

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

## **Пояснювальна записка**

до бакалаврської кваліфікаційної роботи

на ступінь вищої освіти бакалавр

**на тему: «РОЗРОБКА ТЕЛЕГРАМ-БОТА ДЛЯ МОНІТОРИНГУ ЗАЯВОК  
КМДА МОВОЮ PYTHON»**

Виконав: студент 4 курсу, групи ПД-44

---

спеціальності 121 Інженерія програмного  
забезпечення

---

(шифр і назва спеціальності)

**Рибчук Н.В.**

---

(прізвище та ініціали)

Керівник

**Жебка В.В.**

---

(прізвище та ініціали)

Рецензент

---

(прізвище та ініціали)

Нормоконтроль

---

(прізвище та ініціали)

Київ – 2021

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ О.В. Негоденко

« \_\_\_\_ » \_\_\_\_\_ 2021 року

**З А В Д А Н Н Я**

**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

**РИБЧУКУ НАЗАРІО ВОЛОДИМИРОВИЧУ**

1. Тема роботи: «Розробка телеграм-бота для моніторингу заявок КМДА мовою Python»

Керівник роботи Жебка Вікторія Вікторівна, доцент, кандидат технічних наук затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки PyCharm Professional Edition 2021.1.1

3.2. Алгоритм дії бота

3.3. pyTelegramBotAPI

3.4. Google Drive API

3.5. Науково-технічна література, пов'язана з розробкою ботів на основі спеціальних API

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

- 4.1. Аналіз обов'язків розроблюваного бота
- 4.2. Дослідження класифікацій чат-ботів
- 4.3. Проектування інформаційного чат-бота
- 4.4. Розробка та тестування інформаційного чат-бота

5. Перелік графічного матеріалу

- 5.1.1. Дослідження сучасних бот-платформ
- 5.1.2. Функціональні тест кейси.
- 5.1.3. Програмні засоби реалізації
- 5.1.4. Огляд можливостей розробленого бота
- 5.1.5. Апробація результатів досліджень

6. Дата видачі завдання 19.04.2021

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів бакалаврської роботи   | Строк виконання етапів роботи | Примітка |
|-------|-------------------------------------|-------------------------------|----------|
| 1     | Підбір науково-технічної літератури | 19.04.21 – 22.04.21           |          |
| 2     | Аналіз існуючих прототипів          | 22.04.21 – 23.04.21           |          |
| 3     | Дослідження програмних засобів      | 23.04.21 – 01.05.21           |          |
| 4     | Моделювання об'єкту проектування    | 01.05.21 – 05.05.21           |          |
| 5     | Розробка функціоналу бота           | 05.05.21 – 07.05.21           |          |
| 6     | Вступ, висновки, реферат            | 07.05.21 – 08.05.21           |          |
| 7     | Розробка презентації застосунку     | 08.05.21 – 24.05.21           |          |
| 8     | Попередній захист роботи            | 25.05.21                      |          |

Студент

Керівник роботи





## РЕФЕРАТ

Текстова частина бакалаврської роботи 47с., 29 рис., 5 табл., 15 джерел.

PYCHARM, PYTHON, GOOGLE DRIVE API, TELEGRAM, PYTELEGRAMBOTAPI, ЧАТ-БОТ, MVC, ФРЕЙМВОРКИ, МЕСЕНДЖЕРА, HTTP.

*Об'єкт дослідження* – реалізація забезпечення моніторингу заявок користувачів за допомогою месенджера.

*Предмет дослідження* – чат-бот для моніторингу заявок.

*Мета роботи* – розробка функціонального чат-боту на платформі Telegram.

*Методи дослідження* – методи оптимального управління, оброблення та аналіз отриманих результатів.

*Наукова новизна* – на основі існуючих методах оброблення та аналізу отриманих результатів моніторинг заявок користувачів.

В роботі розглянуто основні відомості про існуючі месенджери, їх характеристики. Проаналізовано дослідження сервісів для створення чат-ботів. Також був спроектований, розроблений та протестований інформаційний чат-бот на платформі Telegram.

## ЗМІСТ

|  |           |
|--|-----------|
| <b>ВСТУП .....</b>   | <b>8</b>  |
| <b>1 АНАЛІЗ ДОСЛІДЖЕННЯ РОЗРОБКИ ЧАТ-БОТІВ.....</b>                                  | <b>9</b>  |
| 1.1 Сучасні технології месенджерів .....   | 9         |
| 1.2 Дослідження сучасних бот-платформ .....  | 11        |
| 1.2.1 Платформа Telegram. ....   | 14        |
| 1.3 Постановка задачі дослідження. ....  | 17        |
| <b>2 ДОСЛІДЖЕННЯ КЛАСИФІКАЦІЙ ТА МОЖЛИВИХ ВАРІАНТІВ<br/>РОЗРОБКИ ЧАТ-БОТІВ .....</b> | <b>19</b> |
| 2.1 Дослідження класифікацій чат-ботів. ....   | 19        |
| 2.2 Дослідження сервісів для створення ботів. ....                                   | 21        |
| 2.3 Дослідження фреймворків для створення чат-бота. ....                             | 24        |
| 2.4 Складність створення чат-бота. ....  | 25        |
| <b>3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЧАТ-БОТА.....</b>                                   | <b>27</b> |
| 3.1 Вимоги до інформаційного чат-бота.....   | 27        |
| 3.2 Архітектура інформаційного чат-бота.....   | 27        |
| 3.3 Моделювання варіантів використання інформаційної системи. ....                   | 28        |
| <b>4 РОЗРОБКА ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОГО ЧАТ-БОТА .....</b>                        | <b>34</b> |
| 4.1 Реєстрація чат-бота. ....  | 34        |
| 4.2 Реєстрація в Google Drive Filestorage.....                                       | 38        |
| 4.3 Розробка основного функціоналу чат-бота. ....                                    | 42        |
| 4.4 Тестування чат-бота.....   | 50        |
| <b>ВИСНОВОК .....</b>  | <b>56</b> |
| <b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>  | <b>57</b> |
| <b>ДОДАТОК А.....</b>  | <b>59</b> |

## ВСТУП

Оскільки сучасний світ стрімко розвивається кожний день, технології також не стоять на місці. Через це, щодня з'являються нові тенденції та ідеї, багато з них сильно впливають на ІТ-індустрію, що безпосередньо впливає на наше повсякденне життя.

До недавнього часу були дуже популярними настільні додатки, проте на даний час лідерство отримали чат-боти, які мають великі перспективи в різних сферах нашого життя. Використання чат-ботів є дуже актуальним для ведення бізнесу в онлайн режимі.

Завдяки роботам бізнесу вдається оптимізувати процеси, зменшити витрати та також охопити більше аудиторії. Перевагою чат-бота є те, що користувач відразу отримує відповідь на свій запит.

Актуальність роботи пояснюється зростанням популярності використання месенджерів. На платформах месенжера присутня можливість створення та розгортання власних чат-ботів, які допомагають автоматизувати рутинні задачі такі як: пошук актуального курсу валют, пошук погоди, новин, купівля залізничних квитків, автоматизація внутрішніх процесів компанії.

*Об'єкт дослідження* – реалізація забезпечення моніторингу заявок користувачів за допомогою месенжера.

*Предмет дослідження* – чат-бот для моніторингу заявок.

*Мета роботи* – розробка функціонального чат-боту на платформі Telegram.

Були поставлені наступні завдання, базуючись на встановленій меті:

- аналіз предметної області;
- аналіз технологій розробки чат-ботів;
- розробка та тестування чат-бота.



# 1 АНАЛІЗ ДОСЛІДЖЕННЯ РОЗРОБКИ ЧАТ-БОТІВ

## 1.1 Сучасні технології месенджерів

Месенджер - це клієнтська програма, необхідна для миттєвого обміну повідомленнями. Крім того, багато сучасних месенджерів дозволяють обмінюватися не тільки повідомленнями, але і фотографіями, відеороликами, файлами, смайликами і тому подібне. Найчастіше месенджери підтримують голосовий і навіть відеозв'язок, причому для того, щоб з'єднатися з абонентом, потрібен лише доступ до мережі інтернет. Якщо дзвінок здійснюється через інтернет, де б користувач не знаходився, дзвінок буде безкоштовним, плата стягується тільки за трафік (з пакетними тарифами операторів стільникового зв'язку).

Деякі додатки можна встановити як на смартфон, так і на комп'ютер. Найбільшою перевагою додатка-месенджера є те, що він дозволяють користувачеві бути на зв'язку цілодобово. Як, запитаєте ви? По суті, додаток ніколи не вимикається і знаходиться в фоновому режимі. Головне при цьому бути підключеним до мережі інтернет через Wi-Fi або мобільний інтернет. В цьому випадку, якщо ваш друг відправить вам повідомлення або зробить дзвінок, ви в цю ж мить його отримаєте.

У світі налічуються десятки додатків для спілкування, кожна з яких затребувана в певному регіоні: Whatsapp - в Європі, Америці, частини країн Близького Сходу, Viber - в країнах пострадянського простору, Line - в Азії.

Є платні і безкоштовні програми, месенджери для конкретних операційних систем, корпоративні та загальнодоступні варіанти.

Перше місце в рейтингу світових лідерів займає безкоштовний месенджер WhatsApp (Вотсапп) - понад 5 мільярдів скачувань в Google Play.

Особливості месенджера WhatsApp:

- підтримка операційних систем Android, iOS, Windows Phone;
- створення групових чатів для спілкування відразу з декількома друзями;

- здійснення аудіо та відеодзвінків;
- пересилання файлів і зображень;
- необмежений обмін повідомленнями;
- видалення відправлених повідомлень у себе і співрозмовника;
- наявність версії для ПК, що працюють на Windows 8 і Windows 10;
- наявність веб-версії додатка для відправки повідомлення з комп'ютерів на більш ранніх версіях Windows.

Наступну сходинку рейтингу світових месенджерів посідає Viber, який налічує понад 500 мільйонів завантажень в Google Play і найбільш поширений в країнах пострадянського простору.

Особливості месенджера Viber:

- підтримка операційних систем Android, iOS, Windows Phone;
- здійснення аудіо та відеодзвінків;
- пересилання файлів і зображень;
- необмежений обмін повідомленнями;
- видалення надісланих повідомлень у себе і співрозмовника;
- наявність версії для ПК;
- створення публічних каналів і чатів.

Телеграм стрімко набирає популярність в світі відстаючи від Viber всього лише на декілька тисяч скачувань. На даний момент Телеграм налічує понад 500 мільйонів установок в Google Play. Аплікація орієнтована на всі країни світу і вважається одним з найбезпечніших месенджерів з усіх існуючих.

Особливості месенджера Телеграм:

- підтримка операційних систем Android, iOS, Windows Phone;
- наявність версії для ПК;
- видалення відправлених повідомлень у себе і співрозмовника;
- здійснення аудіозвонків, але відсутність відеодзвінків;

- створення ботів і публічних каналів на Telegram з подальшою їх монетизацією;
- пересилання файлів і зображень;
- необмежений обмін текстовими повідомленнями;
- можливість ділитися геолокацією;
- можливість створення голосувань;
- можливість відправлення запланованих повідомлень;
- повна безпека для користувачів.

## **1.2 Дослідження сучасних бот-платформ**

Чат-бот (англ. Chatbot) - це програма, яка імітує реальну розмову з користувачем. Чат-боти дозволяють спілкуватися за допомогою текстових або аудіо-повідомлень на сайтах, у месенджерах, мобільних додатках або по телефону. Сьогодні чат-боти стали сучасними інструментами спілкування, почали широко застосовуватися в різних сферах людського життя з метою встановлення контакту з користувачами всесвітньої павутини інтернет. Найбільшу популярність чат-боти стали отримувати тоді, коли їх почали інтегрувати в соціальні мережі та месенджери такі як Telegram, Viber, Facebook, WhatsApp та інші.

Поняття «чат-бот» походить від двох англійських слів: tochat - невимушена розмова в мережі Інтернет, bot (robot) - скорочено робот, з чого випливає, що це роботи призначені для здійснення комунікацій з користувачами в мережі Інтернет, які виконують дії відповідно до запрограмованого сценарію. Така програма інтернет-комунікації як чат-бот заснована на сучасних технологіях. В її основі лежать такі інформаційно-комунікаційні технології, як штучний інтелект, нейронні мережі, що дозволяє наближати комунікації чат-ботів до людського віртуального спілкування і вирішувати з їх допомогою комерційні та маркетингові задачі.

Судячи з поточної оцінки стану ринку, чат-боти, особливо для месенджерів – дуже перспективний напрям, який в даний час перспективно розвивається (рис. 1.1).

## Chatbots and AI

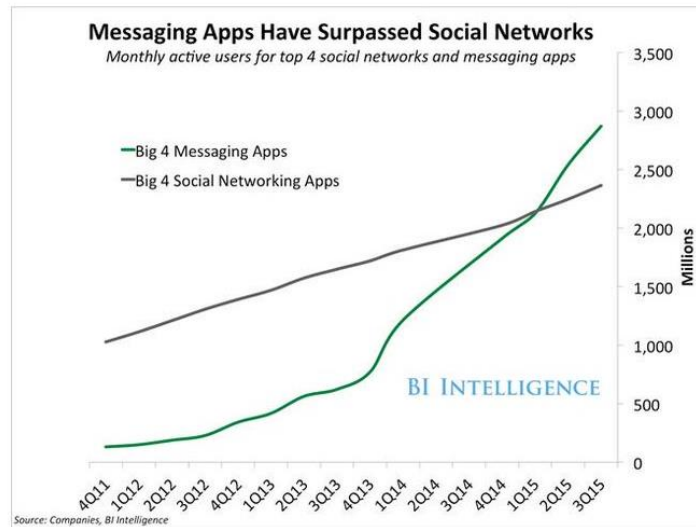


Рисунок 1.1 – Тенденції ринку чат-ботів

За оцінкою світових аналітиків у найближчому майбутньому чат - боти будуть мати все більшу і більшу значимість для різного виду бізнесів. Вони можуть повністю замінити класичні пошукові системи і соціальні мережі. Переваги ботів проявляються у зручності спілкування людини з комп'ютерною мережею, швидкість реакції відповіді на запит та можливості адаптації під потреби користувача. Прикладом може бути Facebook чат-бот M, який на даний момент знаходиться в режимі бета-тестування – це персональний цифровий асистент інтегрований всередині менеджера, що виконує тривіальні задачі в тому числі рекомендації з надання послуг. Технічно Facebook-M заснований на сучасних технологіях машинного навчання (глибинні нейромережі). Крім Facebook, в напрямку розвитку технологій чат-ботів активно рухаються і розвиваються такі компанії гіганти як Google і Apple.

Основні напрямки використання чат-ботів:

1. У сфері бізнесу та електронної комерції. У даних сферах чат-боти мають найбільшу популярність та багатий функціонал і можуть вирішувати багато завдань, включаючи онлайн-консультації, рекламні комунікації, продажу та інше.

2. У сфері надання інформації. Такі чат-боти використовуються в електронних засобах масової інформації для надання користувачам цікавих новин. Наприклад, доктор-бот може порадижити лікування на основі описаних симптомів або підказати адреси найближчих лікарень та медичних центрів. В даний час широку популярність набувають інформаційні чат-боти, які надають користувачам актуальну статистику захворюваності та розповсюдження інфекції COVID-19 на території України та за її межами.

3. У сфері навчання іноземним мовам та інших дисциплін, так звані боти-вчителі. Вони надають навчальний матеріал (найчастіше це невеликі за обсягом тексти) з дисципліни, а потім оцінюють знання учнів. В даному етапі розвитку людства чат-боти допомагають самостійно вивчати іноземні мови.

4. У розважальній сфері. Чат-боти в сфері розваги можуть вести діалог з користувачем, надавати йому цікаву інформацію, грати з ним в ігри і виконувати інші розважальні функції. Також чат-ботів можна інтегрувати у різні бесіди та групи для колективних ігор.

Даний дипломний проект буде корисним та актуальним у адміністративному використанні. Адже використання бота в інформаційній системі Telegram дає набагато більше переваг ніж в конкурентів та також є менш затратним, якщо його зрівняти з негативним додатком чи веб сайтом. В нашому рідному Києві ну і в цілому в Україні є багато проблем з інфраструктурою. Наприклад проблеми з дорожнім покриттям, відкриті люки каналізації, розбиті лавочки в скверах та парках, тощо.

Тому була поставлена задача автоматизувати роботу пошуку проблемних місць інфраструктури в місті Києві. Саме тому було вирішено створити чат-бота, який би збирав дані проблем, для подальшого перенаправлення до КМДА. Чат-бот є ефективним та зручним у використанні і не потребує додаткового навчання

користувачів його використання. Також завдяки зручним функціям в месенджері бот має змогу працювати з картами , а також підтримувати активне спілкування з користувачем.

### **1.2.1 Платформа Telegram.**

Telegram - це безкоштовний месенджер, за допомогою якого можна спілкуватися з іншими користувачами, обмінюватися файлами різних форматів і різних розмірів.

Програму розробив Павло Дуров, російський програміст, один з авторів популярної мережі ВКонтакте. Месенджер використовує спеціальний протокол шифрування MTProto - розроблений самим Миколою Дуровим. Захист стосується як передачі повідомлень, так і авторизації користувача. Безпека на високому рівні також підтримується, оскільки сервери, на яких зберігаються дані користувачів, знаходяться в різних місцях планети: Лондон, Сінгапур, Сан-Франциско.

Послуга надається безкоштовно, її творці вклали гроші в її створення заради вільного та безпечного спілкування. Вони не мають офіційного веб-сайту, гучних рекламних кампаній, своїх популярних сторінок у соціальних мережах. Месенджер просувається клієнтами.

На початок 2021 року кількість користувачів у системі перевищила 400 мільйонів. Щодня додається близько 1,5 мільйона учасників. Додаток працює 13 мовами. Він є конкурентом таких гігантів, як Viber, WhatsApp тощо.

За допомогою Telegram можна:

- спілкуватися в інтернеті з користувачами по всьому світу, включаючи високоякісні звукові дзвінки;
- надсилати та отримувати файли різних форматів (фото, аудіо, відео та інші) об'ємом не більше 1,5 ГБ;
- створювати групові чати до 200 учасників, включаючи засекречені;

- організувати спілкування створюючи супер-групи: до 10 000 людей. Є адміністратор, який може модерувати публікації та застосовувати обмеження до учасників. Використовувати імена користувачів: функція дозволяє знайти людину, не знаючи її номера телефону;
- знайти групи за інтересами та взяти участь в їх обговореннях;
- записати та надіслати звукове повідомлення;
- створювати власні канали. Будь-які від ділових вебінарів та політичних дебатів до рецептів та розваг для дітей;
- слухати безкоштовно музику і навіть без Інтернету;
- використовувати хмарне сховище даних;
- надсилати, створювати самостійно, завантажувати наклейки безкоштовно (на відміну від інших месенджерів);
- використовувати автоматизованих помічників: ботів.

Месенджер має ряд переваг. Серед основних:

- висока робоча швидкість: незважаючи на розташування серверів у різних частинах світу, доставка повідомлення займе частку секунди;
- автоматичне додавання списку користувачів до служби через телефонну книгу;
- легкий спосіб реєстрації – потрібен лише номер телефону;
- весь функціонал – безкоштовний;
- сервіс постійно оновлюється. Не має аналогів;
- високий рівень безпеки та захист від піратства та кореспонденції;
- необмежено для кількості повідомлень, що відправляються на день; кількості файлів, завантажених у хмару; час використання;
- сумісний з усіма основними операційними системами, включаючи мобільні та навіть CLI інтерфейс для Unix систем.

З основних недоліків, які були виявлені і досі не усунені є те, що спілкування зі службою технічної підтримки здійснюється лише англійською та російською мовами.

Донедавна інтерфейс не підтримував українську мову. Для адаптації використовували сторонні русифікатори, тощо. Питання було вирішено в березні 2018 року, і українська мова є одним із можливих варіантів на вибір.

Спеціальний захист листування та відмова надати владі та поліцейським органам зміст розмов є причиною того, що терористичні групи повідомляють через цей комунікатор продаж наркотиків та інші дії, заборонені законом. Це породжує невдоволення деяких людей та звинувачення проти компанії у співучасті з злочинцями.

Неможливість контролювати і зупиняти розповсюдження нелегальної інформації призводить до блокування програми в деяких країнах. Тому в Китаї були введені обмеження під час страйків та урядових демонстрацій та в Ірані за відмову допомагати в нагляді. Незважаючи на високі заходи безпеки, можна отримати частину інформації про користувача наприклад його номер телефону. Це не сприяє повній анонімності профілю.

На каналі Telegram ви можете передавати вміст іншого сайту або соціальної мережі, робити власні повідомлення від адміністратора. Створити власний канал досить просто, головне наповнити його цікавими відео та новинами. Користувач може вказати параметри відображення в налаштуваннях. Є два варіанти: оновлення каналів відображатимуться в розділі чатів, як повідомлення звичайних людей; або тихий режим. Тоді ви можете зайти на канал і переглянути новини у будь-який зручний час.

Канали можуть бути як загальнодоступними (для всіх), так і приватними (доступ відкривається адміністратором за запитом).

Telegram боти - це спеціальні віртуальні помічники, керовані програмою. Вони можуть виконувати різноманітні дії за запитом: шукати курси валют, конвертувати документи, відображати прогнози погоди, Google Analytics, новини, графіки програм та багато іншої інформації.

Є два види Telegram ботів:

- звичайні : їм надсилаються команди та пошукові запити;



- інтегрований: дозволяє шукати, не виходячи з чату, і завантажувати в нього результати пошуку. Наприклад, бот @gif запропонує зображення із використанням ключових слів, введених після його імені.

Все більше компаній просувають свій бізнес в Інтернеті. Створюються веб-сайти, просувають бізнес в соціальних мережах, на форумах. Месенджери - не виняток. Бізнес у Telegram отримав додаткові можливості для спілкування зі споживачами.

Для просування використовуються численні методи, серед яких:

- боти або жива людина для підтримки клієнтів у нових питаннях;
- канали для бізнесу - інформування про нові товари, акції, відео про використання товарів;
- спілкування всередині компанії - постановка завдань, обговорення, звіти, плани. Застосовується як на рівні всієї компанії, так і в підрозділах;
- зберігання та доступ до необхідних документів;
- інтернет-каталог - за допомогою бота вибирає продукт безпосередньо в додатку;
- прийом платежів. Доступні платіжні системи: PaymentWall, Stripe, Яндекс.Каса, Раусом. Ви можете придбати те, що вам подобається, відразу і не переходячи на інші програми - швидко і зручно.

Враховуючи той факт, що Telegram - це послуга, яка лише набирає популярності, це правильний крок, щоб розпочати просування свого бізнесу на ній. Молода та просунута аудиторія, функціональність, безпека та відсутність платежів роблять Telegram одним із найсучасніших інструментів для бізнесу.

### **1.3 Постановка задачі дослідження.**

Об'єкт дослідження – реалізація забезпечення моніторингу заявок користувачів за допомогою месенджера

Мета роботи – розробка функціонального чат-боту на платформі Telegram.

Були поставлені наступні цілі:

- вивчити існуючі підходи до розробки ботів;
- дослідити існуючі проблеми чат-ботів;
- визначення програмної платформи, а саме фреймворків та інтерфейсу прикладні рішення та бібліотеки API з великою кількістю інструменти: функції, класи, структури, придатні для сучасного розвитку Ботів;
- вивчити методи машинного навчання для вдосконалення ботів, та взаємодія з користувачем;
- розробка додатка.

## 2 ДОСЛІДЖЕННЯ КЛАСИФІКАЦІЙ ТА МОЖЛИВИХ ВАРІАНТІВ РОЗРОБКИ ЧАТ-БОТІВ

### 2.1 Дослідження класифікацій чат-ботів.

В останні роки з'явилося досить велика кількість чат-ботів, але незважаючи на кількість, їх усіх можна розділити на дві великі категорії залежно від технології. Технології, що використовуються для інтелектуального діалогу: програми обробки природних мов NLP (Natural Language Processing) та програми на основі мислення конкретних випадків CBR ( Cased-Based Reasoning).

Чат-боти використовують технологію обробки природної мови. Вони починають роботу з аналізу підказки користувача при скануванні. Як результат пропозиції розбиваються на нетермінальні і термінальні символи, знайомі програмі. Чат-боти з використанням технології NLP не мають бази даних з чітко визначеними шаблонами відповідей. Синтезується відповідь на введення пропозиції користувачем кожен раз на основі правил використовуваної граматики, на основі структури тексту, на основі раніше введеного користувачем, а також містяться в ньому ключових слів.

Деякі з цих роботів не мають вбудованих граматик. Через це вони використовують індуктивний метод граматичного виводу. Це дозволяє «налаштуватися» на його шляху і вчитися під час діалогу з користувачем. Найпопулярнішим серед інших представників програм кейс-орієнтованого мислення є програма A.L.I.C.E. Програма була створена в кінці 1990-х. Для написання A.L.I.C.E. спеціально розробили мову програмування AIML. AIML - це модифікована версія XML.

Засновники розробили мову AIML для надання функцій, необхідних для отримання і обробки знань; для відповідних даних структури моделі, а також для формування вихідних сигналів за схемою «стимул-реакція». В теперішній час мова програмування AIML, її специфікації і інтерпретатори для перекладу тексту програми

на стандартні об'єктно-орієнтовані мови програмування безкоштовно поширюються на офіційному сайті A.L.I.C.E.

Програма реєструє всі репліки користувачів, у відповідь на які були використані репліки "за замовчуванням", тому розробники чат-ботів мають доступ до журналів розмов і розширюють та вдосконалюють базу знань. Мова програмування AIML також дозволяє включати команди, написані іншими мовами, такими як, наприклад, Javascript або C ++, що значно збільшує сферу застосування програми та дозволяє інтегрувати її у велику кількість програм. Чат-боти A.L.I.C.E. може підтримувати діалог з кількома користувачами одночасно і мати структуру клієнт-сервер.

Вбудований бот - це бот з елементами ШІ, його можна "годувати" великим обсягом даних, в якому він буде шукати інформацію краще та швидше, ніж людина. Корисно там, де вам потрібно орієнтуватися у величезних обсягах даних, таких як документи страхових або юридичних фірм.

В ідеалі взаємодія між користувачем та ботом повинна бути зручною та плавною. Цього можна досягти завдяки ретельному плануванню схеми розмови, особливо для розмовних ботів. Зокрема, бот міг повторити введені команди, щоб користувач знав, що його «почули» і зрозуміли правильно. Боти також повинні мати можливість виявити, коли діалог зайшов у глухий кут, і надати підказки про можливу подальшу взаємодію.

Боти можуть мати елементи графічного інтерфейсу для прискорення взаємодії. Дозволяючи глобальну перевірку введення для ключових слів, пов'язаних з навігацією («допомога», «назад», «скасувати», «почати спочатку», «вихід» тощо), ви можете уникнути створення занадто «дурного» бота. Такі інструменти, як BotMock, можуть допомогти вам змодельовати, як користувач взаємодіє з ботом. Крім того, на багатьох платформах доступні готові комунікаційні протоколи, яких повинні виконувати боти, наприклад, Facebook має набір простих порад щодо організації розмов. Чат-боти відрізняються один від одного залежно від завдань, які вони повинні вирішити.

Для комерційного чат-бота, який потрібен для роботи в певному діалоговому просторі, темі, зоні відповідальності чат-бота, немає необхідності використовувати технологію машинного навчання. Для чат-ботів цього типу використовуються основні технології обробки мови:

- виділення ключових слів або іменованих сутностей;
- сегментація (це розділення на речення).

Як і технології для створення регулярних граматики типу `input pater` - це шаблон відповідної реакції, приклад з AIML:

- масив (змінні) для вивчення або запам'ятовування контексту;
- регулярні вирази;
- рекурсія;
- цикли та інше.

## **2.2 Дослідження сервісів для створення ботів.**

Найпростішим варіантом є інтерактивна форма, найскладнішим є програма зі штучним інтелектом, що імітує співрозмовника, здатна вести бесіду на будь-яку тему.

Існують шаблони ботів для деяких сфер діяльності - магазини, видавництва, компанії з нерухомості. Ви можете створити бота всього за 7 хвилин. Початок роботи безкоштовний, і буде доступна основна функціональність. Платні підписки надають ще більше можливостей. Ціна коригується залежно від кількості користувачів ботів.

На сайті є повний путівник про те, як розпочати роботу та зробити свого першого робота. Все описано дуже докладно, чудова допомога тим, хто не уявляє, як розроблені онлайн-помічники. Dialogflow використовує машинне навчання, щоб робочі місця могли розпізнавати мову користувачів і вчитися, спілкуючись з ними.

Для створення ботів без використання служби вони використовують API платформи. Наприклад, API Bot - це інтерфейс HTTP для роботи з ботами в Telegram.

Кожен бот - це спеціальний обліковий запис, створений для автоматичної обробки та надсилання повідомлень.

Є два логічно-протилежні способи отримання оновлення бота: тривале витягування - програма автоматично опитує сервер Telegram щодо будь-яких оновлень для бота. За замовчуванням 100 мс; webhook - сервери Telegram самі інформують програму на сервері, щойно з'являються оновлення. Вхідні оновлення зберігатимуться на сервері до їх обробки, але не більше 24 годин. Незалежно від способу отримання оновлення, у відповідь надсилається об'єкт Update, серіалізований у JSON.

Усі запити до API Telegram Bot повинні бути зроблені через HTTPS у такій формі: [https://api.telegram.org/bot<token>/METHOD\\_NAME](https://api.telegram.org/bot<token>/METHOD_NAME).

Приклад відповіді на команду ботом BotFather, який дозволяє створити бота для месенджера Telegram, наведено на рисунку 2.1.

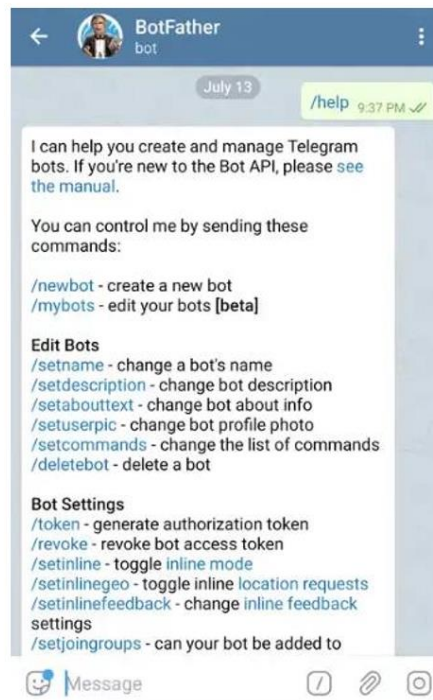


Рисунок 2.1 – Результат обробки команди «/help» ботом BotFather.

У месенджерах працює багато чатових ботів для вирішення різних завдань, деякі з них будуть відображені на малюнках.

Payrbbot - бот ПриватБанку для переказу грошей на Telegram. «Мої платежі» - бот для оплати комунальних рахунків через Приватбанк у Facebook.

" Зоряна" - бот "Київстар" у Telegram, який консультує з питань зв'язку та допомагає визначати тарифи та послуги.

Whereisnpbot - неофіційний бот Telegram, який відображає місце розташування пакета "Нова пошта".

QTrackerBot - бот Telegram, який відстежує пакети від Укрпошти, Нової пошти, Deliveri, Most-Express.

iGovBot - це телеграмний бот, який здійснює пошук на порталі державних служб iGov для отримання інформації про компанію під назвою, кодом ОКРО або іменем директора та повідомляє про зміни в тих компаніях, на які підписаний користувач.

OpenDataBot - це мультиплатформенний бот, який контролює реєстраційні дані українських компаній та судовий реєстр.

Переваги використання ботів:

По-перше, це дешево. Для компаній вигідніше використовувати чатових ботів, ніж наймати людські ресурси. Подібно до людей, різні чат-боти можуть виконувати широкий спектр завдань: легше надавати поради в США та робити покупки та замовлення замість клієнтів. Роблячи все це, чат-бот ніколи не втомлюється і не хворіє. Таким чином, вони не вимагають зворотних відпусток, лікарняних та додаткових виплат за роботу в нічні зміни, на відміну від працівників-працівників.

По-друге, миттєва відповідь, відсутність черг. Чат-боти швидко дають точні відповіді. Потрібно менше секунд, щоб зіставити запитувану інформацію з базою знань та сформулювати результат. Вони не витрачають зайвого часу на роздуми, пошук, згадування та відкриття багатьох вкладок браузера. Більше того, чат-боти можуть вести багато розмов одночасно, тому клієнтам не потрібно чекати в черзі, що вони

ненавидять робити. Швидкий час відгуку забезпечує чудову взаємодію з користувачем.

По-третє, доступність цілодобово та без вихідних. Чат-боти дозволяють компаніям постійно підтримувати зв'язок зі своїми клієнтами і не втрачати можливість отримати прибуток. Для обідніх перерв на форумі чат-ботів та неробочого часу. Також споживачі раді, що вони можуть отримати допомогу в будь-який час дня, тижня чи року - саме тоді, коли їм це потрібно, тоді як їх запитання досі актуальні, а інтерес не зник.

### **2.3 Дослідження фреймворків для створення чат-бота.**

API для створення чат-бота та сервіси для їх використання надаються такими іноземними компаніями, як Pandora-bots та Synthetic Intelligence Network: перша надає доступ до свого API на основі досить простої XML-подібної мови сценаріїв AIML, яка реалізує всі технології, необхідні для реалізація простого комерційного чату - мислення. Вартість використання API Pandotabots становить від 9 \$ до 999 \$, залежно від кількості підтримуваних ботів. Synthetic Intelligence Network пропонує мову SIML, концептуально подібну до AIML і створену на її основі, а також власне середовище розробки ботів - Syn Chatbot Studio для .NET.

Досить відомими фреймворками для створення власних ботів є Errbot у Python та Hubot у CoffeeScript під Node.js. У той же час були написані боти як для Errbot, так і для Hubot, які можна перенести в Slack, Telegram, IRC тощо.

Структури обробки природної мови. Для обробки природних мов на рівні, необхідному для створення чат-бота для полегшення роботи консультантів-людей у будь-якій службі, багато технологій з відкритим чарівництвом можуть бути придатними в базовій формі. Подібний фреймворк також може бути створений на основі, наприклад, Apache OpenNLP. Не забувайте про існуючі API, які забезпечують вибір сутностей тощо, з огляду на їх використання для чатових ботів - вже згадані тут



ari.ai та wit.ai. Чат-боти - дуже перспективна сфера. Їх активне використання в месенджерах та як цифрових помічників у смартфонах, швидше за все, призведе до популяризації парадигми UX для обміну повідомленнями як інтерфейсу. Вже ключові гіганти Facebook, Google, Apple розробляють чат-боти для персональних помічників.

## **2.4 Складність створення чат-бота.**

Складність створення чат-бота полягає в тому, що не дуже легко налаштувати пошук фрази, яку видає користувач, в основі виразів та слів, які використовує бот. Щоб розмовляти зрозумілою мовою, вам потрібна специфічна пошукова машина на основі узагальнених шаблонів реплік, яка здатна розпізнавати подібний шаблон, навіть якщо деякі слова у запиті відсутні або замінені іншими.

Механізм чату повинен шукати відповідну підказку, пам'ятаючи контекст попередньої розмови. Але головна проблема полягає у висвітленні всього різноманіття вхідних сигналів, оскільки людина може запитати що завгодно і в будь-якій формі.

Будь-яка пошукова система має дві характеристики: точність і повноту. Точність у більшості ботів завжди на високому рівні - якщо вдалося розпізнати репліку, тоді відповідь буде точною. Що стосується повноти, то досить важко гарантувати розпізнавання великої частки вхідних фраз для такої пошукової машини, оскільки людина, яка вільно володіє мовою, може придумати будь-які вхідні фрази, їх різноманітність настільки велике, що дуже важко на все відповісти.

З огляду на це, слід мати на увазі, що бот з часом зустріне з "хвилею" нових питань на основі деяких подій. У деяких випадках можна передбачити потік копій, але потрібно швидко реагувати.

Можливості ботів:

- можливість збору відгуків;

- сучасні технології обробки мовлення дають змогу вести складно організований діалог з людиною;
- інтеграція із зовнішніми джерелами даних;
- реалізація на мобільних платформах;
- зручний та зручний інтерфейс систем діалогу.
- вирішення ділових проблем, таких як технічна підтримка, допомога, просування.

Відмінності між автоматизованим та природним спілкуванням полягають у наступному:

- рівні та методи розуміння репліки співрозмовника віртуальним співрозмовником;
- у загальній схемі та глибині всього діалогу ;
- в характері, стилі та інших особливостях формування відповіді.

Важливий аспект, що вимагає уваги при розробці:

- тестування, що приводить до збою зв'язку;
- діалогічність віртуального співрозмовника;
- набір цілей інтерв'ю, будь то розвага, консультація, рекомендація або навіть реклама;
- глибина володіння темою (предметною областю);
- варіативність, стилістична структура та інші мовні вимоги до відповідей віртуального співрозмовника;
- комунікативна організація спілкування.

Віртуальний співрозмовник повинен завжди швидко реагувати, мати багато даних, не відволікатися, використовувати повторювані формулювання, замінювати дані з баз даних та служб, робити правильне комунікативне позиціонування.

## 3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЧАТ-БОТА

### 3.1 Вимоги до інформаційного чат-бота.

Інформаційний чат бот повинен відслідковувати проблеми інфраструктурі міста Києва для подальшого їх вирішення методом написання заявок в КМДА. Бот повинен мати легкий та зрозумілий інтерфейс. Користувач повинен відразу розуміти як ним користуватися. Даний чат-бот включає в себе найновітнішу інтеграцію сховища медіаданих Google Drive, інтеграцію з легкою базою даних SQL Lite, а також бот буде зручним, швидким, мобільним та актуальним.

З аналізу основних задач були сформовані наступні вимоги:

- система повинна працювати на всіх версіях Telegram включаючи всі операційні системи та всі пристрої на які може бути встановлений додаток;
- система повинна завжди підтримувати зв'язок з користувачем, не вводити його в безвихідну ситуацію;
- система повинна записувати всі дані, які надасть користувач в базу даних та віддалене медіасховище;
- перехід до головного функціоналу подачі заявки, повинен бути доступний з любого рівня;
- система повинна стабільно функціонувати та працювати в режимі 24/7.

### 3.2 Архітектура інформаційного чат-бота.

Для того, щоб зрозуміти структура чат-бота, було створено абстрактну модель рис. 3.1 – Архітектура бота. Система включає в себе Telegram як клієнт і як API, Google Drive API, базу даних SQL Lite.

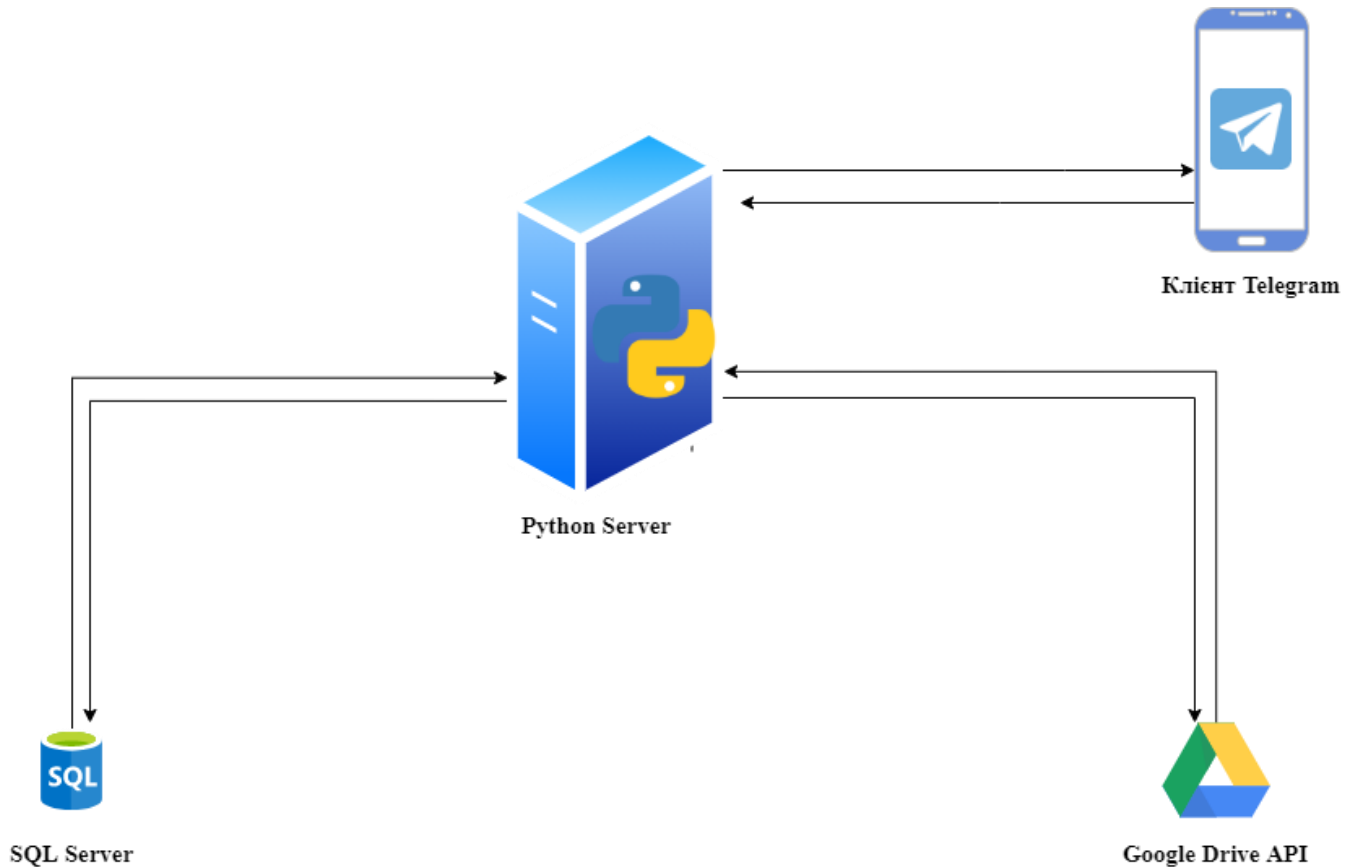


Рисунок 3.1 – Архітектура бота

### **3.3 Моделювання варіантів використання інформаційної системи.**

В даному розділі буде схематично розглянуто основні варіанти використання чат-бота.

Першим представлений кейс реєстрація проблеми в системі згідно рисунку 3.2 – Реєстрація проблеми. Даний кейс – це основна модель користувацького сценарію, яка задумувалась з самого початку. В цій схемі показаний основний процес використання системи.

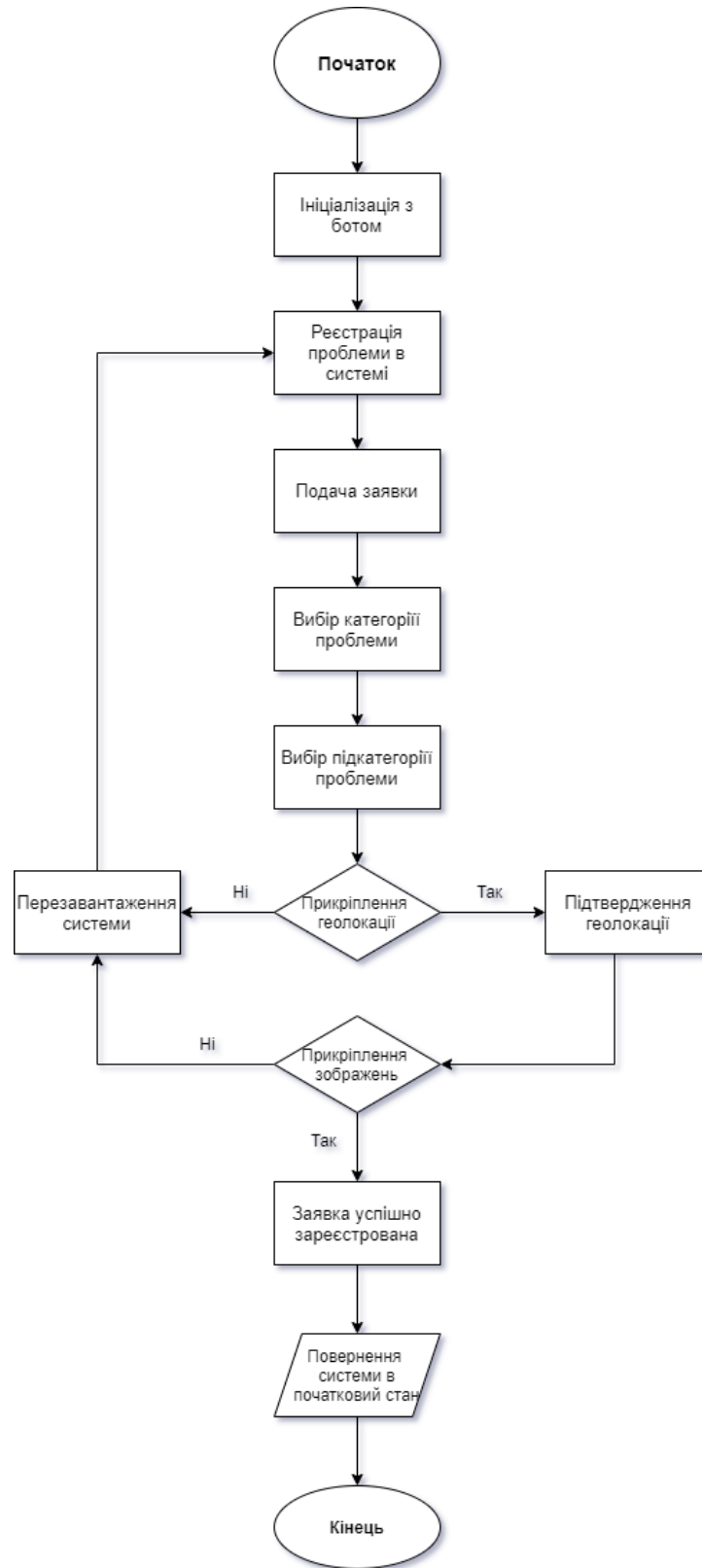


Рисунок 3.2 – Реєстрація проблеми

Наступним кейсом – моделювання реєстрації заявки проблеми в системі вказуючи свою категорію проблеми рисунок 3.3 - Реєстрація проблеми в системі вказуючи свою категорію проблеми.

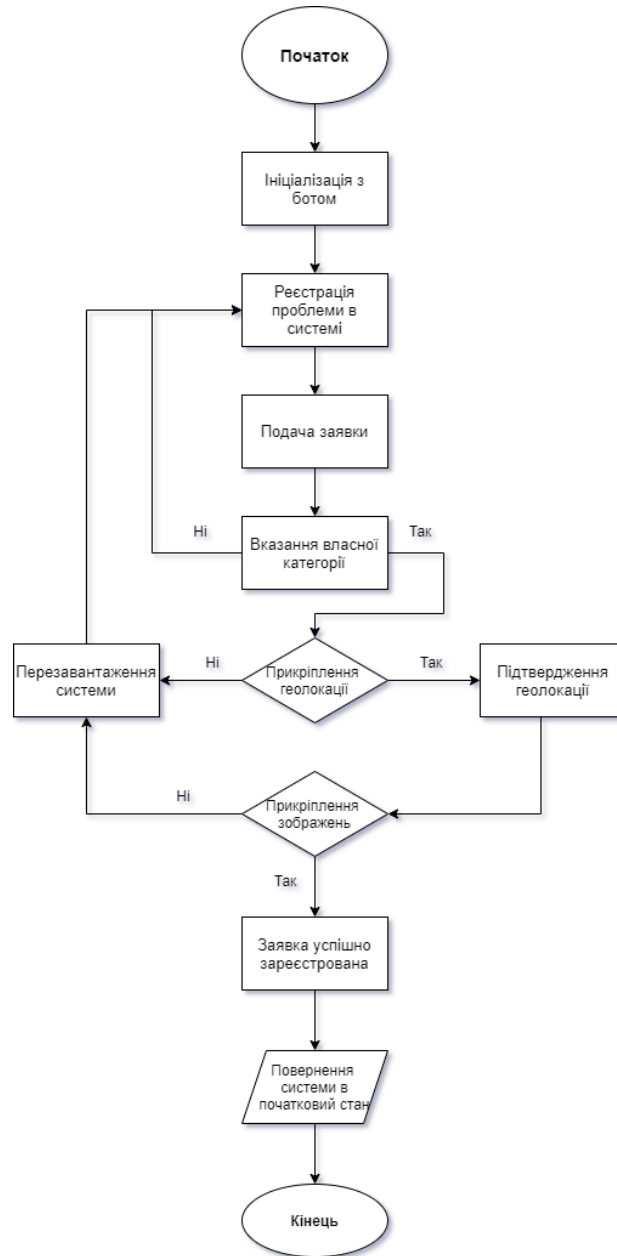


Рисунок 3.3 - Реєстрація проблеми в системі вказуючи свою категорію проблеми.

Третім кейсом є моделювання ситуації, коли користувач вказав свою категорію проблеми використовуючи більше 120 символів рис. 3.4 – Реєстрація проблеми в системі вказуючи свою категорію проблеми більше 120 символів.

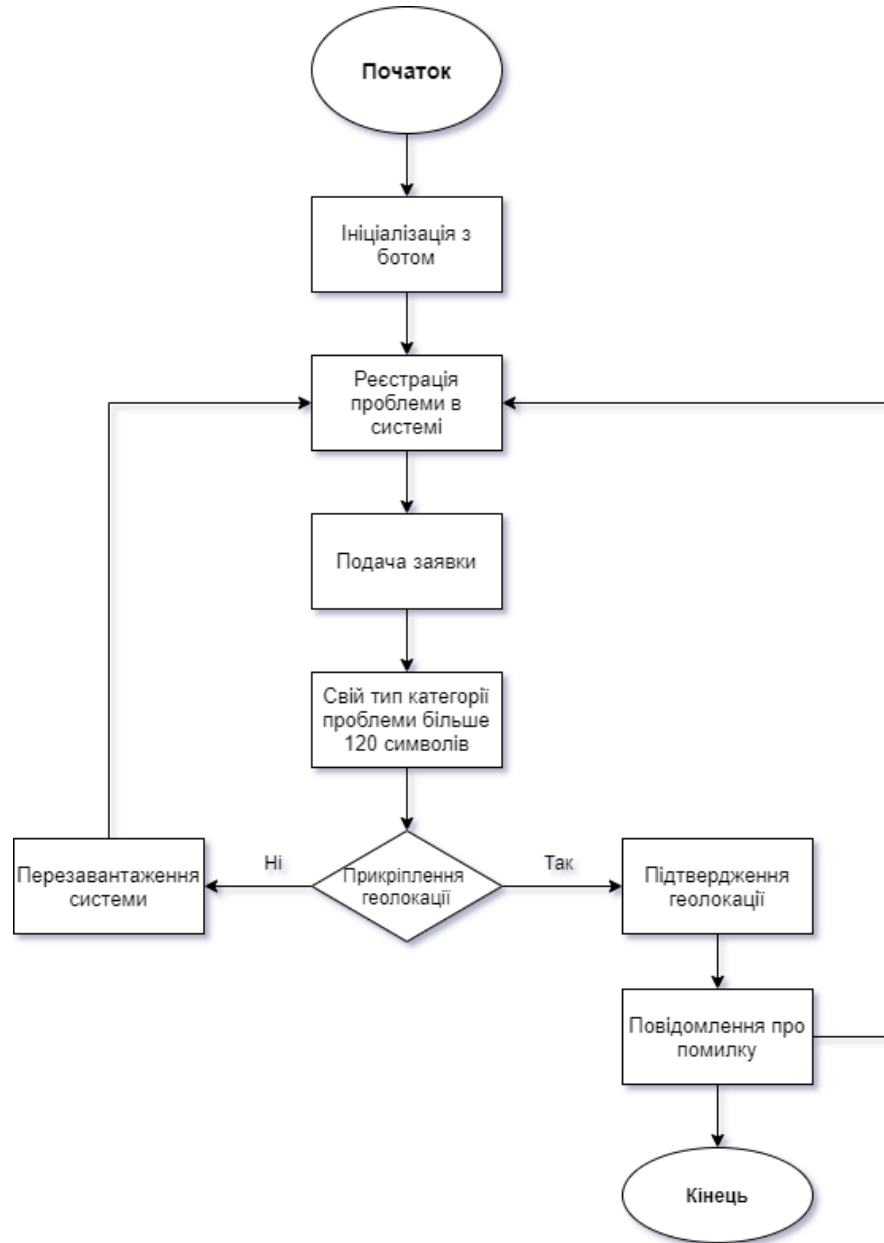


Рисунок 3.4 – Реєстрація проблеми в системі вказуючи свою категорію проблеми більше 120 символів

Наступним кейсом є завантаження даних користувачем, які мають формат відмінний від фотографії чи зображення рис. 3.5 – Завантаження даних в систему, які не підпадають під формат зображень.

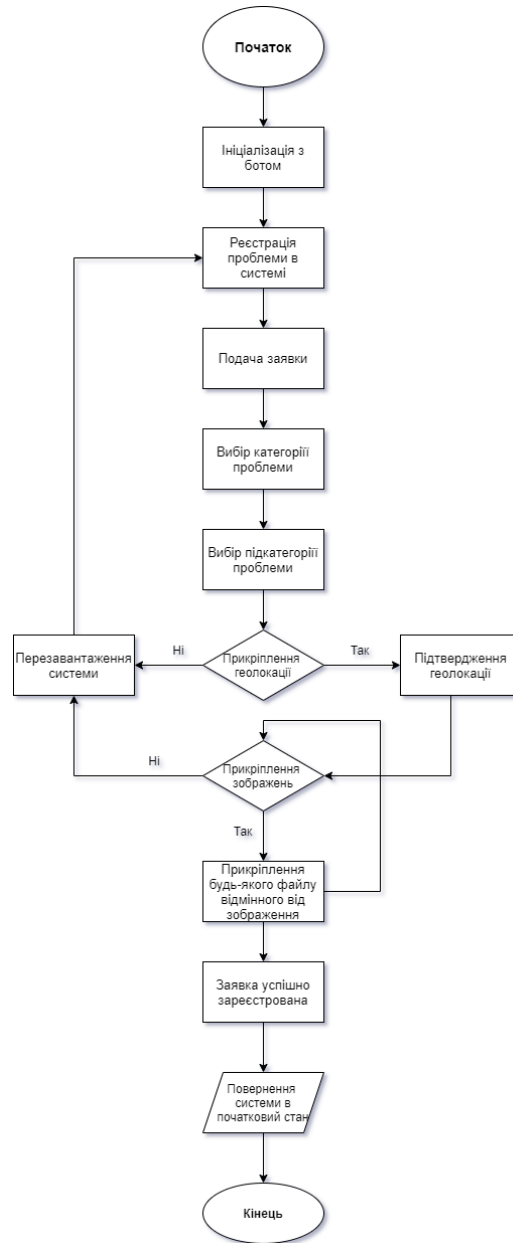


Рисунок 3.5 – Завантаження даних в систему, які не підпадають під формат зображень



В останньому кейсі показано, що система не залишає користувача в безвихідній ситуації, тобто вернутися на стартову точку можна з будь-якого рівня  
 рис. 3.6 – Кнопка «/restart» доступна з будь-якого рівня чат-бота.

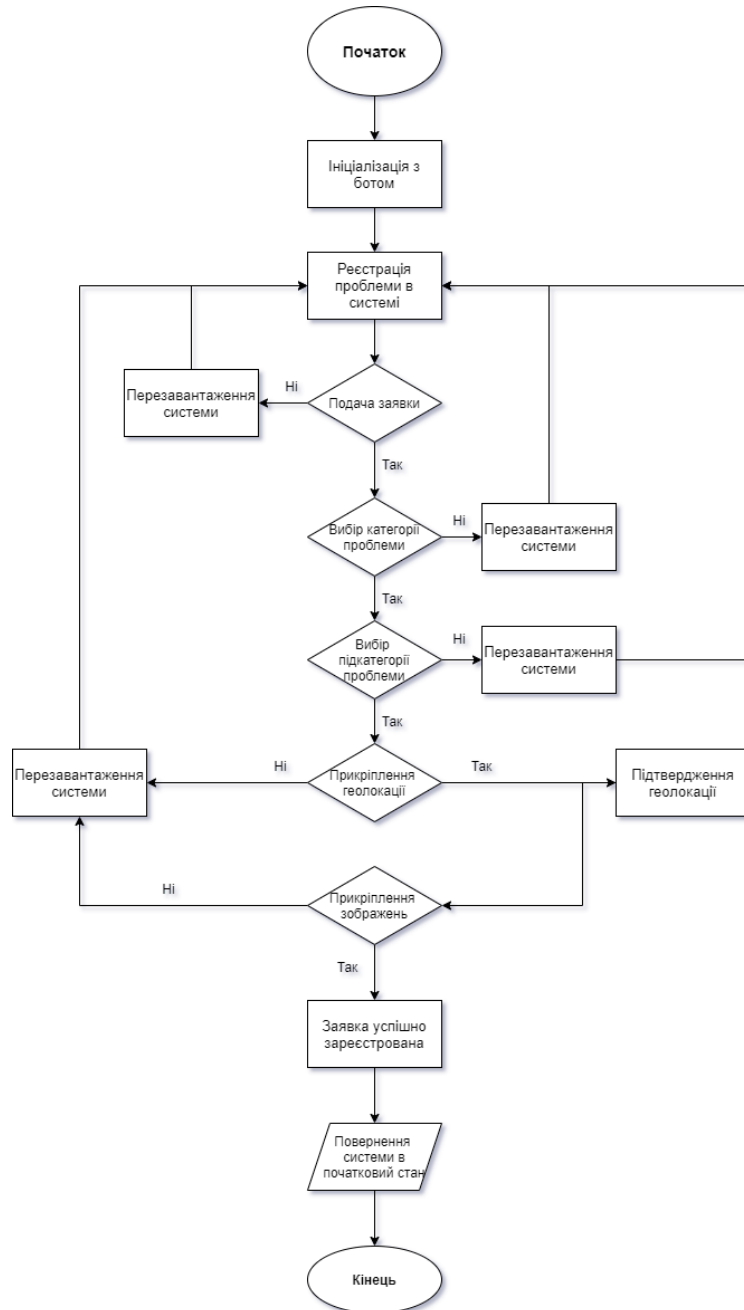


Рисунок 3.6 – Кнопка «/restart» доступна з будь-якого рівня чат-бота

## 4 РОЗРОБКА ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОГО ЧАТ-БОТА

### 4.1 Реєстрація чат-бота.

Щоб створити власного бота, ми повинні зробити запит до основного бота, який називається батько всіх ботів.

Щоб знайти цього бота серед всіх інших, нам потрібно в пошуковій стрічці ввести пошуковий запит @botfather та розпочати з ним діалог (рис. 4.1 - Пошук головного бота).

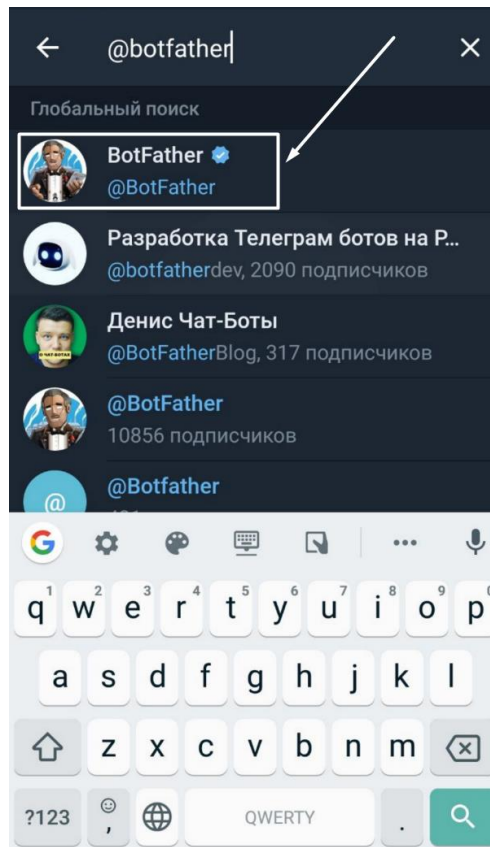


Рисунок 4.1 – Пошук головного бота

Далі клікаємо на іконку бота, після чого у нас з ним відкривається чат, як показано на (рис. 4.2 – Стартовий діалог з головним ботом), та тиснемо на кнопку «СТАРТ»

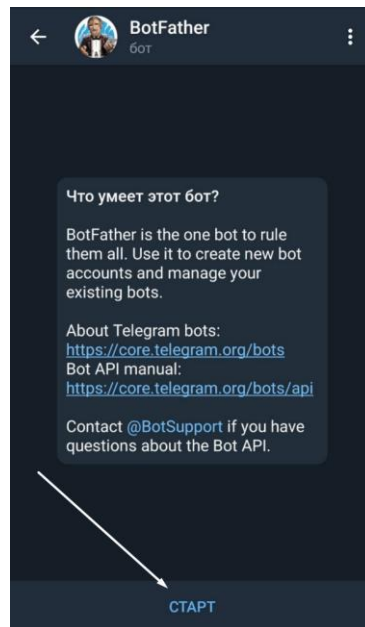


Рисунок 4.2 – Стартовий діалог з головним ботом

Після натискання на кнопку «СТАРТ», бот нам у відповідь надсилає список всіх команд, за допомогою яких ми можемо ним керувати (рис. 4.3 – Список всіх команд бота).

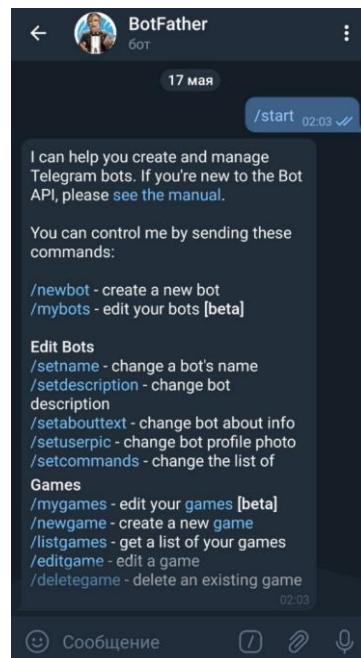


Рисунок 4.3 – Список всіх команд бота

Тут ми не будемо зупинятися на описі кожної з команд, адже весь список нам не знадобиться. Зараз нас цікавить лише створення нового бота. Це команда /newbot. Надалі надсилаємо цю команду для створення нового бота в системі Telegram (рис. 4.4 – Реєстрація нового бота в системі Telegram).

Отримавши це повідомлення, botfather розпочинає процес створення і просить нас встановити ім'я для нового бота, а потім ім'я користувача. Різниця полягає в тому, що ім'я може бути будь-яким, тоді як ім'я користувача обов'язково має закінчуватися на «бот» (щоб людина відразу зрозуміла, що вона спілкується з ботом, а не з кимось іншим). Для простоти ми дамо нашому боту однакове ім'я та ім'я користувача.

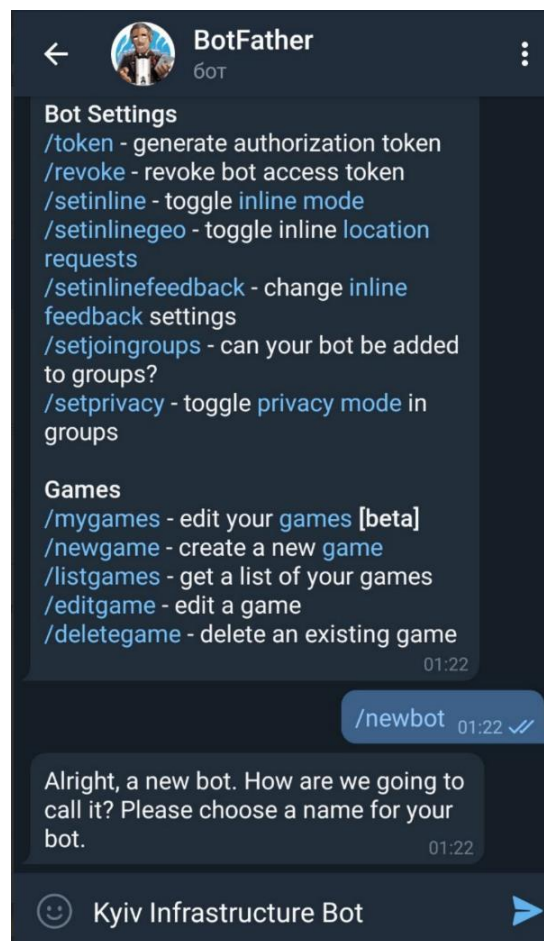


Рисунок 4.4 – Реєстрація нового бота в системі Telegram

Після введення імені новоствореному боту, потрібно присвоїти йому нікнейм, який повинен закінчуватися приставкою bot, для того щоб звичайний користувач міг відрізнити робота від людини (рис. 4.5 – Присвоєння нікнейму чат-боту).

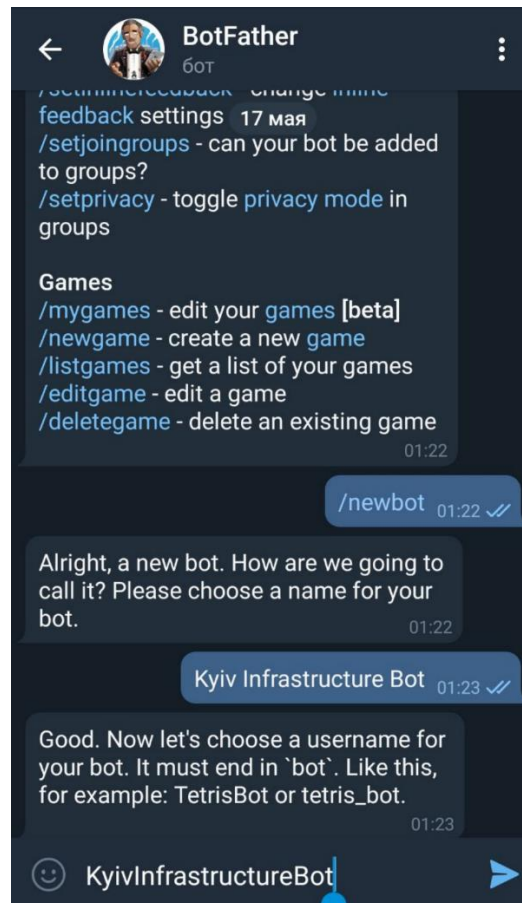


Рисунок 4.5 – Присвоєння нікнейму чат-боту

Після успішного присвоєння нікнейму нам у відповідь приходять повідомлення з секретним токеном (рис. 4.6 – Токен бота). Токен - це унікальний ідентифікатор нашого бота. Точка з'єднання між ботом і кодом на сервері / комп'ютері.

Саме через даний токен буде проходити з'єднання між клієнтом та сервером, тому його потрібно зберігати в таємниці та не передавати його третім особам.

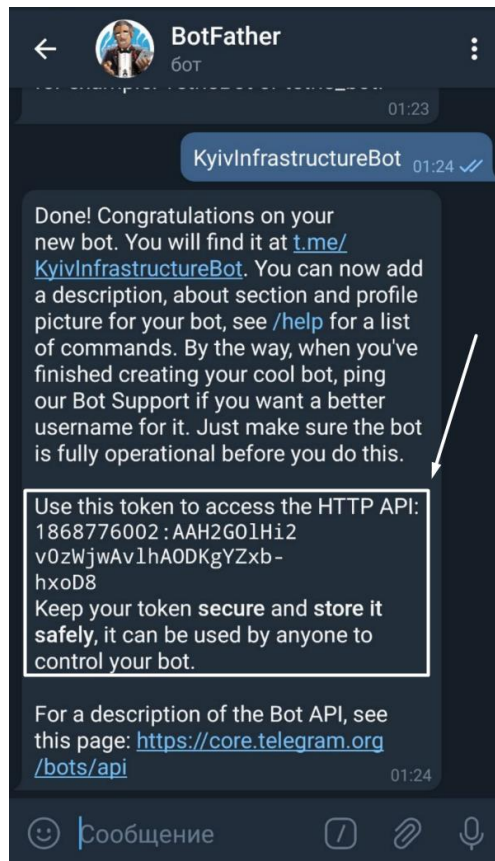


Рисунок 4.6 – Токен бота

Далі ми переходимо до наступної частини дипломної роботи, а саме написання коду (функцій, методів та класів), для обробки вхідних запитів.

## 4.2 Реєстрація в Google Drive Filestorage.

API Google Drive дозволяє створювати програми, які використовують хмарне сховище Google Drive. Ми можемо розробляти додатки, які інтегруються з Google Drive, і створювати надійну функціональність у своєму додатку за допомогою API Google Drive.

На рисунку 4.7 показано взаємозв'язок між нашим додатком Google Drive app, Google Drive і API Google Drive:

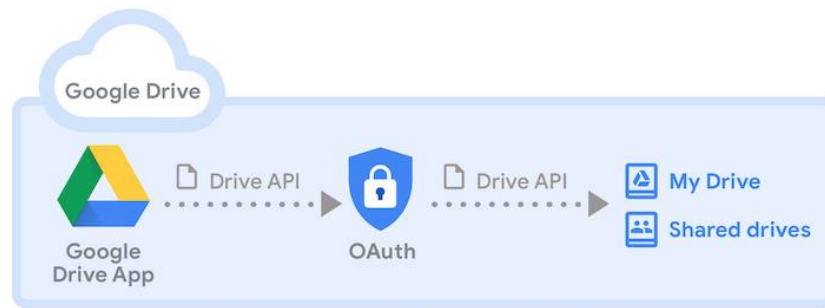


Рисунок 4.7 – взаємозв’язок між нашим додатком Google Drive app, Google Drive і API Google Drive

Схема відображення взаємозв’язків між компонентами:

- Google Drive – служба хмарного зберігання файлів Google, яка надає користувачам особистий простір для зберігання, який називається Мій диск , і можливість доступу до спільних папок, що називаються спільними дисками;
- API Google Drive – REST API, яке дозволяє використовувати пам’ять Google Drive із нашої програми;
- додаток Google Drive – додаток, який використовує Google Drive як рішення для зберігання даних;
- інтерфейс користувача Google Drive – інтерфейс користувача Google, який керує файлами, що зберігаються на Google Drive. Якщо програма є редакторського типу, наприклад, електронною таблицею або текстовим редактором, то можна інтегруватися з інтерфейсом Диска для створення та відкриття файлів у нашій програмі;
- мій диск – місце зберігання на Google Диску, яким володіє певний користувач. Файлами, збереженими на Мій Диск, можна ділитися з іншими користувачами, але право власності на вміст залишається лише для окремого користувача;
- OAuth 2.0 – протокол авторизації, який API Google Drive вимагає для автентифікації користувачів нашого додатку. Якщо у програмі використовується вхід Google , вона обробляє потоки OAuth 2.0 та маркери доступу до програми;

- спільний диск – місце зберігання на Google Диску, яке володіє файлами, якими спільно користуються кілька користувачів. Будь-який користувач, який має доступ до спільного диска, має доступ до всіх файлів, які він містить. Користувачам також може бути наданий доступ до окремих файлів на спільному диску.

API Google Drive можна використовувати для:

- завантаження файлів з Google Диска та завантаження файлів на Google Drive;
- пошуку файлів та папок, що зберігаються на Google Drive;
- створювати складні пошукові запити, які повертають будь-яке з полів метаданих файлів у ресурсі Файли;
- обміну між користувачами файлами, папками та дисками для спільної роботи над вмістом;
- поєднання з API Google Picker для пошуку всіх файлів на Google Диску, а потім повернення ім'я файлу, URL-адресу, дату останньої зміни та користувача;
- створення сторонніх ярликів, які є зовнішніми посиланнями на дані, що зберігаються за межами Диска, в іншому сховищі даних або хмарній системі зберігання;
- створення спеціальних папок на Диску, щоб зберігати дані нашої програми, щоб програма не могла отримати доступ до всього вмісту користувача, що зберігається на Google Диску;
- інтегруванням з інтерфейсом Google Drive , який є стандартним веб-інтерфейсом Google , який ми можете використовувати для взаємодії з файлами Drive.

Щоб зареєструвати обліковий запис для використання Google диска потрібно виконати наступні інструкція, які описані нижче:

- 1) Для початку потрібно увійти в обліковий запис Google Cloud Platform.



- 2) Після успішного входу ми потрапляємо на сторінку панелі приладів, де ми повинні клікнути на посилання з підписом "Create Project", як показано на рис. 4.8 – Створення нового проекту.

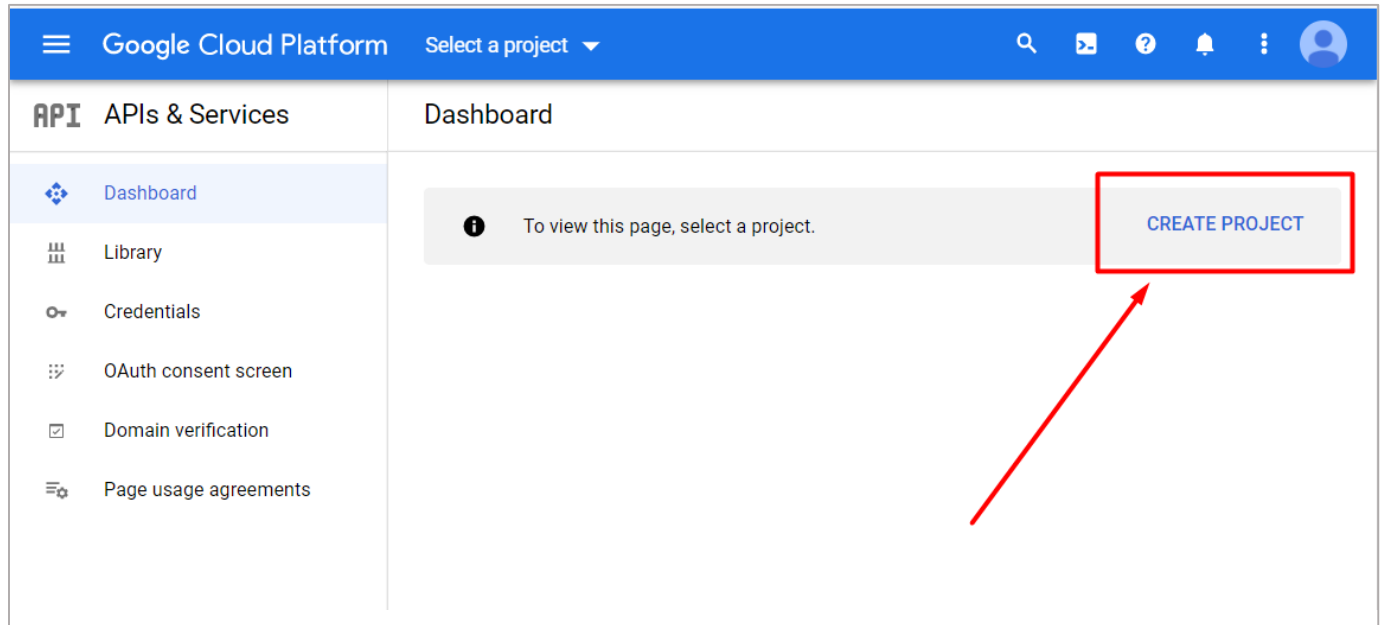


Рисунок 4.8 – Створення нового проекту

- 3) Далі ми повинні назвати наш проект та вибрати йому унікальний ідентифікатор рис 4.9 – Назва проекту та його ідентифікатор . Поле Location ми чіпати не будемо, так як це не комерційний проект. Потім тиснемо кнопку «Create».

Google Cloud Platform

New Project

You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*  
DutEduBot

Project ID \*  
careful-tracer-314222

Project ID can have lowercase letters, digits, or hyphens. It must start with a lowercase letter and end with a letter or number.

Location \*  
No organization [BROWSE](#)

Parent organization or folder

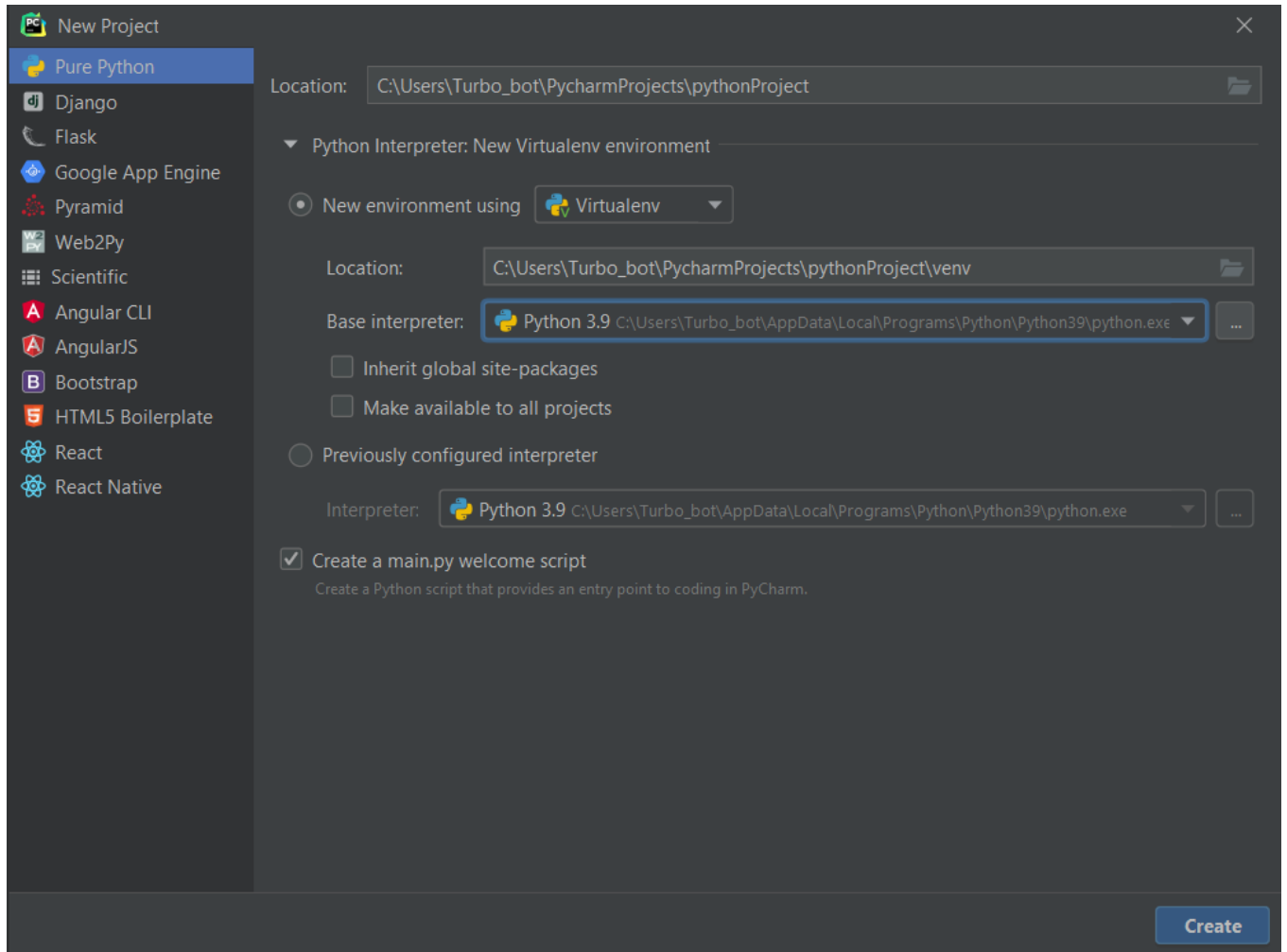
[CREATE](#) [CANCEL](#)

Рисунок 4.9 – Назва проекту та його ідентифікатор

- 4) Наступним кроком нам генерується файл `secret.json`, який ми повинні інтегрувати в наш код.

### 4.3 Розробка основного функціоналу чат-бота.

Для початку створення чат бота нам локально на комп'ютер потрібно інсталиювати Python версії 3.7. Наступним кроком потрібно перейти до інтегрованого середовища розробки додатків, та створити новий проект як показано на рисунку 4.10 – Створення нового проекту.



#### 4.10 – Створення нового проекту

Потім ми повинні назвати наш проект та натиснути на кнопку «Create». Після створення проекту ми повинні перейти в консоль та в якірній дерикторії прописати наступну команду `pip install pyTelegramBotAPI`. Дана команда відповідає за скачування бібліотеки з центрального репозиторія для розробки чат-ботів на мові програмування Python. Якщо все пройшло успішно в терміналі ми маємо побачити повідомлення згідно рис. 4.11 – Успішне встановлення модуля розробки чат-ботів.

```

Downloading certifi-2020.12.5-py2.py3-none-any.whl (147 kB)
|████████████████████████████████████████| 147 kB 6.4 MB/s
Collecting chardet<5,>=3.0.2
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
|████████████████████████████████████████| 178 kB 3.3 MB/s
Using legacy 'setup.py install' for pyTelegramBotAPI, since package 'wheel' is not installed.
Installing collected packages: urllib3, idna, chardet, certifi, requests, pyTelegramBotAPI
  Running setup.py install for pyTelegramBotAPI ... done
Successfully installed certifi-2020.12.5 chardet-4.0.0 idna-2.10 pyTelegramBotAPI-3.7.9 requests-2.25.1 urllib3-1.26.4
(venv) C:\Users\Turbo_bot\PycharmProjects\pythonProject>

```

Рисунок 4.11 – Успішне встановлення модуля розробки чат-ботів

Розроблена база даних з таблицями: problems; users, яка зображена на рис. 4.11 – Таблиці бази даних.

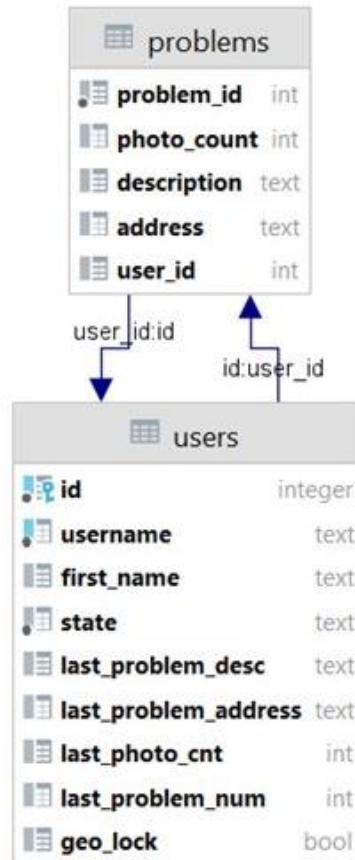


Рисунок 4.12 – Таблиці бази даних

Після проходження всіх вищеописаних кроків настає черга створення структури проекту, тобто створення всіх дерикторій і файлів, які потрібні в процесі розробки інформаційної системи. Структура проекту відображена на рисунку 4.13 – Структура проекту.

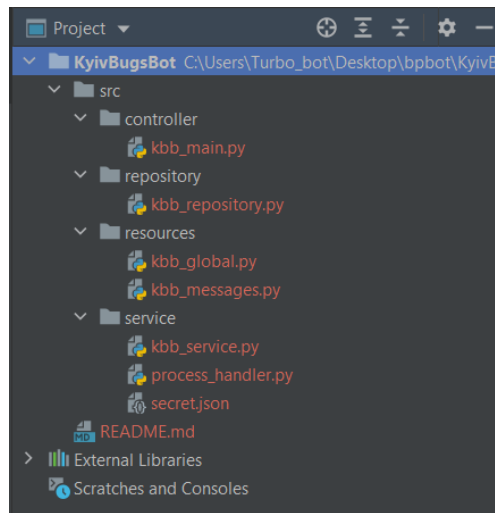


Рисунок 4.13 – Структура проекту

Структура проекту логічно розділена на каталоги: controller, repository, resources, service та також на цьому ж самому рівні знаходиться файл README.md з описом самого чат-бота.

Даний проект побудований по принципу MVC. MVC розшифровується як модель-уявлення-контролер. Це архітектурний стиль програмування, який передбачає розбивку по блокам, що відповідають за різні функції програмного додатку. Один з блоків відповідає за дані, інший відповідає за зовнішній вигляд, а третій контролює роботу програми. Компоненти MVC:

- модель – це компонент який відповідає за дані, а також визначає структуру додатка. Найчастіше відповідає за зв'язок з базою даних, в ньому описується вибірка даних;

- уявлення – даний компонент відповідає за взаємодію з користувачем. Тобто зовнішній вигляд програми. У нашому випадку в ролі уявлення буде виступати клієнтське програмне забезпечення Telegram;
- контролер - цей компонент відповідає за зв'язок між моделлю та уявленням. Компонент контролер відповідає за функціонування програми. Він обробляє вхідні запити, робить вибірку з бази даних та віддає це все на компонент уявлення.

Опис файлів та каталогів ієрархічної структури проекту:

- каталог controller містить файл `kbb_main.py`, який відповідає за логіку програми, тобто за те, як вона буде обробляти вхідні запити і які дані буде віддавати користувачу;
- в каталозі repository присутній файл `kbb_repository.py`, який відповідає за зв'язок з базою даних. В даному файлі описані всі запити та вибірки з бази даних;
- в каталозі resources файл `kbb_global.py` відповідає за глобальні константи та конфігурації бота. Там описані токени, шляхи каталогів та масиви статичних даних кнопок, які присутні в чат боті. Також в цьому ж каталозі присутній ще один файл під назвою `kbb_messages.py` в якому описаний весь статичний текст який присутній в системі;
- в наступному каталозі під назвою service знаходиться файл `kbb_service.py`, який відповідає за функціональність з'єднання та завантаження графічних даних на віддалене сховище Google Drive. Наступний файл, який присутній в каталозі це `process_handler.py`, який відповідає за чергу повідомлень. Тобто даний скрипт відповідає за блокування потоків виконання програми і створює чергу повідомлень, для того, щоб не виникало конфліктів між модулями. Ще один файл, який присутній в даному каталозі це `secret.json`, автоматично

згенерований системою Google Drive, в якому містяться авторизаційні дані від сервіс аккаунта.

Точкою запуску чат-бота є файл `kbb_main.py` в якому присутня найперша функція `start_handler(message)`, яка спрацьовує при ініціалізації користувача з ботом. Функція приймає параметр `message`, який має об'єкт `message.chat.id` за допомогою якого проходить ініціалізація з користувачем. Функція представлена на рисунку 4.14 – Функція ініціалізації з ботом.

```
@bot.message_handler(commands=['start', 'help', 'hello'])
def start_handler(message):
    user_id = message.chat.id
    logging.info("Called /start by %s %s", user_id, message.chat.username)

    validate_user_existence(message.chat)
    try:
        process_start(user_id)
    except: # dat's shit, but I really don't know which exception we can get here
        handle_exception(user_id)
```

Рисунок 4.14 – Функція ініціалізації з ботом

Щоб подати заявку, потрібно викликати функцію `application_handler (message)` рис. 4.15 – Функція з вибором категорії проблеми , яка надсилає повідомлення користувачу з списком категорій проблем.

```

@bot.message_handler(commands=['application'])
def application_handler(message):
    validate_user_existence(message.chat)

    user_id = message.chat.id
    state = get_state(message.chat.id)

    logging.info("")
    logging.info("Called /application by %s %s %s. State is: %s", user_id,
                 message.chat.username, message.chat.first_name, state)

    if state != "start" and state != "description":
        bot.send_message(user_id, restart_default)
        return

    try:
        process_application(user_id)
    except: # dat's shit, but I really don't know which exception we can got here
        handle_exception(user_id)

```

Рисунок 4.15 – Функція з вибором категорії проблеми

Наступним кроком потрібно прикріпити геолокацію проблеми написавши адресу боту в повідомленні, чи прикріпити координати за допомогою вбудованих функцій в месенджері Telegram.

За прикріплення геолокації до проблеми відповідає функція `location_handler(message)`, реалізація якої показана на рисунку 4.16 – Реалізація функції прикріплення геолокації.



```

@bot.message_handler(content_types=['location'])
def location_handler(message):
    validate_user_existence(message.chat)

    user_id = message.chat.id
    state = get_state(user_id)

    logging.info("")
    logging.info("Start process location by: " + str(user_id))

    if state != "address":
        bot.send_message(user_id, restart_default)
        return

    try:
        process_location(message)
    except: # dat's shit, but I really don't know which exception we can got here
        handle_exception(user_id)

```

Рисунок 4.16 – Реалізація функції прикріплення геолокації

Далі ми повинні прикріпити фото до нашої заявки. За прикріплення фото відповідає функція реалізація якої зображена на рисунку 4.17 – Функція прикріплення фото.

```

@bot.message_handler(content_types=['photo'])
def get_photo_handler(message):
    validate_user_existence(message.chat)

    user_id = message.chat.id
    state = get_state(user_id)

    logging.info("")
    logging.info("Got photo from user. State is: %s", state)

    if state != "photo":
        bot.send_message(user_id, restart_photo)
        return

    try:
        process_photo(message)
    except: # dat's shit, but I really don't know which exception we can got here
        handle_exception(user_id)

```

Рисунок 4.17 – Функція прикріплення фото

Після всіх вищеописаних кроків система реєструє проблему в базі. Тому далі ми можемо з даних, які у нас є написати заявку в КМДА для вирішення інфраструктурної проблеми.

#### 4.4 Тестування чат-бота.

Для того, щоб протестувати чат-бота для початку потрібно створити тестову документацію таку як тест кейси. Даний тип тестової документації потрібний для опису функціоналу програми. А також для описання кроків досягнення функціональних цілей. В таблиці 4.1 – Тест кейси, описані функціональні тест кейси.

Таблиця 4.1 – Тест кейси реєстрації проблеми в системі

| Функція   | Кроки   | Очікуваний результат   |
|---|---|--|
| Реєстрація проблеми в системі.<br>Зображена на рисунку 4.18 – Реєстрація проблеми в системі чат-бота. | <ol style="list-style-type: none"> <li>1) Перейти до чат бота.</li> <li>2) Натиснути на кнопку «/application».</li> <li>3) Далі вибрати проблему зі списку.</li> <li>4) Потім вибрати саб проблему зі списку.</li> <li>5) Наступним кроком поділитися геолокацією проблеми.</li> <li>6) Далі завантажити фото проблеми та натиснути на кнопку «/upload».</li> </ol> | <ol style="list-style-type: none"> <li>1) Відкривається чат з ботом.</li> <li>2) Система просить вибрати категорію проблеми.</li> <li>3) Система просить вибрати під категорію проблеми.</li> <li>4) Система просить поділитися геолокацією.</li> <li>5) Система просить завантажити фото.</li> <li>6) Система повідомляє користувача про успішно зареєстрований запит.</li> </ol> |

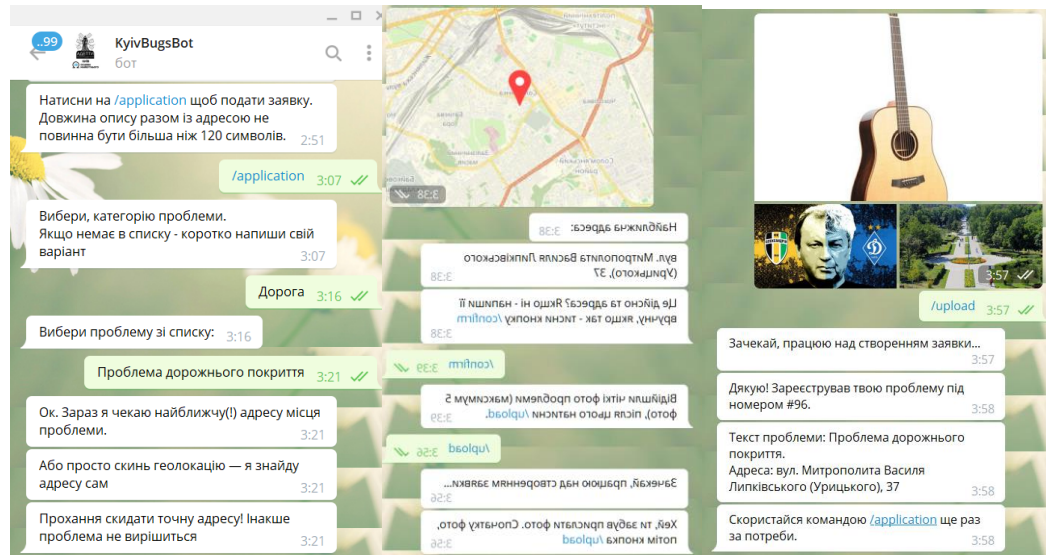


Рисунок 4.18 – Реєстрація проблеми в системі чат-бота

Таблиця 4.2 – Тест кейси реєстрації проблеми в системі вказуючи свою категорію проблеми

|  |  |  |
|--|--|--|
| <p>Реєстрація проблеми в системі вказуючи свою категорію проблеми. Зображена на рисунку 4.19 – Реєстрації проблеми в системі чат-бота вказуючи категорію проблеми.</p> | <ol style="list-style-type: none"> <li>1) Перейти до чат бота.</li> <li>2) Натиснути на кнопку «/application».</li> <li>3) Далі описати свою проблему до 120 символів та відправити.</li> <li>4) Наступним кроком поділитися геолокацією проблеми.</li> <li>5) Далі завантажити фото проблеми та натиснути на кнопку «/upload».</li> </ol> | <ol style="list-style-type: none"> <li>1) Відкривається чат з ботом.</li> <li>2) Система очікує описання проблеми.</li> <li>3) Система просить поділитися геолокацією.</li> <li>4) Система просить завантажити фото.</li> <li>5) Система повідомляє користувача про успішно зареєстрований запит.</li> </ol> |
|--|--|--|

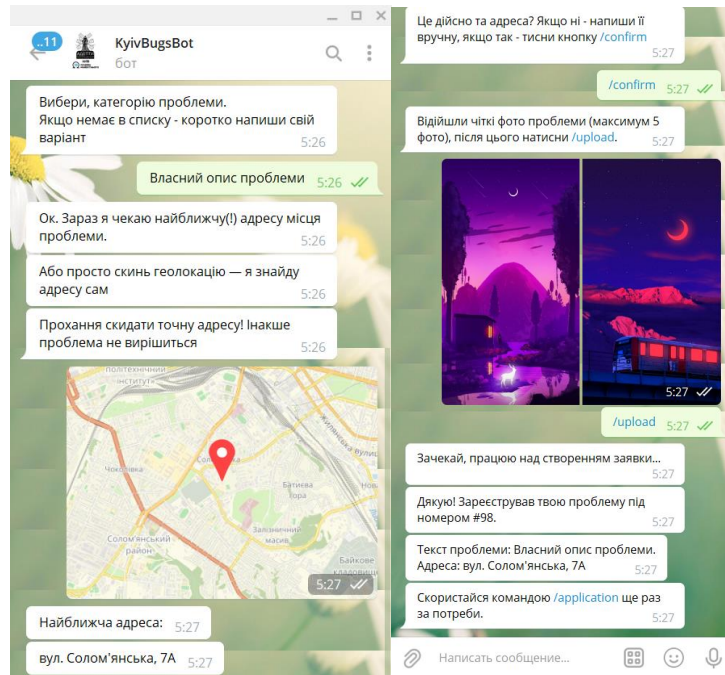


Рисунок 4.19 – Реєстрації проблеми в системі чат-бота вказуючи категорію проблеми

Таблиця 4.3 – Тест кейси реєстрації проблеми в системі вказуючи свою категорію проблеми більше 120 символів

|   |   |   |
|---|---|---|
| <p>Реєстрація проблеми в системі, вказуючи свою категорію проблеми більше 120 символів. Зображена на рисунку 4.20 – Реєстрації проблеми в системі чат-бота з використанням більше 120 символів.</p> | <ol style="list-style-type: none"> <li>1) Перейти до чат бота.</li> <li>2) Натиснути на кнопку «/application».</li> <li>3) Далі описати свою проблему більше 120 символів та відправити.</li> <li>4) Наступним кроком поділитися геолокацією проблеми.</li> </ol> | <ol style="list-style-type: none"> <li>1) Відкривається чат з ботом.</li> <li>2) Система очікує описання проблеми.</li> <li>3) Система просить поділитися геолокацією.</li> <li>4) Система видасть попередження, що опис проблеми та геолокація більше 120 символів.</li> </ol> |
|---|---|---|

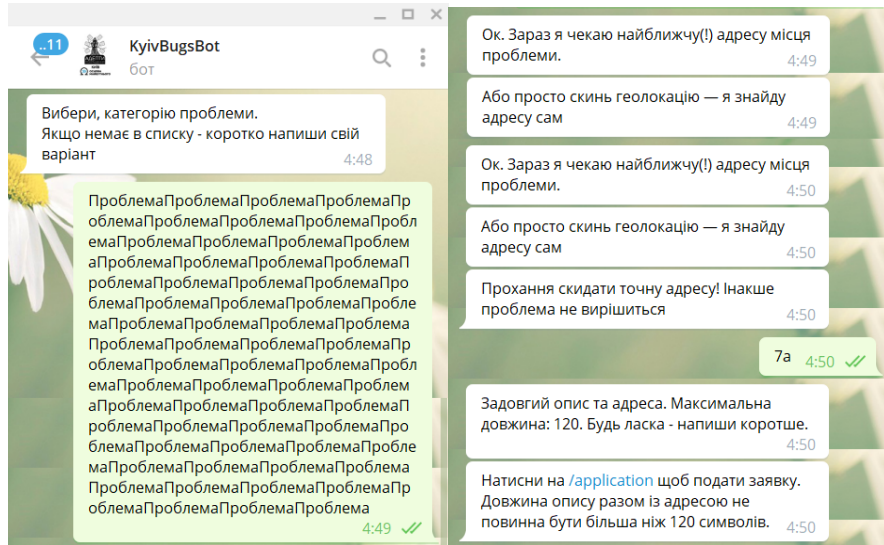


Рисунок 4.20 – Реєстрації проблеми в системі чат-бота з використанням більше 120 СИМВОЛІВ

Таблиця 4.4 – Тест кейси завантаження даних в систему, які не підпадають під формат фотографій/картинок

|   |   |   |
|---|---|---|
| <p>Завантаження даних в систему, які не підпадають під формат фотографій/картинок. Зображена на рисунку 4.21 – Реєстрації завантаження даних в систему.</p> | <ol style="list-style-type: none"> <li>1) Перейти до чат бота.</li> <li>2) Натиснути на кнопку «/application».</li> <li>3) Далі вибрати проблему зі списку.</li> <li>4) Потім вибрати саб проблему зі списку.</li> <li>5) Наступним кроком поділитися геолокацією проблеми.</li> <li>6) Далі завантажити будь-який формат даних відмінний від фото та натиснути на кнопку «/upload».</li> </ol> | <ol style="list-style-type: none"> <li>1) Відкривається чат з ботом.</li> <li>2) Система просить вибрати категорію проблеми.</li> <li>3) Система просить вибрати під категорію проблеми.</li> <li>4) Система просить поділитися геолокацією.</li> <li>5) Система просить завантажити фото.</li> <li>6) Система повідомляє користувача про те що вона працює.</li> </ol> |
|---|---|---|

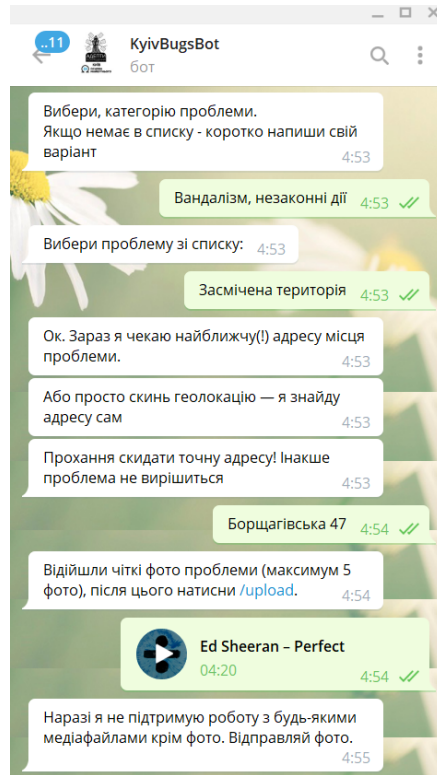


Рисунок 4.21 – Реєстрації завантаження даних в систему

Таблиця 4.5 – Тест кейси кнопки «/restart»

|  |   |   |
|--|---|---|
| <p>Кнопка «/restart», доступна з будь-якого рівня чат-бота. Зображена на рисунку 4.22 – Кнопка «/restart» в системі.</p> | <ol style="list-style-type: none"> <li>1) Перейти до чат бота.</li> <li>2) Натиснути на кнопку «/application».</li> <li>3) Далі вибрати проблему зі списку.</li> <li>4) Потім вибрати саб проблему зі списку.</li> <li>5) Наступним кроком поділитися геолокацією проблеми.</li> <li>6) Далі натиснути на кнопку «/restart».</li> </ol> | <ol style="list-style-type: none"> <li>1) Відкривається чат з ботом.</li> <li>2) Система просить вибрати категорію проблеми.</li> <li>3) Система просить вибрати під категорію проблеми.</li> <li>4) Система просить поділитися геолокацією.</li> <li>5) Система просить завантажити фото.</li> </ol> |
|--|---|---|

## Продовження таблиці 4.5 – Тест кейси кнопки «/restart»

|   |  |   |
|---|--|---|
| <p>Кнопка «/restart» доступна з будь-якого рівня чат-бота.<br/>Зображена на рисунку 4.22 – Кнопка «/restart» в системі.</p> |  | <p>б) Система повинна повернути користувача в початковий стан створення заявки.</p> |
|---|--|---|

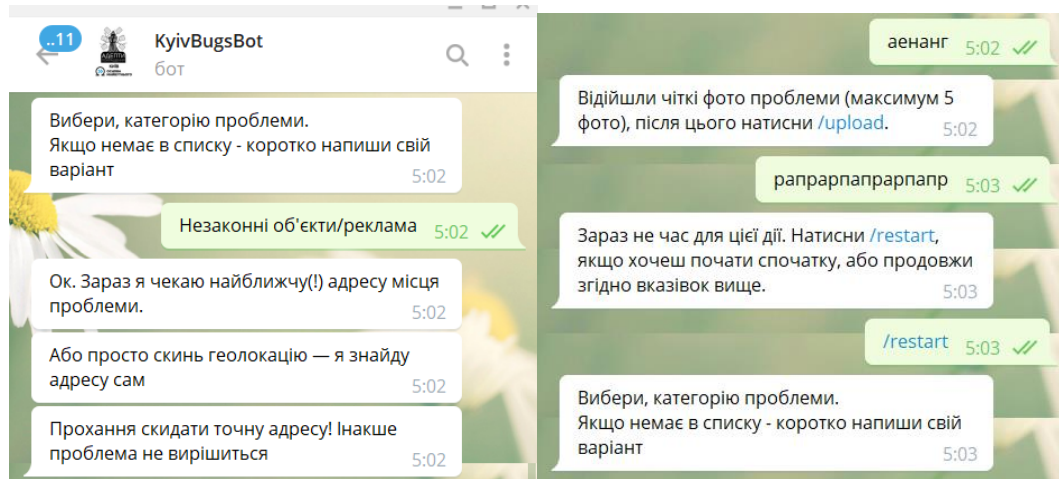


Рисунок 4.22 – Кнопка «/restart» в системі

## ВИСНОВОК

В ході виконання дипломної роботи було проведено дослідження предметної області, визначено головні вимоги до інформаційної системи. Також було проведено дослідження реалізації інформаційної системи.

Результатом дослідження технологій був підібраний стек, який на даний час є найактуальнішим та найзрозумілішим. В роботі були використані наступні технології:

- мова програмування – Python 3.7;
- модулі – TeleBot, pathlib, os, google.oauth2, googleapiclient.discovery, googleapiclient.http, requests;
- база даних – SQL lite;
- хмарне сховище зображень – Google Drive.

Результатом дослідження стало створення сучасного чат-бота на платформі Telegram, для моніторингу заявок КМДА. Система розроблена згідно найсучасніших патернів проектування, має зрозумілий інтерфейс та не потребує додаткового навчання користування системою.

Найбільшою перевагою чат-бота є те, що він кросплатформений, тобто його можна використовувати з будь-де, де є можливість комунікації через протокол передачі даних HTTP. Також бот може бути запущений на різних пристроях включаючи смартфони, комп'ютери, планшети та інші пристрої, які можуть бути підключені до глобальної павутини інтернет.



## ПЕРЕЛІК ПОСИЛАНЬ

1. Чат-бот [Електронний ресурс]. - Режим доступу : [http:// inlnk.ru/WypLP](http://inlnk.ru/WypLP).
2. Токарева Ю.А., Розробка чат-ботів і розмовних інтерфейсів. – Перев. с англ. – Київ: ДМК Пресс, 2018 р. – 340 стор.
3. Python для інженерів [Електронний ресурс]. - Режим доступу : <https://inlnk.ru/bk1wO>.
4. Telegram Bot API [Електронний ресурс]: Telegram Documents. - Режим доступу: <https://core.telegram.org/bots/api>.
5. Кравченко Н.В, Telegram, як приклад месенджера: можливості та перспективи розвитку. [Електронний ресурс] / Н. Л. Аванесян // Науковий потенціал ХХІ століття. - 2017. - Режим доступу: [https://elibrary.ru/download/\\_4017.pdf](https://elibrary.ru/download/_4017.pdf).
6. Офіційний сайт Telegtam [Електронний ресурс]: API - Режим доступу: <https://core.telegram.org/api>.
7. Statistical Data and Programming Languages Analysis for Developing a Chatbot in Telegram – 2020 - [Електронний ресурс] / Victoria O. Rytikova, Maxim A. Medvedev – Режим доступу: <https://aip.scitation.org/doi/epdf/10.1063/5.0041762>.
8. Most popular global mobile messenger apps [Електронний ресурс] – Режим доступу: <https://www.statista.com/statistics/258749/most-popular-globalmobile-messenger-apps>.
9. Statistical Data and Programming Languages Analysis for Developing a Chatbot in Telegram – 2020 - [Електронний ресурс] / Victoria O. Rytikova, Maxim A. Medvedev – Режим доступу: <https://aip.scitation.org/doi/epdf/10.1063/5.004176>.
10. Аристова, А. С., Безносюк, Ю. С., Ведикер, П. К. & Воронович, Н. Е. Використання чат-ботів в навчальному процесі. The 2th International Conference on Digitalization 2019 [Електронний ресурс] – Режим доступу: [https://elar.urfu.ru/bitstream/10995/82473/1/978-80-88327-04-2\\_017.pdf](https://elar.urfu.ru/bitstream/10995/82473/1/978-80-88327-04-2_017.pdf).

- 11.Virtual interlocutor 2021 [Електронний ресурс] – Режим доступу:  
[https://www.chatbot.com/?a=IwK09GOWg&gclid=CjwKCAjw-qeFBhAsEiwA2G7NlyqZooRFMGohqRy6Gh1VvZK2bV15AygRHHpYb3IfjPhOvHg0II\\_3vRoC-rsQAvD\\_BwE](https://www.chatbot.com/?a=IwK09GOWg&gclid=CjwKCAjw-qeFBhAsEiwA2G7NlyqZooRFMGohqRy6Gh1VvZK2bV15AygRHHpYb3IfjPhOvHg0II_3vRoC-rsQAvD_BwE).
- 12.Chat Widget for ChatBot [Електронний ресурс] – Режим доступу:  
<https://www.chatbot.com/integrations/chat-widget>.
- 13.Messenger and everything about it [Електронний ресурс] –  
<https://aboutmessengers.ru/messendzher-hto-eto-takoe>.
- 14.pyTelegramBotAPI [Електронний ресурс] – <https://github.com/eternnoir>.
- 15.Основні переваги фреймворків [Електронний ресурс] – [https://web-creator.ru/articles/about\\_frameworks](https://web-creator.ru/articles/about_frameworks).

## ДОДАТОК А

1



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра інженерії програмного забезпечення



ТЕМА БАКАЛАВРСЬКОЇ РОБОТИ:

### «РОЗРОБКА ТЕЛЕГРАМ-БОТА ДЛЯ МОНІТОРИНГУ ЗАЯВОК КМДА МОВОЮ PYTHON»

Доповідач: студент групи ПД-44  
Рибчук Назарій Володимирович

Науковий Керівник: доцент каф. ІІЗ,  
к.т.н., доц. Жебка В.В.

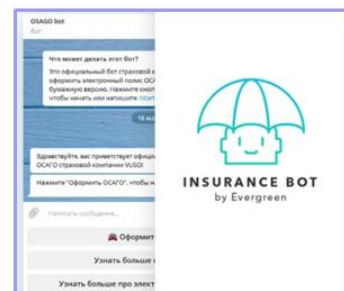
КИЇВ 2021 Р.

2

## АНАЛОГИ

Insurance Bot – чат-бот для страхової компанії.

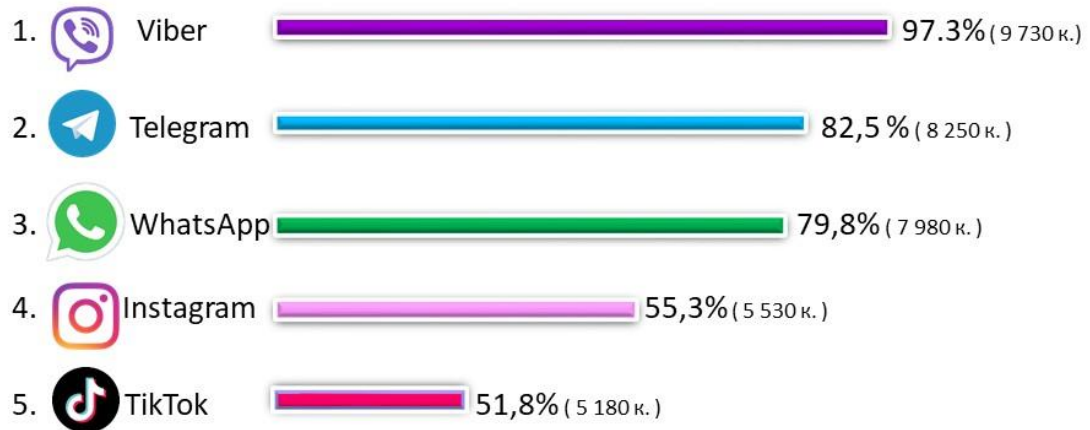
|          |  |
|----------|--|
| Переваги | <ul style="list-style-type: none"> <li>- зручність в використанні;</li> <li>- технічна підтримка.</li> </ul> |
| Недоліки | <ul style="list-style-type: none"> <li>- повільність;</li> <li>- платне рішення.</li> </ul>                  |



## ПОПУЛЯРНІСТЬ МЕСЕНДЖЕРІВ В УКРАЇНІ

3

Рейтинг мобільних додатків (месенджерів) в березні 2021р.






Опитування проходили 10 000 користувачів.

Умовні позначення: к. - користувачі

## ПЕРЕВАГИ МЕСЕНДЖЕРІВ

4

|   |   |
|---|---|
|  | <ul style="list-style-type: none"> <li>✓ підтримка операційних систем Android, iOS, Windows Phone;</li> <li>✓ здійснення аудіо та відеодзвінків;</li> <li>✓ пересилання файлів і зображень;</li> <li>✓ необмежений обмін повідомленнями;</li> <li>✓ видалення надісланих повідомлень у себе і співрозмовника;</li> <li>✓ створення публічних каналів і чатів.</li> </ul>  |
|  | <ul style="list-style-type: none"> <li>✓ підтримка операційних систем Android, iOS, Windows Phone;</li> <li>✓ видалення відправлених повідомлень у себе і співрозмовника;</li> <li>✓ здійснення аудіозвоноків, але відсутність відеодзвінків;</li> <li>✓ створення ботів і публічних каналів на Telegram з подальшою їх монетизацією;</li> <li>✓ пересилання файлів і зображень;</li> <li>✓ необмежений обмін текстовими повідомленнями;</li> </ul> |
|  | <ul style="list-style-type: none"> <li>✓ підтримка операційних систем Android, iOS, Windows Phone;</li> <li>✓ створення групових чатів для спілкування відразу з декількома друзями;</li> <li>✓ здійснення аудіо та відеодзвінків;</li> <li>✓ пересилання файлів і зображень;</li> <li>✓ необмежений обмін повідомленнями;</li> </ul>   |

## ОБ'ЄКТ, ПРЕДМЕТ, МЕТА ДОСЛІДЖЕННЯ

- *Об'єкт дослідження* –моніторинг заявок користувачів за допомогою месенджера.
- *Предмет дослідження* – чат-бот для моніторингу заявок.
- *Мета роботи* – розробка функціонального чат-боту на платформі Telegram, що забезпечить ефективний моніторинг заявок КМДА.

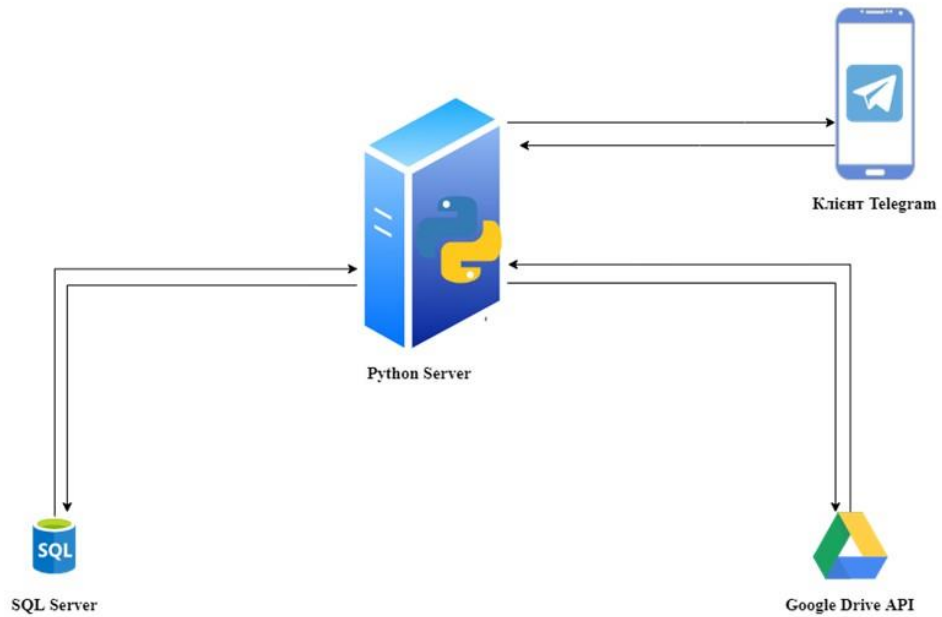
## ТЕХНІЧНІ ЗАВДАННЯ

- 1) Дослідити предметну область.
- 2) Дослідити реалізацію інформаційної системи.
- 3) Проаналізувати існуючі месенджери.
- 4) Проаналізувати найкращі практики та технології розробки чат-ботів.
- 5) Спроекувати архітектуру чат-бота.
- 6) Розробити та протестувати систему чат-бота.

# ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

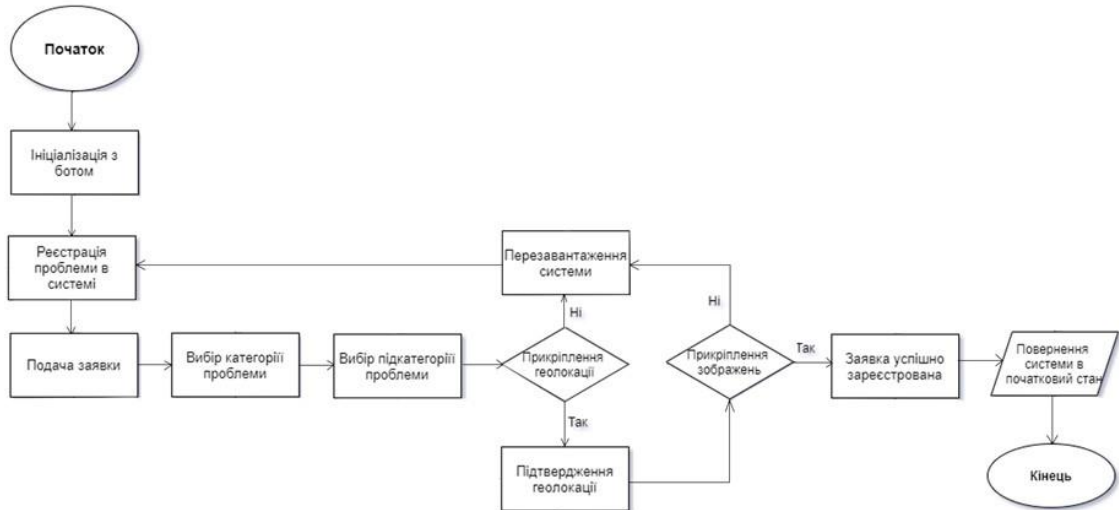


# АРХІТЕКТУРА ІНФОРМАЦІЙНОГО ЧАТ-БОТА



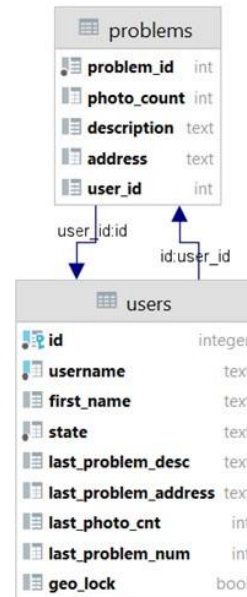
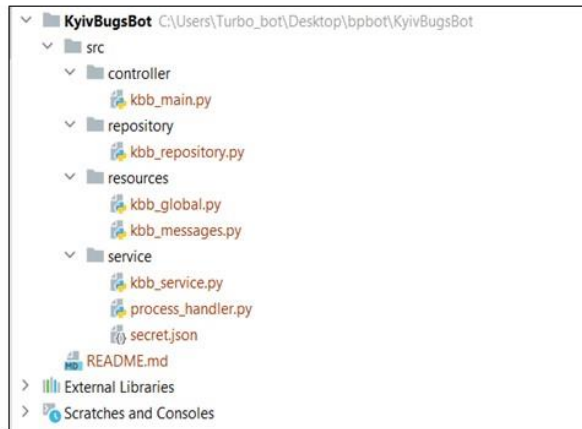
9

## Алгоритм роботи чат-бота

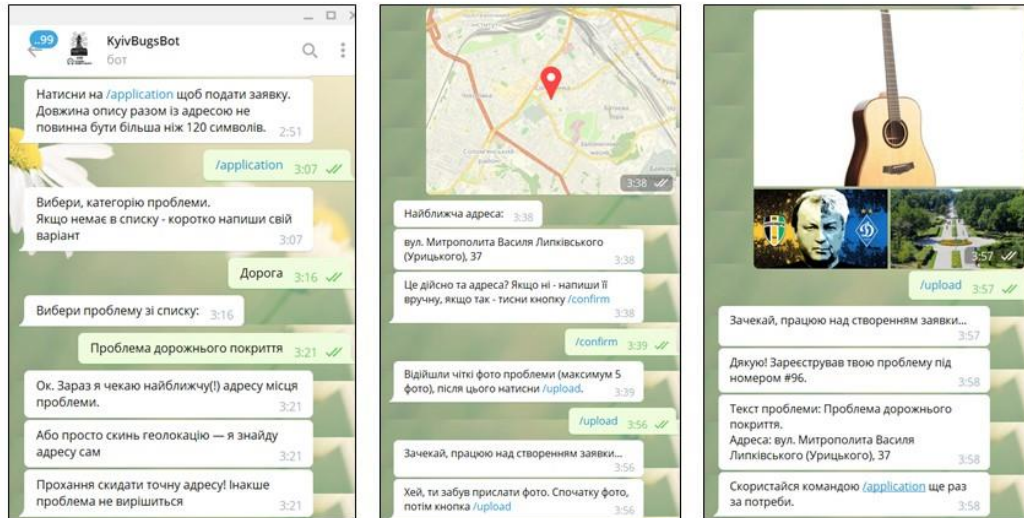


10

## СТРУКТУРА ПРОЕКТУ ТА БАЗА ДАНИХ



## Тестування системи



## ВИСНОВОК

1. Досліджено предметну область.
2. Досліджено реалізацію інформаційної системи.
3. Проаналізовано існуючі месенджери.
4. Проаналізовані найкращі практики та технології розробки чат-ботів.
5. Спроектовано архітектуру чат-бота.
6. Розроблено та протестовано систему чат-бота.



## АПРОБАЦІЇ

1) Рибчук Н.В., Ставицька Ю.В. Чат-боти у сучасному IT-просторі / XII Науково-технічна конференція «Сучасні Інфокомунікаційні Технології. Збірник тез 20.05.2021, ДУТ, м. Київ – К.: ДУТ, 2021. С. 87.

2) Рибчук Н.В. Автоматизація та управління бізнес процесами за допомогою чат-ботів / Всеукраїнська Науково-технічна конференція застосування програмного забезпечення в інфокомунікаційних технологіях. Збірник тез 12.02.2021, ДУТ, м. Київ – К.: ДУТ, 2021. С. 82.

**Дякую за увагу!**