОТ, TELEGRAM BOT API.

Об'єкт дослідження – програмні засоби для закладів вищої освіти (ЗВО) у Телеграм.

Предмет дослідження – комп'ютерні програми для надання довідок.

Мета роботи – створення Телеграм бота для автоматизації та прискорення пошуку інформації, що цікавить користувача.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мова програмування Python, персональний комп'ютер з процесором Intel Core i5 під управлінням операційної системи Microsoft Windows 10.

Результати. Розроблено бота у месенджері Телеграм, який у простій формі допоможе знайти інформацію про ЗВО.

Висновки. Розроблено програмну систему, яка автоматизує та прискорює процес пошуку інформації стосовно ЗВО, у якому даний бот використовується.

Галузь використання – ЗВО.

ABSTRACT

Explanatory note to the bachelor's thesis: 51 pages, 28 figures, 3 tables, 3 appendixes, 11 sources.

API PYTHON, TELEGRAM, CHAT-BOT, BOT, TELEGRAM BOT.

The subject of the research is computer programs for higher education institutions in Telegram.

The purpose of the work is to create a telegram bot to automate and speed up the search for information of interest to the user.

Materials, methods and technical means: structural and object-oriented programming, Python programming language, personal computer with Intel Core i5 processor running Microsoft Windows 10 operating system.

Results. A bot has been developed in the Telegram messenger, which in a simple form will help to find information about higher education institutions.

Conclusions. A software system has been developed that automates and speeds up the process of finding information about the higher education institutions in which this bot is used.

Field of use - higher education institutions.

ЗМІСТ

	С.
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Месенджери.....	9
1.2 Чат-боти	10
1.3 Огляд аналогів чат-ботів	11
1.4 Постановка завдання.....	14
2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ.....	15
2.1 Огляд особливостей мови програмування Python	15
2.2 Telegram Bot API	16
2.3 Огляд особливостей системи управління базами даних SQLite	17
2.4 Огляд особливостей мови програмування SQL.....	18
2.5 PyCharm.....	18
2.6 Висновки з розділу.....	19
3 ОСНОВНІ РІШЕННЯ З РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ	20
3.1 Реєстрація чат-бота	20
3.2 Розробка програмної частини	22
3.3 Основні рішення щодо реалізації користувацького інтерфейсу	23
3.4 Проєктування та реалізація БД.....	24
3.4 Алгоритм функціонування телеграм-бота	25
3.5 Висновки з розділу.....	26
4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ.....	27
4.1 Призначення програми	27
4.2 Умови запуску програми.....	27
4.3 Робота з програмою	27
ВИСНОВКИ.....	32
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	33
ДОДАТОК А - Текст програми.....	34
ДОДАТОК Б - Технічне завдання	40
ДОДАТОК В - Слайди презентації.....	44

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API - Application Programming Interface;

IDE - Integrated development environment;

IOS - iPhone Operation System;

SQL - Structured Query Language;

XML - eXtensible Markup Language;

БД - база даних;

ЗВО - заклад вищої освіти;

ОС – операційна система;

СУБД - система управління базами даних.

ВСТУП

Користування інтернетом з року в рік стрімко зростає. Зараз важко сказати скільки у даний момент людей знаходяться у всесвітній мережі.

Інтернет універсальний, він має багато можливостей для самореалізації. Можна зайти абсолютно все, від розваг та навчання до спілкування з будь-якою людиною у світі.

На даний момент існує багато засобів спілкування через інтернет.

Актуальністю даної роботи є те, що багато людей надає перевагу спілкуванню через месенджери та розроблених в них автоматизованих чат-ботів. Чат-боти спрощують щоденні задачі, такі як надання актуальної інформації про курс валют, погоду, останні новини та багато чого іншого. Величезним плюсом є те, що все це можна поєднати на базі одного месенджера.

Метою роботи є розробка бота-помічника на базі месенджера Telegram.

Згідно з метою роботи було поставлено такі завдання:

- аналіз предметної області;
- порівняння існуючих аналогів чат-ботів;
- вибір технологій та середовища розробки;
- розробка чат-бота на базі месенджера Telegram.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Месенджери

Месенджери - від англійського "кур'єр" або "зв'язковий". Це програми для миттєвого обміну повідомленнями між користувачами. Саме в швидкості полягає їх головна перевага перед звичайною електронною поштою. Тут послання передається блискавично, тоді як оновлення поштової скриньки відбувається раз на кілька хвилин. Говорячи про те, що таке месенджер, слід уточнити важливу особливість - він є клієнтською програмою. Це означає, що самостійно програма працювати не може, для її використання необхідно підключення до сервера (центрального комп'ютера мережі) [1].

Оглянемо у порівняльній таблиці декілька прикладів месенджерів (табл. 1.1).

Таблиця 1.1 – Порівняльна таблиця відомих месенджерів

	WhatsApp	Viber	Telegram
Голосові та відео- дзвінки	+	+	+
Зберігання у хмарі	-	-	+
Інтеграція зі сторонніми сервісами	+	+	-
Надсилення великих обсягів даних(1Гб+)	-	-	+
Платформи	IOS,Android	IOS, Android, Windows Phone, Desktop	IOS, Android, Desktop
Повідомлення, що самовидаляється	-	+	+(тільки у секретних чатах)
Можливість використання ботів	-	+	+

У таблиці представлено не всі аспекти функціональної складової месенджерів. Проте, аналізуючи знайдену інформацію, можна побачити функціональну подібність.

1.2 Чат-боти

Останнім часом у месенджерах набирають популярності чат-боти.

Чат-бот – штучний розум, програма-посередник, головне призначення якої - в текстовому або голосовому діалоговому режимі допомогти користувачеві з пошуком інформації, вибором товарів і послуг магазину або сервісу, виконанням нескладного дії (на кшталт бронювання столика або запису до лікаря) або просто розважити за допомогою спілкування [2].

З такого визначення стає зрозуміло, сфера застосування чат-ботів досить широка.

На даний момент існує велика кількість різноманітних чат-ботів, кожен з яких має різний напрямок. Наведемо декілька прикладів:

- ігрові чат-боти;
- рекламні чат-боти;
- новостні чат-боти;
- чат-боти для доставок, сервісів, магазинів;
- консультативні чат-боти та багато інших.

Вони настільки глибоко ввійшли у повсякденне життя користувачів, що використовуються у всіх сферах діяльності людини. Це можна пояснити тим, що більшу частину свого часу людина проводить у смартфоні. Тому зараз важко знайти того, хто б не користувався месенджерами або email-клієнтами.

Бот – спеціальна програма, яка виконує різноманітні завдання для користувача, яка знаходиться у месенджері. Бот виглядає як звичайний чат, проте спілкування відбувається не з живою людиною, а з програмою, яка може замовити автомобіль, якщо це – бот таксі, або прислати останні новини, якщо

це – інформаційний бот, також можна замовити доставку їжі, якщо це – бот ресторану.

Чат-боти останнім часом дуже популярними, бо вони переформатувалися з розважального типу в більш вагому річ, так як стали використовуватися компаніями для вирішення серйозних бізнес-задач. В епоху коли швидко розвиваються інформаційні технології, це нормальний феномен. По-перше, чат-бот це «плацдарм» для вирішення усіх бізнес-задач. По-друге, чат-бот це програма розроблена для підтримання спілкування з користувачем, заздалегідь прописаними, розробником, відповідями, які користувач миттєво отримує.

Чат-боти мають базу знань, яка має усі можливі питання та відповіді на них. Найбільш поширеними варіантами для отримання відповіді є ключові слова, збіги по фразах, збіги по контенту. Епоха друкованих машинок, бігання по бібліотекам та безкінечні черги до крамниць з пресою пішли у забуття. Для збору інформації тепер можна використовувати чат-ботів, при цьому навіть не потрібно виходити з дому.

1.3 Огляд аналогів чат-ботів

1.3.1 RailwayBot

RailwayBot — допомагає шукати і моніторити квитки на українські потяги. З часом розробник обіцяє додати опцію моніторингу квитків для поїздок за кордон [3].

Робота з ботом дуже проста (див. рис. 1.1): вказуєш місто, напрямок та дату, коли планується поїздка. Бот перевіряє наявні квитки та надсилає користувачу, і якщо результат не задовольнив забаганку користувача, у боті також присутній моніторинг. Коли квиток, який задовольняє потреби користувача з'явиться, бот надсилає повідомлення про це.

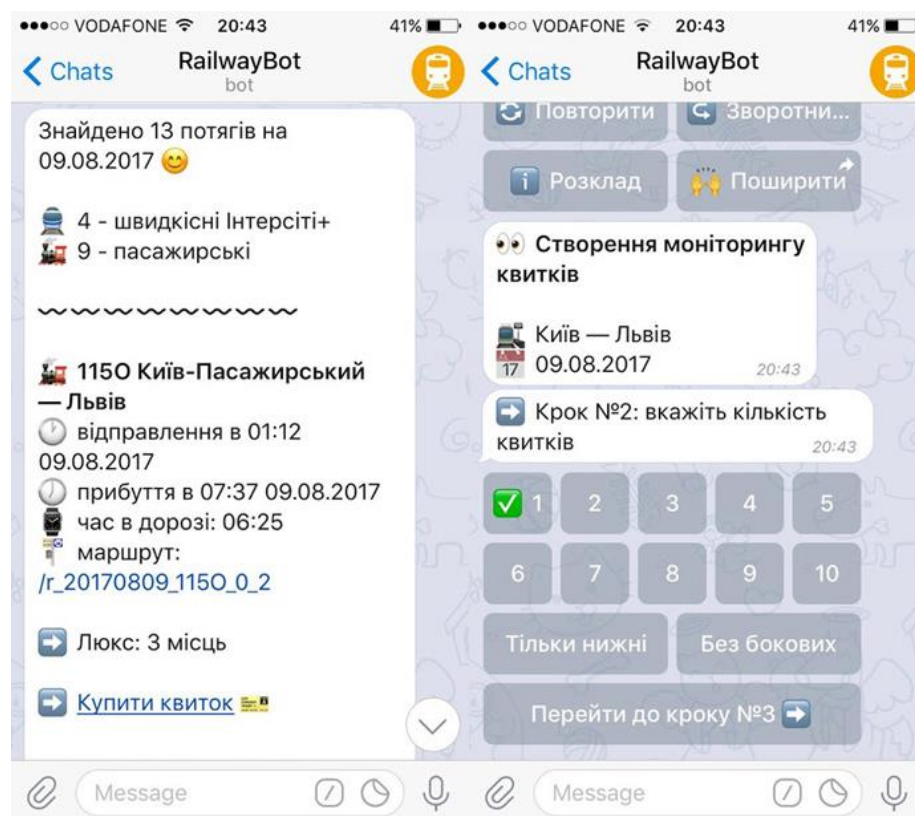


Рисунок 1.1 – Приклад роботи RailwayBot[3]

1.3.2 OpenDataBot

OpenDataBot — інструмент для моніторингу відкритих даних, який буде корисним для підприємців, юристів та журналістів. Бот допомагає знайти будь-яку українську компанію (див. рис. 1.2) за назвою, кодом ОКПО або іменем директора та повідомляє про будь-які зміни в компаніях, на які підписаний користувач.

Бот поєднує дані Єдиного державного реєстру юридичних осіб, фізичних осіб-підприємців та громадських формувань з документами Єдиного державного реєстру судових рішень. Недавно бот придбав дані Мін'юсту і тепер за кодом ЄДРПОУ користувачі можуть отримати виписку з повною інформацією про фірму [3].



Рисунок 1.2 – Приклад роботи OpenDataBot [3]

1.3.3 WolframBot

WolframBot допоможе знайти відповіді на математичні задачі, зробити складні розрахунки, в тому числі обчислити відстані між містами і навіть планетами (рис. 1.3) [3].

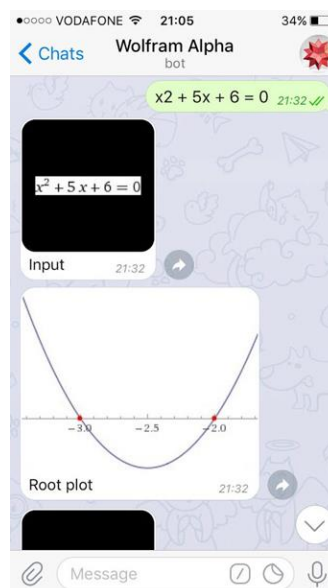


Рисунок 1.3 – Приклад роботи WolframBot [3]

1.4 Постановка завдання

В результаті аналізу предметної області й огляду існуючих програмних аналогів, було поставлено завдання розробити чат-бота для ЗВО з такими характеристиками:

- розподіл на категорії, аби прискорити пошук потрібної інформації;
- надання актуальної інформації;
- можливість залишити запитання або пропозицію;
- необтяжливий користувацький інтерфейс.

2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ

В даному розділі розглянуто основні особливості програмних засобів, за допомогою яких реалізовано дипломний проєкт.

2.1 Огляд особливостей мови програмування Python

Python - інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестувати функції при розробці “знизу-вверх”. Він також зручний як настільний калькулятор. Python дозволяє писати дуже компактні й зручні для читання програми. Програми, написані мовою Python, звичайно значно коротші еквівалента на C або C++ з декількох причин:

- типи даних високого рівня дозволяють програмісту виразити складні операції однією інструкцією;
- групування інструкцій виконується за допомогою відступів замість фігурних дужок;
- немає необхідності в оголошенні змінних [4].

Python працює майже на всіх відомих операційних системах, включаючи: Linux/UNIX, Window, Macintosh, Solaris, macOS, iPhone OS, Palm OS, Windows Mobile, Symbian та Android, тощо. Дозволяє в режимі реального часу взаємодіяти з інтерпретатором, не вимагає компіляцій для виконання коду [5].

На сьогодні Python використовується при реалізації найрізноманітніших проєктів, серед яких:

- розробка сценаріїв для роботи з Web та Internet-програмами;
- мережеве програмування;

- засоби підтримки технологій HTML і XML;
- програми для роботи з електронною поштою й підтримки Internet-протоколів;
- програми для обслуговування найрізноманітніших баз даних;
- програми для наукових розрахунків;
- програми з графічним інтерфейсом;
- створення ігор і комп'ютерної графіки та багато іншого [6].

Мова програмування Python сильно відрізняється від інших мов. Перерахуємо основні відмінності [4]:

- керування пам'яттю - цілком автоматичне — не потрібно хвилюватися щодо розподілу або звільнення пам'яті. Немає загрози “небезпечного посилання”. Java - єдина мова, що пропонує таку концепцію;
- типи зв'язані з об'єктами, а не зі змінними. Це означає, що змінній може бути призначене значення будь-якого типу, і що (наприклад) масив може містити об'єкти різних типів. Традиційні мови не надають такої можливості [4];
- операції звичайно виконуються в більш високому рівні абстракції. Це частково результат того, як написана мова, і частково результат розширеної стандартної бібліотеки кодів, що поставляється разом з Python [4].

2.2 Telegram Bot API

Bot API являється HTTP-інтерфейсом для роботи з ботом в Telegram [7]. Кожен створений бот — це спеціальний аккаунт, створений для автоматичної обробки та надсилання повідомлень.

Існує два діаметрально протилежних за логікою способу отримувати оновлення від створеного бота: `getUpdates` та вебхуки. Вхідні оновлення будуть зберігатися на сервері до тих пір, поки вони не будуть опрацьовані, але не довше 24 годин [7].

Незалежно від способу отримання оновлень, у відповідь користувач отримує об'єкт Update, серіалізований в JSON [7].

Для того аби почати розроблювати, а в подальшому користуватись розробленим ботом, треба отримати токен. Для того аби отримати токен необхідно написати спеціальному боту, @BotFather.

Всі запити до Telegram Bot API повинні здійснюватися через HTTPS в наступному вигляді: https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ.

Допускаються GET і POST запити. Для передачі параметрів в Bot API доступні 4 способи:

- запит в URL;
- application / x-www-form-urlencoded;
- application / json (не підходить для завантаження файлів);
- multipart / form-data (для завантаження файлів) [7].

2.3 Огляд особливостей системи управління базами даних SQLite

SQLite – це вбудована реляційна база даних, з відкритим вихідним кодом. Тобто вона не використовує звичну модель роботи бази даних клієнт-сервер і не є окремим працюючим процесом. Наприклад, база даних MySQL. Іншими словами движок SQLite стає як би частиною нашого веб-додатку. При такому підході база даних SQLite (з усіма таблицями) являє собою звичайний текстовий файл, який користувач може розташувати в зручному для нього місці [8].

Клієнтська частина програмного забезпечення SQLite здійснюється через командний рядок, що дозволяє звернутись до файлу бази даних (БД) на основі типових функцій операційної системи (ОС).

2.4 Огляд особливостей мови програмування SQL

SQL - простими словами, це мова програмування структурованих запитів (SQL, Structured Query Language), який використовується в якості ефективного способу збереження даних, пошуку їх частин, поновлення, вилучення з бази і видалення [9].

Переваги SQL:

- Незалежність від існуючої в даній системі управління базами даних (СУБД). Тексти SQL є універсальними для багатьох СУБД. Однак, це правило поширюється на прості завдання, пов'язані з обробкою даних в таблицях.
- Наявність стандартів SQL сприяє «стабілізації» мови.
- Декларативність. Ця перевага полягає в тому, що при роботі з даними, програміст вибирає тільки ту інформацію, яка повинна бути змінена або модифікована. Те, яким чином це буде зроблено, в автоматичному режимі вирішується на програмному рівні самої СУБД [10].

Недоліки SQL:

- SQL не відповідає реляційної моделі побудови даних. У цьому плані, SQL заміщає мову Tutorial D, який є істинно реляційних.
- Складність SQL визначає його призначення. Мова настільки складний, що їм може користуватися тільки програміст. Хоча спочатку він замислювався як засіб управління, з яким буде працювати звичайний користувач.
- Деякий невідповідність стандартів. Багато компаній, що розробляють СУБД додають свої особливості в діалект мови SQL, що істотно впливає на універсальність мови [10].

2.5 PyCharm

PyCharm є IDE для Python.

IDE (Integrated development environment) – це інтегроване середовище розробки, інакше кажучи, це комплекс програмних засобів, які дають змогу вести розробку певною мовою програмування більш зручніше. Як правило, IDE має відладчик, компілятор або інтерпретатор, текстовий редактор та інше програмне забезпечення.

PyCharm вражає дуже зручним редактором коду, який має усі корисні функції, а саме: підсвічування синтаксису, автоматичне форматування, доповнення та відступи. Ця програма дозволяє швидко перевірити необхідну версію інтерпретатора мови на сумісність. Також є можливість використовувати шаблони програмного коду.

Швидкість перетворення програмного коду (рефакторинг коду) з використанням зручного графічного відладчика – це характерна риса PyCharm.

Утиліта підтримує всі свіжі версії Django, а також IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK і багато інших інструментів [11].

PyCharm дозволяє проводити інтегроване Unit тестування, а також використовувати різні інтерактивні консолі, а саме для: SSH, Python, Django, відладчика і баз даних.

PyCharm є крос-платформним середовищем розробки.

2.6 Висновки з розділу

У цьому розділі було розглянуто основні особливості програмних засобів, використаних при реалізації проєкту: мов програмування Python, SQL, СУБД SQLite, Telegram Bot API та PyCharm.

3 ОСНОВНІ РІШЕННЯ З РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

В даному розділі розглянуто основні рішення з реалізацій компонентів системи, за допомогою яких було реалізовано дипломний проєкт. Реалізація програми наведена у додатку А.

3.1 Реєстрація чат-бота

Першим кроком до розробки бота є реєстрація в спеціального чат-бота «BotFather». Реєстрація починається з команди «/newbot», після чого пропонується ввести назву та ім'я бота. Важливо пам'ятати, що при вводі імені обов'язковою умовою є те, що воно має бути написано без пропусків та в кінці має бути вказано «Bot» або «_bot». Якщо всі умови виконані, то «BotFather» видає токен (спеціальний набір символів для доступу до HTTP API Telegram Bot) та URL-адресу для доступу до чат-боту. Приклад реєстрації чат-бота на рисунку 3.1.

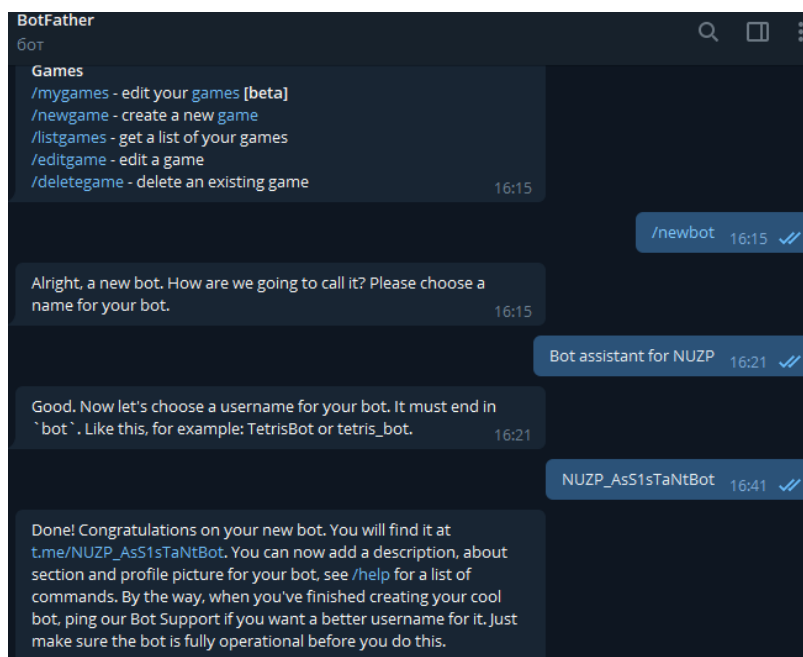


Рисунок 3.1 – Приклад реєстрації чат-бота

Для встановлення додаткових параметрів, таких як: фото чат-бота, вітальне повідомлення, опис чат-бота, а також видалення існуючих чат-ботів, існують наступні команди (таблиця 3.1).

Таблиця 3.1 – Доступні команди для зміни налаштувань в чат-боті

Команда	Опис
/setname	Змінює існуюче ім'я
/setdescriptions	Присвоює текст, який буде відображатись при першому відкритті чат-бота
/setabouttext	Присвоює текст в полі «Про чат-бот»
/setuserpic	Присвоює обрану картинку
/setcommands	Дозволяє створити список доступних команд
/deletebot	Видаляє обраного чат-бота

Крім команд для зміни основних параметрів чат-бота існують команди, які дозволяють вводити незмінні параметри, а також значення, які представлені в таблиці 3.2.

Таблиця 3.2 – Доступні команди для додаткових налаштувань чат-бота

Команда	Опис
/token	Повертає отриманий раніше токен у обраного бота
/revoke	Анулює токен доступу до бота
/setinline	Включає або виключає можливість викликати бота з інших чатів
/setinlinegeo	Включає або виключає можливість передачі місцезнаходження бота з іншого чату

Продовження таблиці 3.2

Команда	Опис
/setinlinefeedback	Дозволяє отримувати інформацію про кількість обраних користувачем команд
/setjoininggroup	Визначає чи може бот бути доданим до групових чатів
/setprivacy	Включає режим конфіденційності. В цьому режимі бот отримує, обробляє та надсилає назад інформацію окремо для кожного користувача в чаті

Після налаштування в Телеграм та отримання токена можна переходити до розробки програмної частини чат-бота.

3.2 Розробка програмної частини

Для того аби розпочати створювати чат-бота на мові програмування Python, треба встановити відповідні бібліотеки (рис. 3.2 та 3.3).

```
C:\Users\Minot>pip install pyTelegramBotAPI
```

Рисунок 3.2 – Встановлення бібліотеки pyTelegramBotAPI

```
C:\Users\Minot>pip install aiogram
```

Рисунок 3.3 – Встановлення бібліотеки aiogram

При розробці бота були реалізовані наступні функції програми:

- `def register(message)` – це функція, яка при виклику через команду, `/register`, відправляє користувачу повідомлення з проханням залишити питання або пропозицію;

- `def next_step_register(message)` – це продовження функції `def register(message)`, в якій підключається база даних та створюється таблиця (якщо її не існує), та одразу у створену таблицю записуються залишені користувачем дані ;

- `def send(id, text)` – ця функція передбачена для того, якщо користувач вводить будь-який текст, то для нього відкривається вбудована клавіатура;

- `def answer(message)` – це функція, яка при виклику через команду, `/start`, відправляє користувачу повідомлення з вітанням, описом вбудованої команди та проханням обрати питання, яке цікавить користувача;

- `def next_step_auditories(message)` – ця функція передбачена для того, аби отримувати повідомлення користувача, шукати співпадіння в базі даних та виводити їх;

- `def main(message)` – це головна функція, у якій прописано як діяти після обрання користувачем тої чи іншої дії.

Програма розроблена таким чином, що можна обрати як відповідати боту, або обрати відповідну кнопку на клавіатурі або написати повідомлення з назвою відповідного розділу.

3.3 Основні рішення щодо реалізації користувацького інтерфейсу

Користувацький інтерфейс створений на базі месенджера Telegram.

Вибір елементів керування та їх розміщення здійснювалося з міркувань раціонального використання простору, а також враховано очікування користувачів сформованих використанням програм-аналогів.

Для безпосередніх маніпуляцій по ходу виконання програми були розроблені спеціальні клавіатури з кнопками. На рис. 3.4-3.5 зображені розроблені клавіатури.

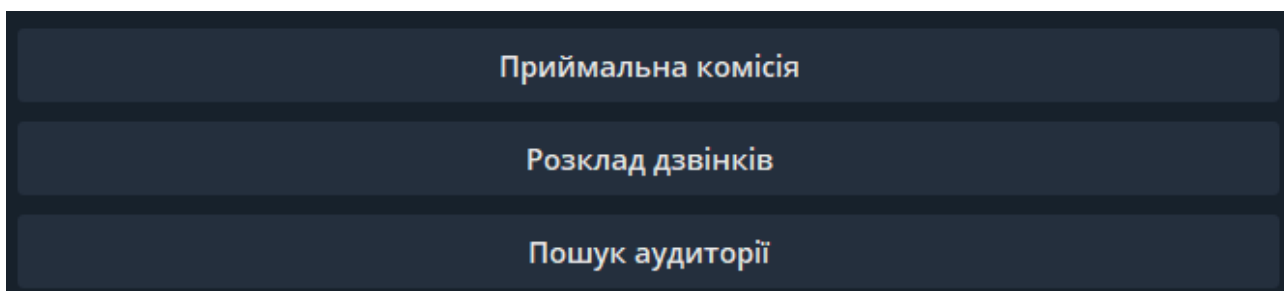


Рисунок 3.4 – Розроблено Reply клавіатуру

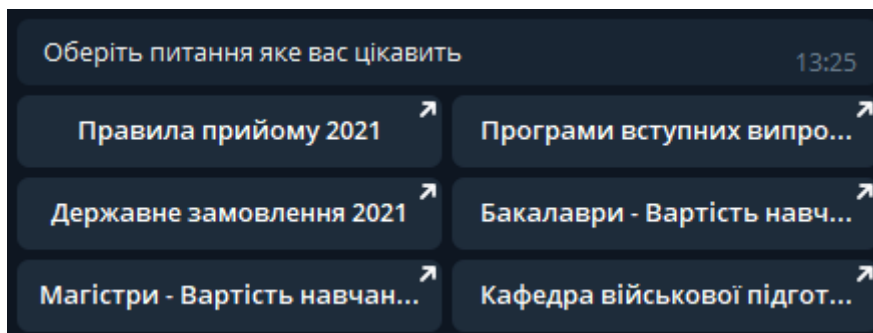


Рисунок 3.5 – Розроблено Inline клавіатуру

3.4 Проєктування та реалізація БД

Для реалізації поставленої задачі з'явилась необхідність використання бази даних. В проєкті використовується СУБД SQLite.

Для роботи з СУБД SQLite було використано наступні SQL-запити:

- CREATE TABLE IF NOT EXIST (створення нової таблиці, якщо вона ще не існує);
- SELECT FROM (вибірка елементів з таблиці);
- INSERT INTO (вставка рядка в таблицю).

На рисунку 3.6 та 3.7 зображено розроблені таблиці в БД, відкриті у спеціальній програмі для огляду та створення баз даних.

Table name: ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	id	INTEGER							NULL
2	user_id	INTEGER							NULL
3	user_name	STRING							NULL
4	user_message	TEXT							NULL

Рисунок 3.6 – Розроблена таблиця messages

Table name: ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	number_name	INTEGER							NULL
2	number_message	TEXT							NULL

Рисунок 3.7 – Розроблена таблиця auditories

3.4 Схема функціонування телеграм-бота

На рисунку 3.8 показано схему функціонування телеграм-бота.

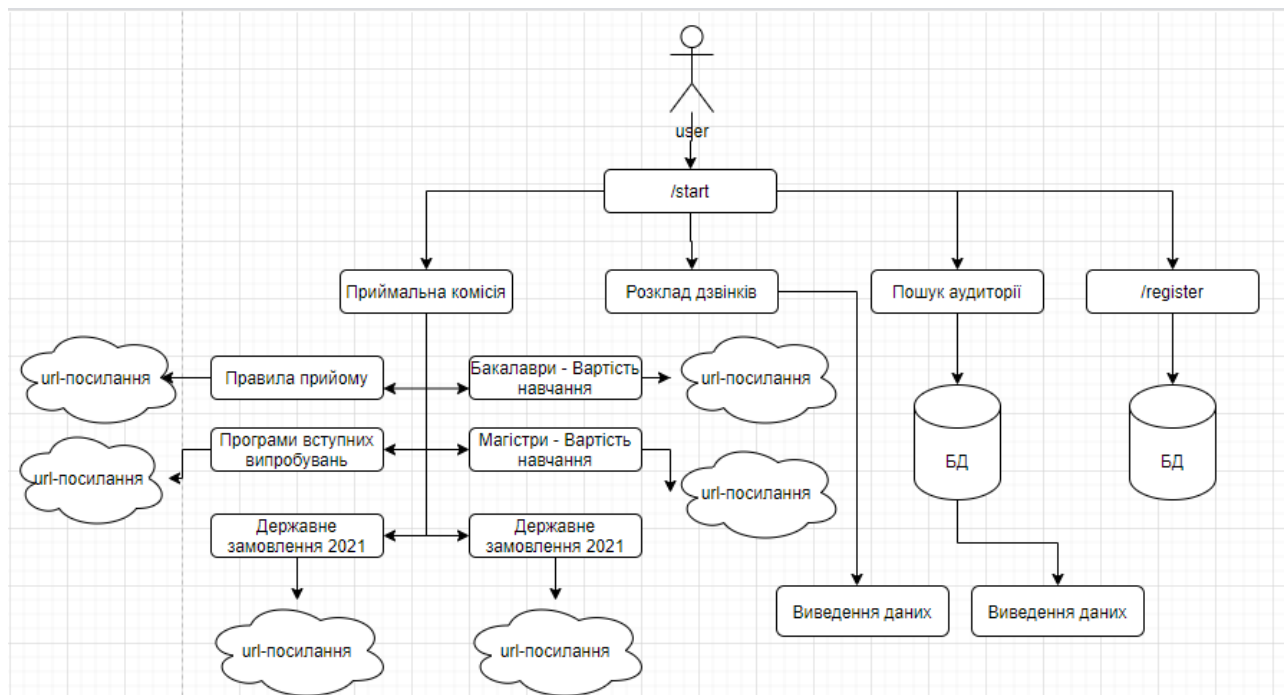


Рисунок 3.8 – Схема функціонування телеграм-бота

3.5 Висновки з розділу

У цьому розділі було виконано реалізацію попередньо визначених компонентів системи. Було розроблено структуру таблиць БД, а також прийнято рішення щодо реалізації користувацького інтерфейсу.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Програмне забезпечення дозволяє знаходити потрібну користувачу інформацію стосовно ЗВО, в якому даний застосунок використовується.

4.2 Умови запуску програми

До складу технічних засобів повинен входити персональний комп'ютер або ноутбук, який включає у себе:

- ОС – Windows 10;
- оснащений 256 МБ оперативної пам'яті;
- процесор Intel CORE I5 2600 МГц;
- має доступ до мережі Інтернет;
- засоби керування та відображення інформації (клавіатура, миша та монітор).

4.3 Робота з програмою

Робота з програмою розпочинається тоді, коли оператор запускає виконуваний файл програми. Після запуску для користувачів даного Телеграм бота стає можливим користуватись його функціями. Приклад роботи наведено на рис.4.1.

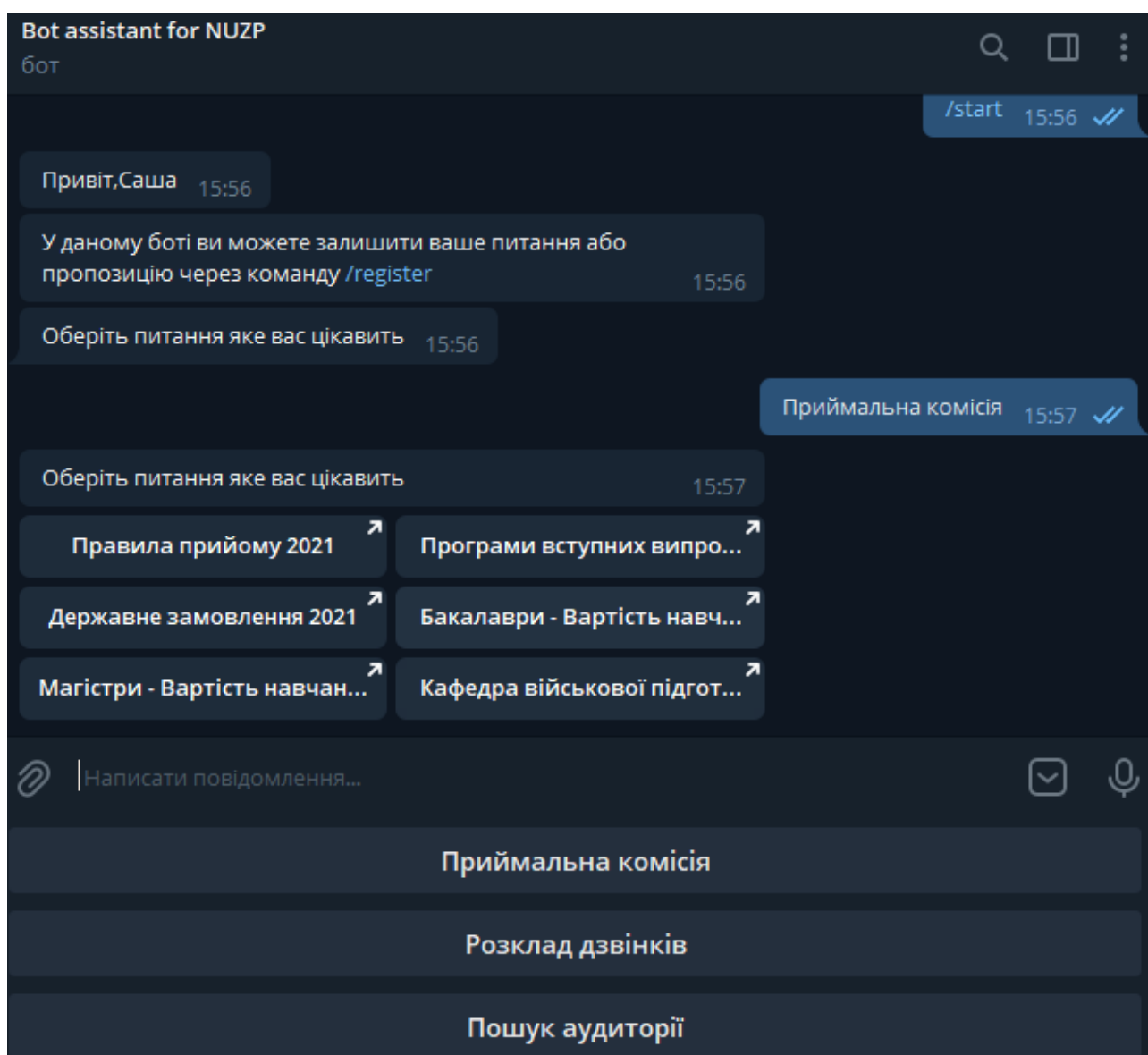


Рисунок 4.1 – Приклад роботи програми в месенджері Телеграм

За допомогою команди `/start` розпочинається робота програми. Далі треба обрати один з запропонованих розділів розміщених на кнопках. У даному випадку було обрано приймальну комісію. Після обрання цього пункту бот надсилає відповідь у вигляді Inline клавіатури вже з конкретного питання.

На рисунку 4.2 зображено повідомлення з вибором дії щодо переходу по посиланню чи ні.

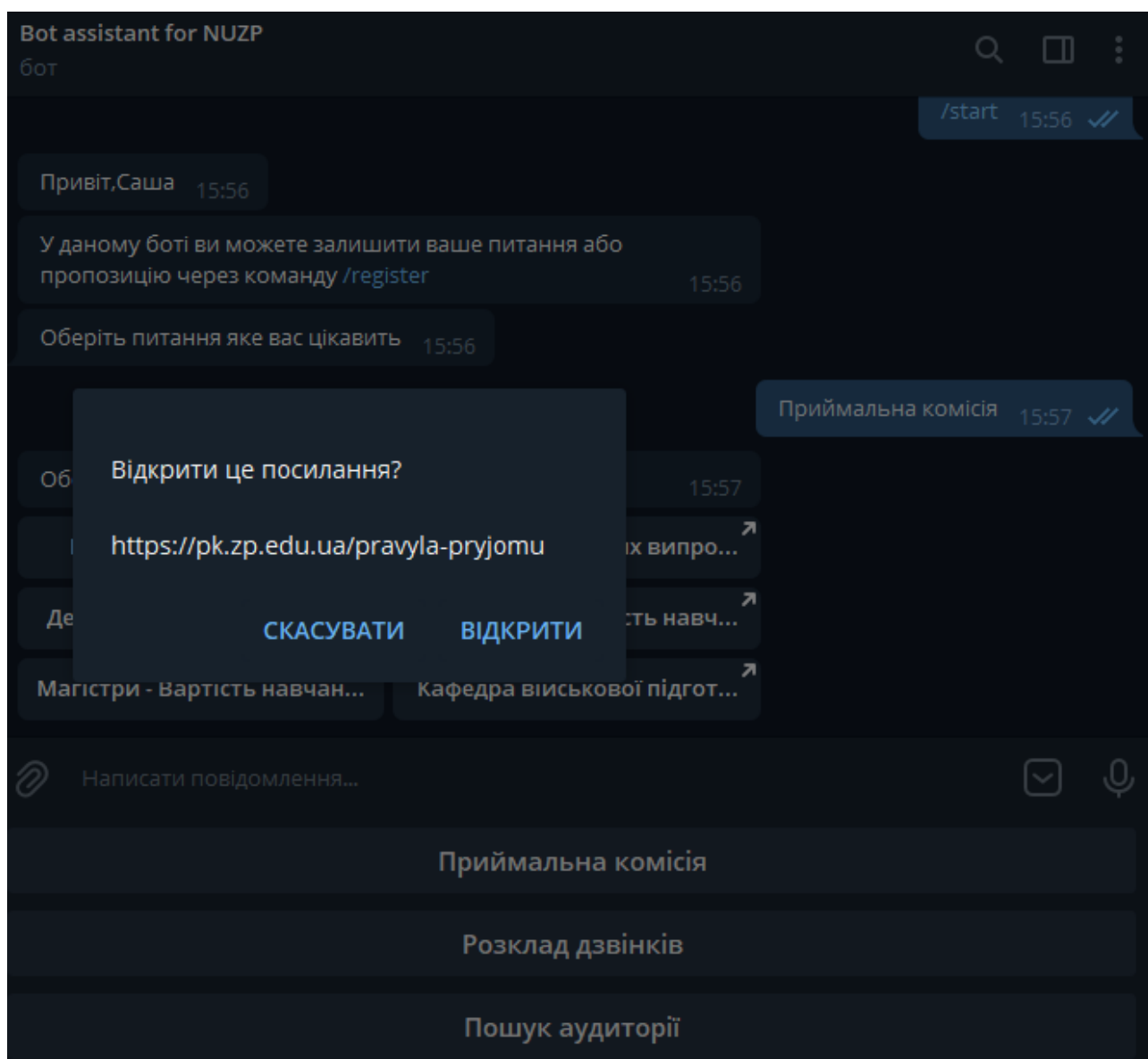


Рисунок 4.2 – Приклад повідомлення

На рисунку 4.3 та 4.4 зображено приклад роботи команди /register, за допомогою якого можна залишити своє повідомлення.

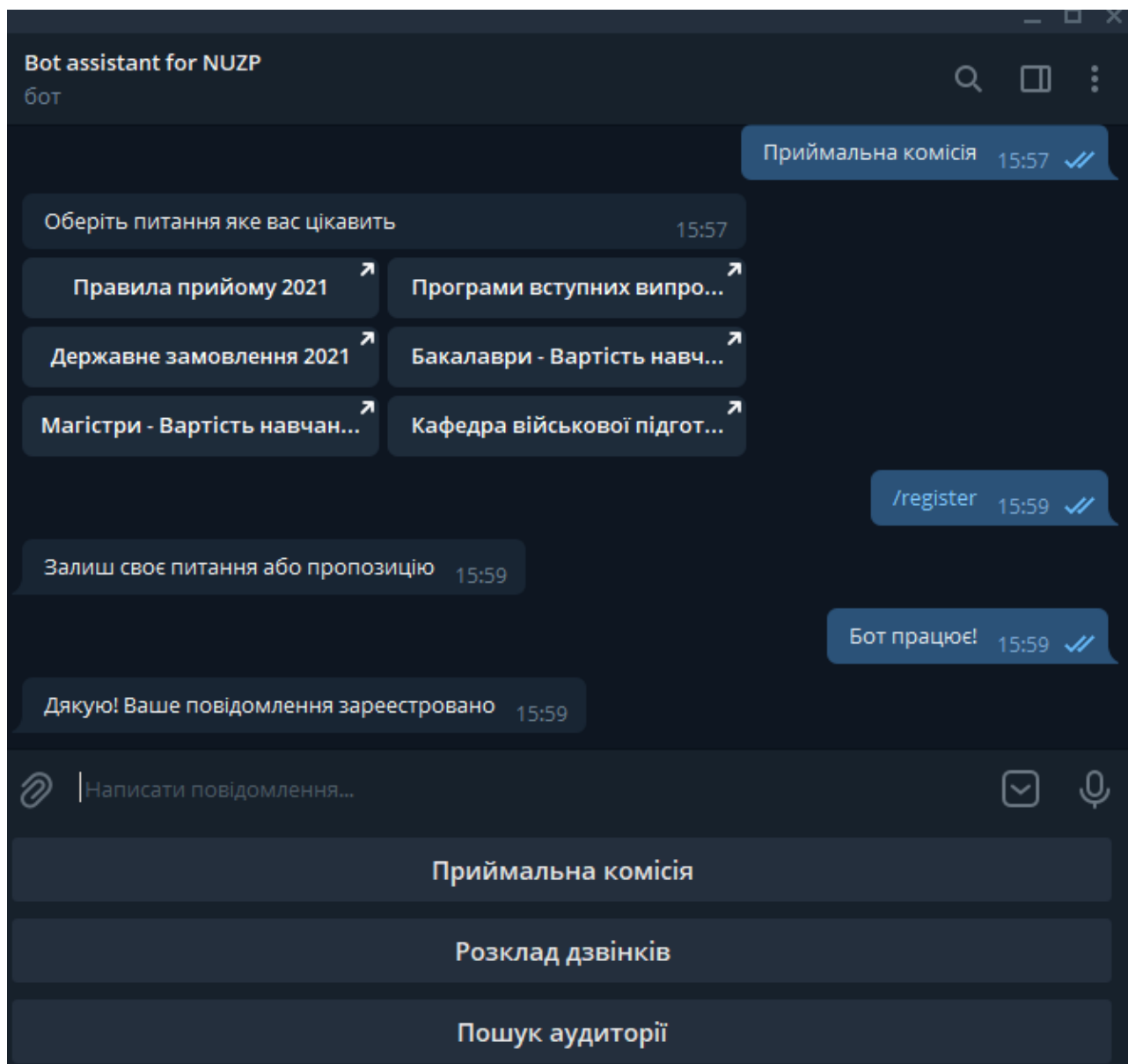


Рисунок 4.3 – Приклад роботи команди /register

	user id	user name	user message
1	467291431	Саша	Бот працює!

Рисунок 4.4 – Приклад занесення даних до БД за допомогою команди /register

4.4 Схематичне зображення роботи системи

На рисунку 4.5 показано схематичне зображення роботи системи.

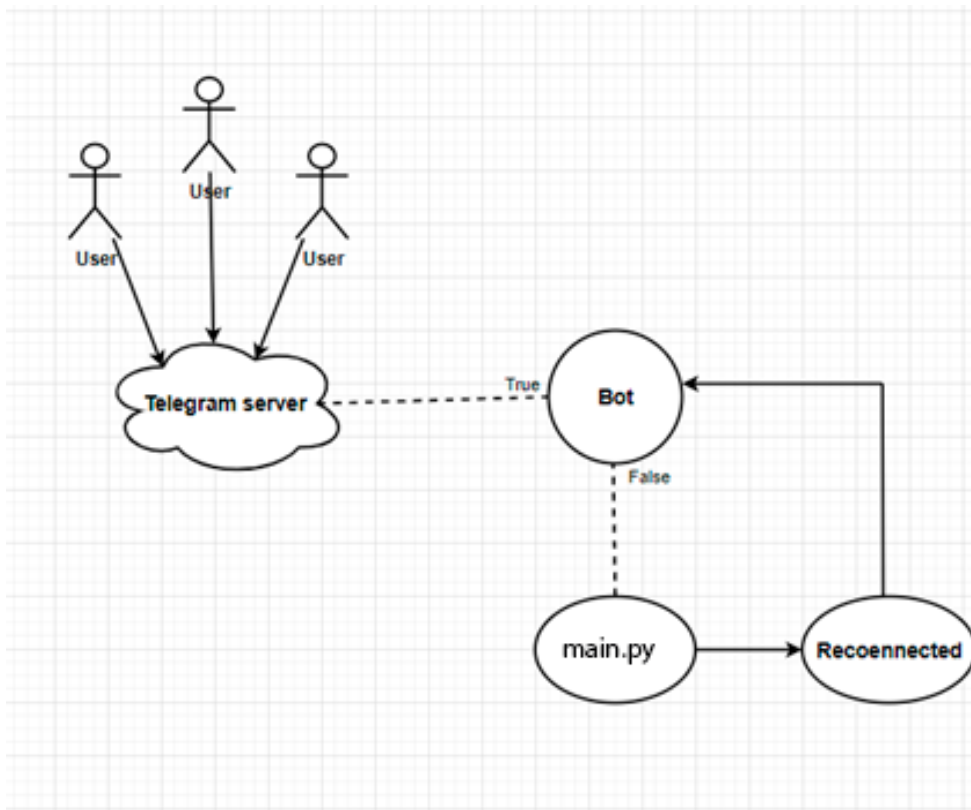


Рисунок 4.5 – Схематичне зображення роботи системи

4.5 Висновки з розділу

У розділі «Експлуатація та тестування програми» було розглянуто призначення програми, умови її запуску та роботу програми.

ВИСНОВКИ

В ході виконання роботи було виконано аналіз предметної області, виконано розробку архітектури програми, реалізовано її компоненти, а також оформлено такі програмні документи як «Експлуатація та тестування програми» та «Технічне завдання».

На першому етапі роботи було ретельно проаналізовано предметну область: сформовано проблему та актуальність роботи, розглянуто основні поняття, існуючі приклади месенджерів, виявлено їх переваги та недоліки, також були оглянуті аналоги чат-ботів. Наступним кроком став «Аналіз програмних засобів», у ньому було обрано інструменти для розробки ПЗ, а також мова програмування. На наступному етапі було реалізовано компоненти програми на основі «Технічного завдання» та надано їхнє обґрунтування. Після цього було складено розділ «Експлуатація та тестування програми».

В результаті роботи було отримано програмний продукт, що дозволяє виконати поставлене завдання, а саме – створення Телеграм бота для автоматизації та прискорення пошуку інформації, що цікавить користувача. Роботу програми було протестовано на реальних користувачах. За необхідності програма може бути модифікована та використана для подальшої роботи в НУ «Запорізька політехніка».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Що таке месенджер? Популярні мобільні месенджери [Електрон. ресурс]. – Режим доступу: <https://ukr.kagutech.com/4323712-what-is-an-instant-messenger-popular-mobile-messengers>
2. Чат-боты: электронный умный собеседник [Электрон. ресурс] – Режим доступа: <https://sharkdevelop.com/chat-botyi/>
3. Топ-10 полезных Telegram-ботов для украинских пользователей [Электрон. ресурс] – Режим доступа: <https://www.epravda.com.ua/rus/publications/2017/08/7/627822/>
4. Що таке Python? [Електрон. ресурс] – Режим доступу: <http://www.plug.org.ua/documentation/about-python>
5. Цікаві факти про Python [Електрон. ресурс] – Режим доступу: <http://www.itschool.vn.ua/interesting-python/>
6. Характеристики Програмування мовою Python. – Навчальна книга Богдан, 2019. – 504 с.
7. Справочник по Bot API [Электрон. ресурс] – Режим доступа: <https://tlgrm.ru/docs/bots/api>
8. Введения в базу даних Sqlite [Электрон. ресурс] – Режим доступа: <http://yoip.com.ua/vvedennya-v-bazu-danih-sqlite/>
9. SQL — что это такое простым языком [Электрон. ресурс] – Режим доступа: <https://www.zeluslugi.ru/info-czentr/it-glossary/term-sql>
10. Що таке SQL? [Електрон. ресурс] – Режим доступу: <https://a-yak.com/shho-take-sql/>
11. PyCharm: IDE для Python [Електрон. ресурс] – Режим доступу: <https://timeweb.com/ru/community/articles/pycharm-ide-dlya-python-1>

ДОДАТОК А
Текст програми

Main.py:

```
bot =
telebot.TeleBot('1655018751:AAH7l3MXLOF2kdeOzh0BKuHCJ6L4j58SvZ0')
```

```
bells = {
'first_lesson': '1 пара | 08:30 / 09:50',
'second_lesson': '2 пара | 10:05 / 11:25',
'third_lesson': '3 пара | 11:55 / 13:15',
'four_lesson': '4 пара | 13:25 / 14:45',
'five_lesson': '5 пара | 14:55 / 16:15',
'six_lesson': '6 пара | 16:45 / 18:05',
'seven_lesson': '7 пара | 18:15 / 19:35',
'eight_lesson': '8 пара | 19:45 / 21:05'
}
```

```
connect = sqlite3.connect('message.db')
cursor = connect.cursor()
```

```
cursor.execute("""CREATE TABLE IF NOT EXISTS auditories(
```

```
    number_name INTEGER,
    number_message TEXT
) """)
connect.commit()
```

```
audience = [
    ( 10, 'Корпус №3: 1 поверх'),
    ( 11, 'Корпус №3: 1 поверх'),
    ( 12, 'Корпус №3: 1 поверх'),
    (13, 'Корпус №3: 1 поверх'),
    (14, 'Корпус №3: 1 поверх'),
    (15, 'Корпус №3: 1 поверх'),
    (16, 'Корпус №3: 1 поверх'),
    (17, 'Корпус №3: 1 поверх'),
    (18, 'Корпус №3: 1 поверх'),
    (19, 'Корпус №3: 1 поверх'),
    (20, 'Корпус №3: 2 поверх'),
    (21, 'Корпус №3: 2 поверх'),
    (22, 'Корпус №3: 2 поверх'),
    (23, 'Корпус №3: 2 поверх'),
    (24, 'Корпус №3: 2 поверх'),
    (26, 'Корпус №3: 2 поверх'),
    (28, 'Корпус №3: 2 поверх'),
    (29, 'Корпус №3: 2 поверх'),
```

```

(30, 'Корпус №3: 3 поверх'),
(31, 'Корпус №3: 3 поверх'),
(32, 'Корпус №3: 3 поверх'),
(33, 'Корпус №3: 3 поверх'),
(34, 'Корпус №3: 3 поверх'),
(35, 'Корпус №3: 3 поверх'),
(36, 'Корпус №3: 3 поверх'),
(37, 'Корпус №3: 3 поверх'),
(38, 'Корпус №3: 3 поверх'),
(39, 'Корпус №3: 3 поверх'),
(40, 'Корпус №3: 4 поверх'),
(41, 'Корпус №3: 4 поверх'),
(42, 'Корпус №3: 4 поверх'),
(43, 'Корпус №3: 4 поверх'),
(44, 'Корпус №3: 4 поверх'),
(45, 'Корпус №3: 4 поверх'),
(46, 'Корпус №3: 4 поверх'),
(47, 'Корпус №3: 4 поверх'),
(48, 'Корпус №3: 4 поверх'),
(49, 'Корпус №3: 4 поверх'),
(50, 'Корпус №3: 5 поверх'),
(51, 'Корпус №3: 5 поверх'),
(52, 'Корпус №3: 5 поверх'),
(54, 'Корпус №3: 5 поверх'),
(55, 'Корпус №3: 5 поверх'),
(56, 'Корпус №3: 5 поверх'),
(57, 'Корпус №3: 5 поверх'),
(58, 'Корпус №3: 5 поверх')
]
cursor.executemany("INSERT INTO auditories VALUES(?,?)", audience)
connect.commit()

keyboard1 = telebot.types.ReplyKeyboardMarkup(True)
keyboard1.row("Приймальна комісія")
keyboard1.row("Розклад дзвінків")
keyboard1.row("Пошук аудиторії")

@bot.message_handler(commands=['register'])
def register(message):

    mess = bot.send_message(message.chat.id, "Залиш своє питання або пропозицію")

    bot.register_next_step_handler(mess, next_step_register)

```

```

def next_step_register(message):

    connect = sqlite3.connect('message.db')
    cursor = connect.cursor()

    cursor.execute("""CREATE TABLE IF NOT EXISTS messages(

        user_id INTEGER,
        user_name STRING,
        user_message TEXT
    ) """)

    connect.commit()

    user_id = message.chat.id
    user_name = message.from_user.first_name
    user_message = message.text

    cursor.execute(f"INSERT INTO messages VALUES(?,?,?)", ( user_id, user_name,
user_message))
    connect.commit()

    bot.send_message(message.chat.id, "Дякую! Ваше повідомлення
zareestrowano")

def send(id, text):
    bot.send_message(id, text, reply_markup = keyboard1)

@bot.message_handler(commands = ['start'])
def answer(message):
    bot.send_message(message.chat.id, f"Привіт, {message.from_user.first_name}")

    bot.send_message(message.chat.id, "У даному боті ви можете залишити ваше
питання або пропозицію через команду /register")

    bot.send_message(message.chat.id, "Оберіть питання яке вас
цікавить",reply_markup=keyboard1)

@bot.message_handler(content_types = ['text'])
def main(message):
    id = message.chat.id
    msg = message.text

```

```

if msg == "Приймальна комісія":
    keyboard = types.InlineKeyboardMarkup(row_width=2)

    callback_button = types.InlineKeyboardButton(text="Правила прийому 2021",
url = "https://pk.zp.edu.ua/pravyla-pryjomu")

    callback_button_1 = types.InlineKeyboardButton(text="Програми вступних
випробувань", url = "https://pk.zp.edu.ua/pravyla-pryjomu/prohramy-vstupnyh-
vyprobuvan")

    callback_button_2 = types.InlineKeyboardButton(text=" Державне замовлення
2021", url = "https://pk.zp.edu.ua/wp-content/uploads/2021/04/91_max_obs.pdf")

    callback_button_3 = types.InlineKeyboardButton(text=" Бакалаври - Вартість
навчання", url = "https://pk.zp.edu.ua/fakultety-ta-napryamy-pidhotovky/vartist-
navchannya-bakalavr")

    callback_button_4 = types.InlineKeyboardButton(text=" Магістри - Вартість
навчання", url = "https://pk.zp.edu.ua/fakultety-ta-napryamy-pidhotovky/vartist-
navchannya-spetsialisty-mahistry")

    callback_button_5 = types.InlineKeyboardButton(text=" Кафедра військової
підготовки", url = "https://pk.zp.edu.ua/fakultety-ta-napryamy-pidhotovky/kafedra-
vijskovoji-pidhotovky")

    keyboard.add(callback_button, callback_button_1, callback_button_2,
callback_button_3, callback_button_4, callback_button_5)

    bot.send_message(message.chat.id, "Оберіть питання яке вас
цікавить",reply_markup=keyboard)

elif msg == "Розклад дзвінків":
    bot.send_message(message.chat.id, "Розклад дзвінків:")
    for values in bells.values():
        bot.send_message(id, values, '\n')

elif msg == "Пошук аудиторії":
    mess = bot.send_message(message.chat.id, "Введіть аудиторію:")
    bot.register_next_step_handler(mess, next_step_auditories)

def next_step_auditories(message):
    connect = sqlite3.connect('message.db')
    cursor = connect.cursor()

```

```
numm = message.text

sql = "SELECT * FROM auditories WHERE number_name=?"
request = cursor.execute(sql, numm)

bot.send_message(message.chat.id, "Обрана аудиторія знаходиться: ", request)

bot.polling(none_stop = True)
```

ДОДАТОК Б
Технічне завдання

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломний проєкт на тему «Програмна реалізація Телеграм-бота для закладу вищої освіти », затверджене наказом по НУ «Запорізька політехніка» № 103 від 30 березня 2021 року.

A.2 Призначення розробки

Дане ПЗ призначено для використання в ЗВО, зокрема, в НУ «Запорізька політехніка».

Було встановлено, що використовуючи розроблене ПЗ, диспетчери ЗВО можуть скоротити час роботи за рахунок автоматизації процесу надання довідкових послуг, а також уникнути зайвих витрат часу, що виникають в результаті постійних дзвінків з поверхневими питаннями.

A.3 Вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик:

- розподіл на категорії, аби прискорити пошук потрібної інформації;
- надання актуальної інформації;
- можливість залишити запитання або пропозицію;
- необтяжливий користувацький інтерфейс;
- створюваний для ЗВО.

A.3.2 Вимоги надійності

Програма повинна реалізовувати наступні вимоги по надійності:

- обробка виняткових ситуацій (наприклад, введення тексту з помилкою);
- стабільне функціонування при безперервній роботі комп'ютера.

А.3.3 Умови експлуатації

Умови експлуатації програми відповідають вимогам до експлуатації технічних засобів (комп'ютера, сервера тощо). Для використання програми не потребується додаткова кваліфікація користувача.

А.3.4 Вимоги до складу і параметрів технічних засобів

Вимоги до складу і параметрів технічних засобів:

- ОС Microsoft Windows 7 x32/ 10 x64;
- процесор із тактовою частотою 2.5 ГГц і вище;
- жорсткий диск місткістю не менше, ніж 100 МБ;
- мінімум 2 ГБ обсягу оперативної пам'яті (рекомендується 4 ГБ і більше);
- додаткове ПЗ: месенджер Telegram .

А.3.5 Вимоги до маркування та упакування

Програмне забезпечення не потребує додаткових операцій щодо маркування та упакування.

А.3.6 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання програми відповідають аналогічним вимогам носія, на якому вона записана.

А.3.7 Вимоги до програмної документації

Склад програмної документації:

- технічне завдання;
- експлуатація та тестування програми.

А.3.8 Стадії і етапи розробки

Стадії і етапи розробки програми:

- аналіз предметної області;
- проектування;
- реалізація;
- тестування.

А.3.9 Порядок контролю та приймання

Порядок контролю та приймання відповідає затвердженому порядку контролю та приймання дипломних кваліфікаційних робіт студентів денної та заочної форми навчання освітнього ступеня «бакалавр» НУ «Запорізька політехніка».

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька Політехніка»

Кафедра програмних засобів

Розробка телеграм-бота для закладу вищої освіти.

Виконав
ст.гр.КНТ-117
О.В. Суцєвських

Керівник
к.т.н., доцент
Т.І. Каплієнко



2021

Рисунок В.1 – Слайд №1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – програмні засоби для ЗВО у Телеграм.

Предмет дослідження – комп'ютерні програми для надання довідок., бот реалізований на Python.

Мета роботи – створення Телеграм бота для автоматизації та прискорення пошуку інформації, що цікавить користувача.



2

Рисунок В.2 – Слайд №2

Завдання до роботи

В процесі виконання роботи були поставлені такі задачі

- ▶ аналіз предметної області;
- ▶ розробка технічного завдання;
- ▶ порівняння існуючих аналогів чат-ботів;
- ▶ вибір технологій та середовища розробки;
- ▶ розробка чат-бота на базі месенджера Telegram;
- ▶ тестування розробленого програмного забезпечення.

3

Рисунок В.3 – Слайд № 3

Аналіз предметної області

	WhatsApp	Viber	Telegram
Голосові та відео-дзвінки	+	+	+
Зберігання у хмарі	-	-	+
Інтеграція зі сторонніми сервісами	+	+	-
Надсилення великих обсягів даних (1Гб+)	-	-	+
Платформи	IOS, ANDROID	IOS, ANDROID, WINDOWS PHONE, DESKTOP	IOS, ANDROID, DESKTOP
Повідомлення, що самовидаються	-	+	+(тільки у секретних чатах)
Можливість використання ботів	-	+	+

4

Рисунок В.4 – Слайд № 4

Вибір середовища розробки

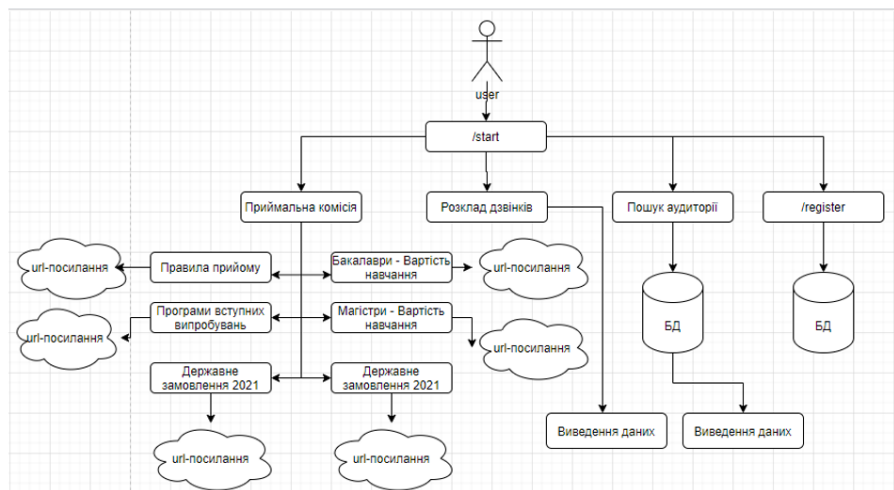
	PyCharm	Visual Studio Code	Komodo	Visual Studio
Підсвічування коду	+	+	+	+
Перевірка орфографії	+	+	+	+
Підтримка Git	+	+	-	+
Гарячі клавіші	+	++	+	++
Недоліки	Може повільно працювати	Працює як JavaScript-процес	На безкоштовній версії доступні не всі функції	Займає дуже багато місця

необхідно встановлювати розширення

5

Рисунок В.5 – Слайд № 5

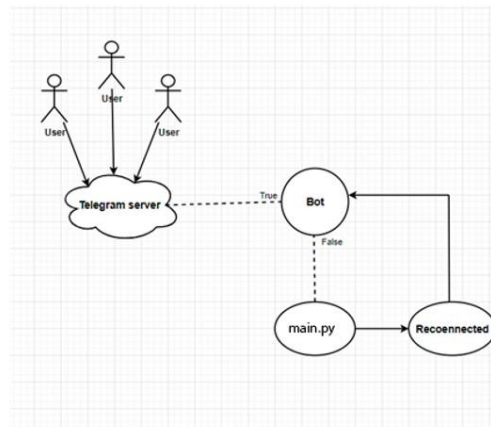
Схема функціонування телеграм-бота



6

Рисунок В.6 – Слайд № 6

Схематичне зображення роботи системи



7

Рисунок В.7 – Слайд № 7

Модулі використані в програмі

В реалізованій програмі були використані такі модулі: Telebot та Aiogram.

```

import telebot
from telebot import types

bot = telebot.TeleBot('1655018751:AAH713MXLOF2kdeOzh0BKuHCJ6L4j58SvZ0')
keyboard1 = telebot.types.ReplyKeyboardMarkup(True)

import aiogram
from aiogram.types import InlineKeyboardButton, InlineKeyboardMarkup

keyboard = types.InlineKeyboardMarkup(row_width=2)
  
```

8

Рисунок В.8 – Слайд № 8

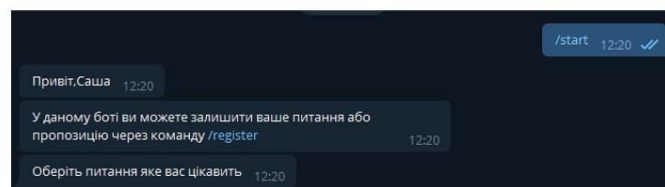
Головні функції програми

Функції	Опис
register	Реєструє повідомлення користувача у базі даних.
answer	Відправляє користувачу повідомлення з вітанням, описом вбудованої команди та проханням обрати питання яке цікавить користувача.
main	Головна функція у якій прописано як діяти після обрання користувачем тої чи іншої дії.
send	На будь який текст користувача надсилає вбудовану клавіатуру.
next_step_auditories	Отримує повідомлення користувача, шукає співпадіння в базі даних та виводить їх.

9

Рисунок В.9 – Слайд № 9

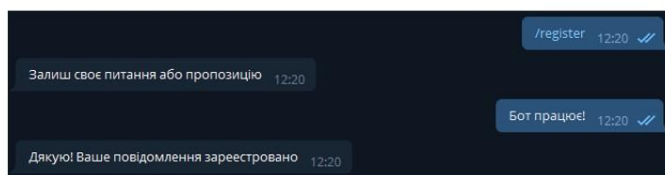
Реалізація "Команди /start"



10

Рисунок В.10 – Слайд № 10

Реалізація "Команди /register"



Відображення повідомлення у базі даних

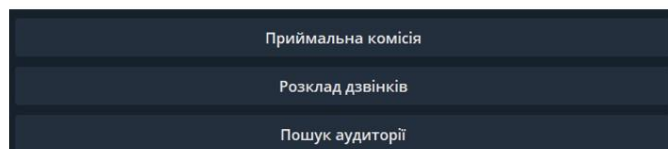
	user id	user name	user message
1	467291431	Саша	Бот працює!



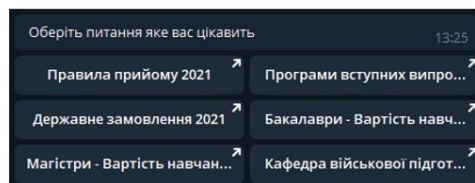
11

Рисунок В.11 – Слайд № 11

Реалізація вбудованих клавіатур



Розроблена Reply клавіатура



Розроблена Inline клавіатура



12

Рисунок В.12 – Слайд № 12

Висновки

В процесі виконання роботи було створено телеграм-бота для ЗВО.

Було розв'язано такі завдання:

- 1) аналіз предметної області;
- 2) розробка технічного завдання;
- 3) вибір середовища розробки;
- 4) розробка схеми взаємодії модулів програми;
- 5) конструювання телеграм-бота для крамниці;
- 6) тестування розробленого програмного забезпечення.

Були розроблені команди:

`/register` - Реєструє повідомлення користувача у базі даних;

`/start` - команда для привітання користувача та опис для подальшої роботи.

Також було розроблено 2 різні за типом клавіатури: Reply клавіатура та Inline клавіатура.

Серед особливостей проекту можна виділити використання сучасних бібліотек: Telebot та Aiogram.

Рисунок В.13 – Слайд №13