

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

Кафедра системного аналізу

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр

на тему: **«РОЗРОБКА TELEGRAM-ЧАТУ ДЛЯ МОНІТОРИНГУ
ЗАЛІЗНИЧНИХ КВИТКІВ»**

Виконав: студент 6 курсу, групи САДМ-61
спеціальності

124 Системний аналіз

(шифр і назва спеціальності)

Запаріна Е.О.

(прізвище та ініціали)

Керівник Гордієнко Т.Б.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ – 2022

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

Спеціальність 124, Системний аналіз

Гордієнко Т.Б.
“ ” 2022 року

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка telegram-чату для моніторингу залізничних квитків

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від“ ” 2022 року №

2. Строк подання студентом роботи

3. Вихідні дані до роботи

Матеріали за технологіями, рекомендації керівник, технічна література

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

інформаційної системи, висновок, список використаних джерел, додатки

5. Перелік графічного матеріалу

Презентація, виконана в Microsoft Office PowerPoint

6. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		Виконав
2	Пошук та аналіз обраної предметної галузі		Виконав
3	Порівняльний аналіз наявних чат-ботів, визначення аналогів, формування ТЗ		Виконав
5	Вибір технологій та середовища розробки		Виконав
6	Спроекувати чат-бот, розробка чат-бота на платформі Telegram, провести тестування розробленого чат-бота		Виконав
7	Розробка обов'язкових демонстраційних матеріалів		Виконав
8	Попередній захист роботи		Виконав
9	Пред'явлення роботи в деканат		Виконав

Студент

(підпис)

Запаріна Е.О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гордієнко Т.Б.

(прізвище та ініціали)

РЕФЕРАТ

Загальний обсяг роботи становить 60 с., 19 рис., 1 дод., 38 джерел.

Дипломний проект на тему «Розробка telegram-чату для моніторингу залізничних квитків» складається зі вступу, основної частини із трьох розділів, висновків, переліку джерел посилання, додатків.

Мета дослідження – аналіз особливостей розробки та практичної реалізації автоматизованої системи моніторингу залізничних квитків.

Об’єкт дослідження – процес проектування автоматизованої системи моніторингу залізничних квитків на прикладі мобільного додатку telegram-боту.

Предмет дослідження – особливості застосування засобів системного програмування для реалізації клієнт-сервісних систем.

Наукова новизна одержаних результатів: В результаті досліджень був запропонований аналогічний метод центрадізованого пошуку та автоматизованого моніторингу квитків з використанням засобів системного програмування протягом розробки автоматичної системи моніторингу квитків на потяг на основі telegram-боту.

Практичне значення одержаних результатів розробка бота-помічника для системи миттєвого обміну повідомленнями (месенджера) в Telegram, що пришвидшує вирішення простих щоденних завдань, а також мінімізує затрати для співробітників.

Ключові слова: АВТОМАТИЗОВАНА СИСТЕМА, БАЗА ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ЧАТ-БОТ, ШТУЧНИЙ ІНТЕЛЕКТ, TELEGRAM.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АС - автоматизована система;

БД – база даних;

ІС – інформаційна система;

ЕОМ – електронно-обчислювальна машина;

ЕОД – електронний обмін даними;

НМ – нейронна мережа;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект

ЗМІСТ

ВСТУП.....	9
Розділ 1.....	12
ЛІТЕРАТУРНИЙ ОГЛЯД ЗАСТОСУВАННЯ ЧАТ-БОТІВ.....	12
І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	12
1.1. Визначення поняття «чат-боту» та його роль в автоматизаційних процесах.....	12
1.2. Характеристика сучасних бот-платформ.....	15
1.3. Постановка задачі дослідження.....	22
Висновки за розділом 1.....	22
РОЗДІЛ 2.....	25
МЕТОДИ ТА ЗАСОБИ ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАЛІЗНИЧНИХ КВИТКІВ НА ПРИКЛАДІ ТЕЛЕГРАМ-БОТУ	25
2.1. Опис існуючих моделей та сервісів	25
2.2. Підходи до проектування чат-боту	36
2.3. Типовий алгоритм створення системи.....	44
2.4. Порівняльний аналіз мов програмування	51
Висновки за розділом 2	53
РОЗДІЛ 3.....	54
ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАЛІЗНИЧНИХ КВИТКІВ НА ПРИКЛАДІ ТЕЛЕГРАМ-БОТУ	54
3.1. Створення Telegram-боту	54
3.2. Реалізація клієнтської частини чат-бота.....	55
3.3. Тестування чат-боту.....	61
3.2. Аналіз програмної реалізації додатку	66
Висновки за розділом 3	67
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	70

	8
ДОДАТКИ.....	73
Додаток А.....	73

ВСТУП

Актуальність дослідження. На сьогоднішній день процеси комунікації потребують постійного вдосконалення та оперативності, оскільки кожного року світ все більше потребує розвитку у різних напрямках. Акцент робиться на мережі Інтернет та IT-галузі, адже сучасні технології значно спрощують життя суспільства. В цьому сенсі на допомогу людству приходять гаджети у вигляді смартфонів, комп'ютерів, сайтів, веб-додатків та мобільних додатків.

Проте особливе місце у житті суспільства посідають технології, завдяки яким людина має змогу дистанційно налагоджувати контакти з іншими людьми. Такими технологіями є месенджери та соціальні мережі.

В цьому аспекті проектування та впровадження сучасних бот-систем із застосуванням передових мобільних технологій видається дуже важливим, адже розумний бот призначений для виконання одноманітної та рекурсивної роботи та, на відміну від комп'ютерних програм, на даний момент має величезні перспективи в різноманітних галузях суспільної діяльності. У порівнянні з людьми, він має прискорену реакцію на процеси і не втомлюється від монотонних дій. Чат-бот є своєрідним помічником, що має діалог з користувачем завдяки обміну повідомленнями та виконує багато функцій одночасно. Тобто, можна отримати певні дані, написавши чат-боту спеціальну команду, яку він певним чином перекладе. Таким чином можна оперативно перекладати, коментувати, знаходити, тестувати, шукати, навчати, транслювати, вбудовуватися в інші сервісні програми, взаємодіяти з усією мережею інтернет.

Водночас, як засвідчує аналіз літературних джерел, питання формування та використання інтерактивних засобів комунікації із застосуванням автоматизованих систем вивчене недостатньо. Все ще залишаються не вирішеними в повному обсязі питання, пов'язані з розробкою методів моделювання автоматизованої системи моніторингу залізничних квитків із застосуванням мобільних технологій. Недостатньо чітко описані задачі та

алгоритми створення таких моделей, а також особливості їх реалізації в логістичній галузі.

Отже, виходячи з вищенаведеного, наше дослідження особливостей розробки та впровадження автоматизованої системи моніторингу залізничних квитків на прикладі телеграм-боту є актуальним.

Мета дослідження – аналіз особливостей розробки та практичної реалізації автоматизованої системи моніторингу залізничних квитків.

Завдання дослідження:

- 1 проаналізувати теоретичні засади застосування чат-ботів в галузі автоматизації послуг;
2. охарактеризувати сучасні бот-платформи;
3. дослідити алгоритм створення автоматизованої системи моніторингу залізничних квитків на прикладі чат-боту;
4. спроектувати модель телеграм-боту у вигляді мобільного додатку;
5. розробити методичні рекомендації щодо успішного використання засобів мобільних технологій під час створення телеграм-боту для пошуку залізничних квитків.

Об'єкт дослідження – процес проектування автоматизованої системи моніторингу залізничних квитків на прикладі мобільного додатку telegram-боту.

Предмет дослідження – особливості застосування засобів системного програмування для реалізації клієнт-сервісних систем.

Методи дослідження:

- методи системного аналізу;
- аналіз наукової літератури;
- спостереження;
- абстрагування;
- узагальнення.

Теоретично - інформаційну базу дослідження склали праці таких науковців, як В. Антонюк, Е. Вентцель, Г. Вороновський, Р. Вудс, В. Глушков, Б. Головкін, А. Дейч, С. Кашлев, В. Комашинський, В. Круглов, В. Левикін,

В. Мельник, Н. Островерхова, І. Підласий, Г. Поспелов, Д. Рутковська, Д. Тархов, Л. Ясницький та інших.

Наукова новизна одержаних результатів. Автоматизована система моніторингу квитків на основі telegram-боту – в результаті дослідження пропонується розробка альтернативного методу з використанням засобів системного програмування.

Практичне значення одержаних результатів зазначається в тому, що дослідження ґрунтується на результатах поглибленого вивчення особливостей застосування штучного інтелекту під час проектування telegram-боту для моніторингу залізничних квитків.

Структуру роботи складає вступ, три розділи, висновки, перелік джерел посилань, додатки. Загальний обсяг роботи становить 60 сторінок. Перелік джерел посилання нараховує 38 найменувань.

Розділ 1.

ЛІТЕРАТУРНИЙ ОГЛЯД ЗАСТОСУВАННЯ ЧАТ-БОТІВ

І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1. Визначення поняття «чат-боту» та його роль в автоматизаційних процесах

Telegram - це безкоштовний месенджер для смартфонів та персональних комп'ютерів під управлінням всіх найбільш поширених на сьогоднішній день операційних систем, що дозволяє обмінюватися не лише текстовими повідомленнями, а й різними медіафайлами (картинки, музика, архіви, текстові документи).

Функціонально Telegram схожий на інші месенджери, його головні переваги перед конкурентами - швидкість, захищеність, зберігання даних у хмарі (віддалений сервер) та безкоштовність.

Свій трафік Telegram надійно зашифровує, всі обчислення виконуються на віддаленому сервері, а клієнтська частина лише шифрує дані та відправляє їх на сервер. Для стійкої роботи було створено унікальний протокол MTProto, що суттєво підвищило безпеку та захист від несанкціонованої втрати інформації.

Можна побачити, що на сьогодні соціальними мережами користуються кілька мільярдів людей і згідно прогнозів, до 2021 року число користувачів збільшиться до 2,48 мільярдів. Дослідження 2019 року показало, що 4% компаній використовували чат-ботів. Месенджери забезпечують користувачів великою кількістю зручних способів онлайн-спілкування як із родичами та друзями, так і з діловими партнерами. Проведення онлайн-опитування та голосування є одним з найбільш популярних способів швидкої комунікації. Голосування та опитування

дозволяють в інтерактивному режимі організувати процес опитування та збору думок зацікавлених осіб (наприклад, клієнтів та співробітників компаній).

Згідно з дослідженням 2016 року, 80% підприємств заявили, що мають намір створити його до 2022 року (Рис. 1.1).

Телеграм-боти – наші помічники, які тримають зв'язок з користувачем через миттєвий обмін повідомленнями та мають широкий та різноманітний функціонал, як для збору інформації, так і для розсилки.

Сфера використання чат-ботів в системах зв'язку проходить в різноманітних сферах, таких як збирання інформації обслуговування людей або великих організацій, маршрутизація запитів. Хоча деякі програми чат-бота використовують розширені процеси класифікації слів, процесори природної мови шукають лише загальні слова-ключі та показують відповіді, які отримують з бібліотек через знайомі загальні фрази.

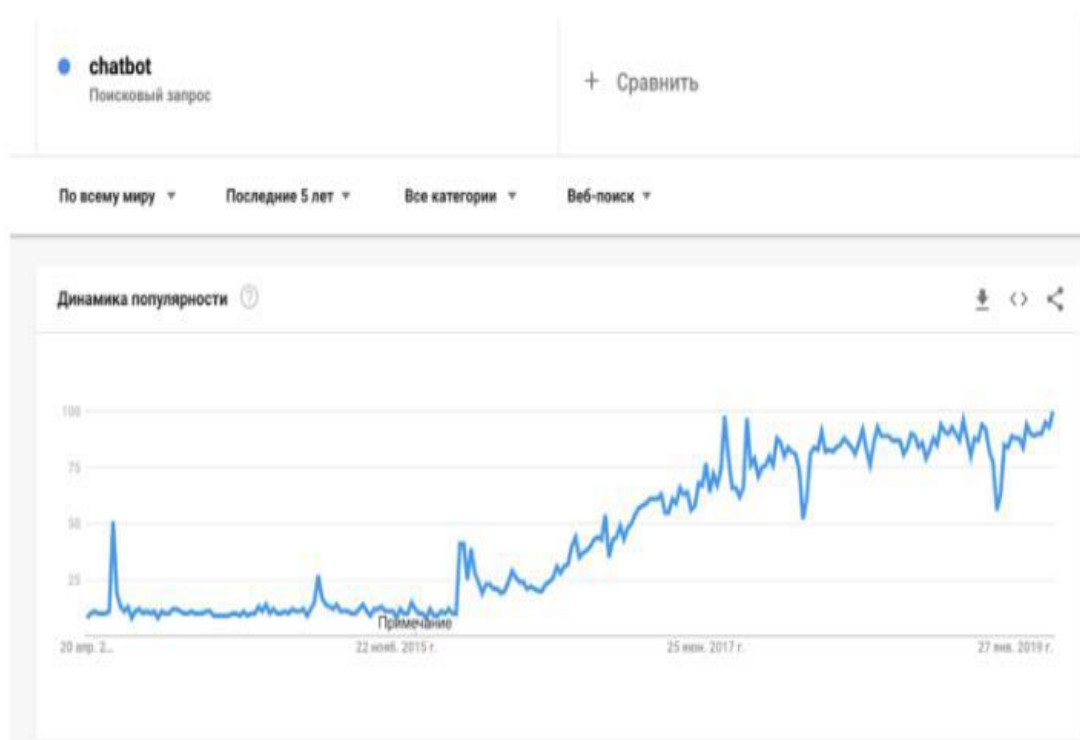


Рисунок 1.1 – Динаміка популярності чат-ботів

Бот – це програма, яка виконує різні функції та спрощує різні процеси. Він призначений для роботи від прийому повідомлень до пошуку та обробки інформації, написаної для платформи Telegram. Основна дія бота – діалог шляхом введення команд та отримання. Він працює безпосередньо через інтерфейс Telegram та імітує використання цього робота.

Їх можна розділити по таких категоріях:

- **Чат-боти** це класичний вид ботів і є найпростішим способом змодельовати спілкування за темами, що визначаються користувачем.
- **Боти-інформатори** є іншими типами ботів, і основна мета закладена в повідомленні користувача про деякі події (новини, публікації і т.д.)
- **Ігрові боти** - це боти, в яких можна грати в кількох іграх. У більшості випадків це текстові версії інших ігор.
- **Helper bot**- бот, розроблений декількома онлайн-сервісами шляхом додавання до веб-версії за замовчуванням

Надіслані користувачем повідомлення, команди та запити надсилаються до програмного забезпечення, запущеного на сервері розробників. Прихований проксі-сервер телеграми керує шифруванням та створює зворотний зв'язок між утилітами та користувачами.

Взаємодія user-bot така: користувач видає команду боту, а команда ботом відправляє на сервер. Потім програма на сервері обробляє запит з бота, і сервер відповідає за запит. Таким чином, бот відображає екран програми користувача у відповідь. І цей цикл є повторювальним.

Взаємодія з серверами відбувається за допомогою простого HTTPS-інтерфейсу, який є спрощеною версією API Telegram. Інакше цей інтерфейс можна назвати програмним каталогом чи бот-алгоритмом. Нові bot-утиліти створюються за допомогою спеціальної утиліти @BotFather.

Про популярність програм можна судити з розробки таких сервісів, як Alexa, Siri, IBM Watson та Google Now. Є також багато роботів на платформах, що використовуються розробниками програмного забезпечення для спілкування з колегами та сервісами, такими як Slack, Microsoft Team і Hip Chat (рисunek 1.2).

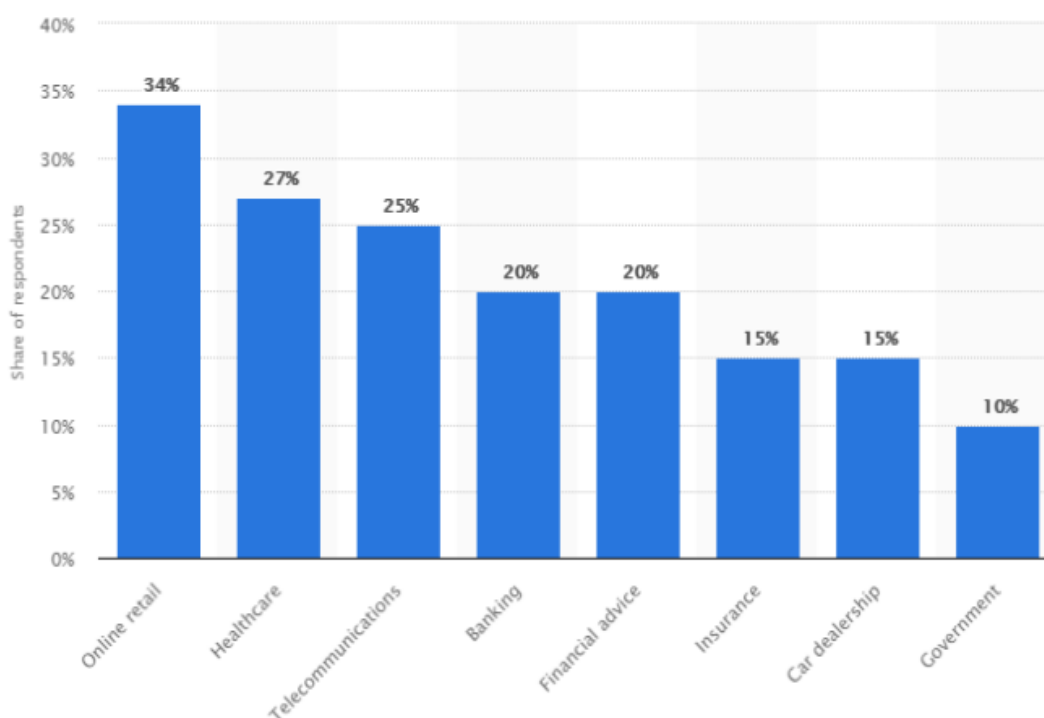


Рисунок 1.2 – Дані опитування Statista на 2021 рік [10]

Таким чином, боти стають реальними стандартами інтерфейсу взаємодії з програмними сервісами. Це платформа обміну повідомленнями (наприклад, Slack, Facebook для розробників та користувачів соціальних мереж). Широке використання і частково велика кількість ботів обумовлені прогресуванням підтримуваного природного розуміння.

Інша причина - масове використання алгоритмів машинного навчання та великих даних. Боти корисні при взаємодії із такими системами.

1.2. Характеристика сучасних бот-платформ

Згідно з дослідженням LiveWorld, три роки тому 58% маркетологів не хотіли вкладати кошти в месенджери та чат-ботів. Водночас аналітики зробили більш оптимістичні прогнози щодо автоматизованих процесів та штучного інтелекту – 85% назвали месенджер-маркетинг новим стандартом для бізнесу. І вони мали рацію.

Соціальні мережі процвітають вже майже 10 років: вони були не лише платформою для спілкування між людьми, а й для ведення бізнесу. Багато компаній завдячують своєю популярністю та фінансовим успіхом соціальним мережам.

Кардинальне зрушення відбулося у 2018 році. Близький родич соціальних мереж – Messenger – обігнав соціальні мережі за кількістю користувачів. За оцінками The Economist, люди витрачають більше двох годин на день на програми для обміну повідомленнями. Така активна аудиторія могла впливати лише на нові тенденції у маркетингу. Компанії використовували як месенджери як повноцінної платформи для спілкування з клієнтами і рекламними послугами, а й у прямих продажах. Переваги месенджерів:

1) Конфіденційність.

Соціальні мережі за своєю природою є відкритими платформами. Через них бізнесу легко транслювати свої повідомлення, спілкуватися з аудиторією, яка вже знайома з брендом, та залучати нову. Але соціальні мережі що неспроможні похвалитися надійним зберіганням конфіденційної інформації. У 2019 році Facebook заплатив найбільший в історії штраф за витік даних - п'ять мільярдів доларів.

Месенджери WhatsApp, Facebook Messenger, Viber та Telegram – це канали, якими інформація передається безпосередньо одержувачу. Крім того, більшість програм обміну миттєвими повідомлення мають додаткову наскрізну систему шифрування, що також робить цю платформу безпечною для банківського сектора.

2) Управління репутацією.

Спілкування тет-а-тет через месенджери набагато краще. По-перше, проблеми ваших клієнтів не стануть предметом обговорення для всієї аудиторії Facebook та ЗМІ. В особистому листуванні можна виконувати запит більш якісно і ймовірність, що клієнт залишиться задоволеним зростає. По-друге, при листуванні в месенджері можна виявити як позитивні якості компанії, і негативні.

3) Зручне середовище зв'язку.

В основі месенджерів – асинхронність, швидкість та всі ваші діалоги в одному місці. Поєднання цих властивостей робить їх основним цифровим засобом комунікації.

Вже є велика кількість каналів для зв'язку створених для розробників з багатьма популярними платформами, де вони беруть активну участь, обмінюються досвідом і розмовами, отримують поради і відповіді на запитання. Деякі з цих спільнот, такі як Slack's Botmaker Group і ChatbotMagazine Forum, і наразі обговорюють широкий спектр тем, пов'язаних з роботизованими програмами [26].

Якщо говорити про платформи, на яких використовують ботів, то більший відсоток належить саме месенджерам.

Коли ми говоримо про платформи, в які активно інтегруються чат-боти, то більшість пропорційного відсотку належить месенджерам.

1. Сервіси для миттєвого обміну повідомленнями.

Чат-боти багатьох компаній працюють в додатках для обміну повідомленнями або просто через SMS. Вони використовуються для обслуговування клієнтів, продажів і маркетингу B2C. Боти зазвичай з'являються як один з контактів користувача, але іноді можуть виступати в якості учасників групового чату.

Багато корпоративних чат-ботів працюють через програми для обміну повідомленнями або просто SMS. Вони використовуються для обслуговування клієнтів, продажу та продажу, B2C. Бот зазвичай відображається як один із контактів користувача, але також може виступати як учасник групового чату.

Крім того, чат-боти використовуються в якості додатків і веб-сайтів компанії. Веб-сайти компанії використовують чат-боти в якості додаткових послуг. Попередні версії чат-бота були на корпоративних веб-сайтах, таких як Alaska Airlines Ask Jenn, яка почалася в 2008 році, і агент віртуального обслуговування клієнтів Expedia, який був запусканий в 2011 році. Нове покоління чат-ботів включає IBM Watson. Rocky, представлена в лютому 2017 року в Нью-

Йорку - Rare Carat надає інформацію потенційним покупцям алмазів в електронній комерції [32].

Є боти з набором послідовностей, якими користуються маркетологи для того, щоб створити сценарії послідовностей сповіщень, які можуть бути схожі на відповіді автовідповідача.

Такі послідовності можуть бути викликані згодою користувача або використанням ключових слів при взаємодії з користувачем. Після спрацьовування тригера відбувається доставка послідовності повідомлень до наступної очікуваної відповіді користувача.

Кожен призначений для користувача відповідь використовується в дереві рішень, щоб допомогти чат-боту орієнтуватися в послідовності відповідей для доставки правильного листа у відповідь.

2. Внутрішні платформи компанії.

Компанії вивчають способи використання чат-ботів всередині організації, наприклад, для підтримки клієнтів, управління персоналом або навіть в проектах Інтернету речей (IoT). Overstock.com, наприклад, як повідомляється, запустив чат-бота Mila для автоматизації деяких простих, але трудомістких процесів при запиті відпустки через хворобу. Такі компанії як Lloyds Banking Group, Royal Bank of Scotland, Renault і Citroën тепер використовують автоматизованих онлайн-помічників замість колл-центрів з людьми, щоб забезпечити першу точку контакту [35].

Чат-боти використовуються компаніями всередині в тестовому режимі, наприклад, для адміністративного управління персоналом, комунікації з клієнтами, або в проектах з IoT. Компанія Overstock.com за інформацією запустили чат-бот Mila для того, щоб автоматизувати багато невеликих, але трудомістких процесів, наприклад, для запит на відпустку. Організації Lloyds Banking Group, Royal Bank of Scotland, Renault і Citroën стали використовувати автоматизовані онлайн-помічників замість колл-центрів з людьми, щоб забезпечити першу точку контакту [35].

З тих пір як було оголошено Марком Цукенбергом, що Facebook Messenger дозволив використання ботів у своїх додатках, бізнес створення чат-ботів перетворився на великі компанії, і впровадився в такі заклади, як лікарні та авіаційні компанії, а IT-архітектори в минулому стали більш ефективно ділитися ними, використовуючи інтелектуальні знання і досвід.

Ці розумні чат-боти використовують різні види штучного інтелекту, включаючи аудит зображень і розуміння природної мови (NLU), формування природної мови (NLG), машинне навчання і глибоке навчання.

3. Служба підтримки клієнтів.

Багато високотехнологічні банківські організації хочуть включити чат-боти (автоматизовані рішення на основі ШІ), в свою діяльність для підтримки клієнтів. Це робиться для більш швидкого реагування на запит допомоги для своїх клієнтів, також це більш дешевше без залучення зайвих операторів call-centre, тим більш, що все більше і більше людей звикають до технологій.

Зокрема, чат-боти можуть ефективно вести діалог, без потреби в інших каналах зв'язку, таких, як наприклад електронна пошта, телефон або SMS.

У банківській справі їх основне застосування пов'язано з швидким обслуговуванням клієнтів, відповідаючи на загальні запити, а також з підтримкою транзакцій.

У липні 2016 року Barclays Africa також запустив чат-бота в Facebook, ставши першим банком, який зробив це в Африці.

Переваги використання чат-ботів для взаємодії з клієнтами в банківській сфері включають зниження витрат, фінансові консультації і підтримку 24/7.

Свій розвиток інтернет-сервіси для спілкування почали з чатів, потім месенджери, потім соціальні мережі, але відтоді месенджери знову очолюють список перспективних сервісів.

Месенджер — це програма обміну миттєвими повідомленнями. Англійське слово messenger є похідним від message — повідомлення.

За рахунок використання чат-ботів, месенджери допомагають користувачу не завантажувати різні програми, що займають пам'ять смартфона або інших

технологій, що використовуються. Як описано вище, у сучасному світі компанії та фірми, які воліють переносити консультацію користувачів у месенджери, де оператори або чат-боти, працюють з клієнтами. Найпопулярнішими месенджерами в Україні можна вважати:

- Whats App,
- Viber,
- Telegtam.

Telegram - безкоштовний месенджер для телефонів і ПК, що працює на найпоширенішій сьогодні операційній системі і може обмінюватися текстовими повідомленнями, а також різними медіа-файлами (фото, музика, архіви, текстові документи).

Функціонально Telegram схожий на інші миттєві повідомлення. Основними перевагами для конкурентів є швидкість, безпека, хмарне сховище (віддалений сервер) і вільний доступ.

Свій трафік Telegram надійно зашифровує, всі обчислення виконуються на віддаленому сервері, а клієнтська частина лише шифрує дані та відправляє їх на сервер. Для стійкої роботи було створено унікальний протокол MTProto, що суттєво підвищило безпеку та захист від несанкціонованої втрати інформації.

При одночасному обміні повідомленнями та фотографіями учасників чату, конкуренти значно поступаються Telegram за швидкістю доставки інформації. Отже, Telegram доступний для всіх, хто хоче швидких і надійних повідомлень і дзвінків. На відміну від Whats App, Telegram - це хмарний месенджер з безшовною синхронізацією. Мовою розробки Telegram є C ++.

Бот Telegram створюється за допомогою іншого бота під назвою BotFather. Для цього необхідно надіслати йому команду /newbot, вибрати ім'я, яке відображатиметься у списку контактів, та адресу.

Якщо адреса не зайнята, а ім'я введено правильно, BotFather надішле у відповідь повідомлення з токеном - "ключом" для доступу до створеного бота.

Коли бот зареєстрований, його потрібно наділити функціоналом. І тому можна використовувати різні методи. Існують різні послуги, за допомогою яких

можна наділяти робота певними функціями, але для більш гнучкої налаштування знадобиться робота з мовою програмування.

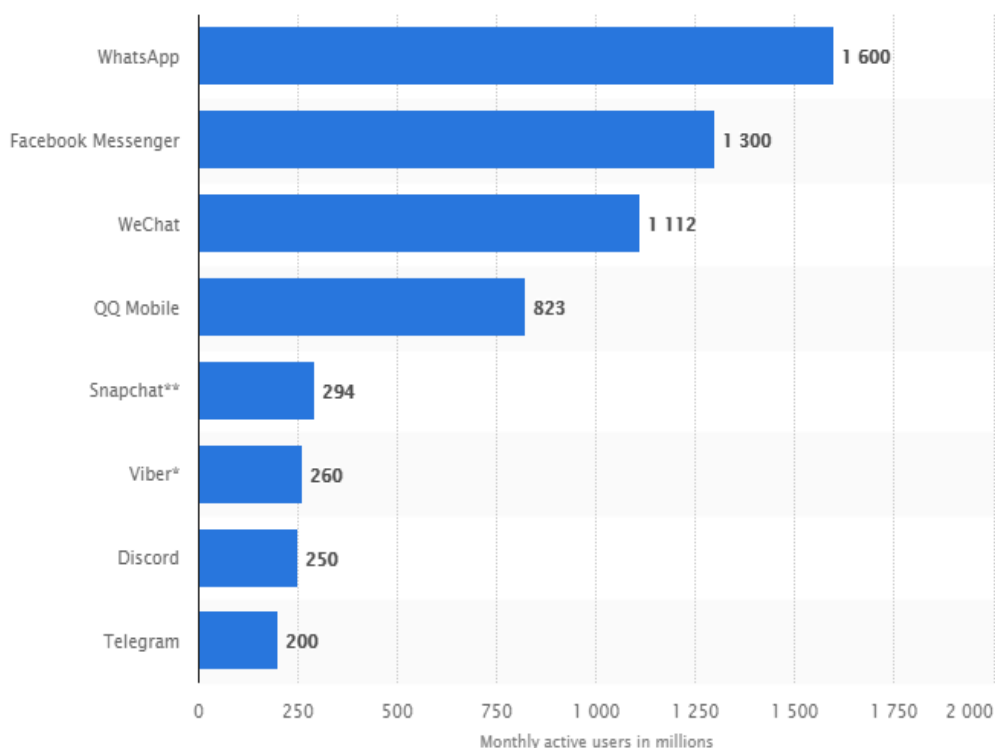
У мовах програмування існують різні фреймворки, створені задля роботи безпосередньо з Telegram. Вибір мови буде розглянуто нижче, у відповідному розділі.

За бажанням можна видалити і заблокувати бота. Важливо, щоб бот не запускав розмову, тобто щоб бот не надсилав повідомлення без дозволу користувача.

Люди проводять багато свого часу в месенджера, підтвердженням може слугувати статистика користувачів за липень 2019 рік, яка була взята з ресурсу [statista.com](https://www.statista.com) (рисунок 1.3).

Авжеж, всі платформи мають різні недоліки та переваги, але всі точно мають user friendly інтерфейс. Кожен месенджер дасть змогу створити групові чати, виконати аудіо та відео дзвінки, відправити голосові повідомлення.

Виходячи з цієї статистики, можна з упевненістю сказати, що ці сервіси є важливим фактором взаємодії з користувачами Інтернету.



Отже, розробка ботів в месенджерах почалася після того, як Telegram заснував платформу для створення акаунтів, які обробляють і відправляють повідомлення. Велика частина ботів, які ми знаємо, засновані на Telegram. Перевага їх використання в тому, що вони знаходяться в основному на самому месенджері і в соціальних мережах.

1.3. Постановка задачі дослідження

За мету беру розробку автоматизованої моделі групового навчання засобами мобільних технологій на прикладі додатку телеграм-боту.

Завдання, які повинні реалізувати в ході роботи:

1. дослідження та аналіз теоретичних засади застосування чат-ботів в галузі автоматизації послуг;
2. охарактеризувати сучасні бот-платформи;
3. проаналізувати та створити алгоритмічно автоматизовану систему моніторингу залізничних квитків на прикладі чат-боту;
4. спроектувати модель телеграм-боту у вигляді мобільного додатку;
5. розробити методичні рекомендації щодо успішного використання засобів мобільних технологій під час створення телеграм-боту для пошуку залізничних квитків.

Висновки за розділом 1

В кінці першого розділу, можна зробити такі висновки:

1. Використання чат-ботів можливе в різних сферах діяльності людини і зараз активно використовуються завдяки розвитку технологій. Головне, необхідно пам'ятати, що участь людини потрібна як мінімум для контролю чат-бота.

2. З'ясовано, що чат-боти є одним з найбільш перспективних нововведень для впровадження в автоматизовану систему управління будь-якою установою. В

тому числі використання новітніх мобільних технологій економічно обґрунтоване для процесу здійснення інтерактивного навчання з вивчення різноманітних дисциплін.

3. Такі компанії, як Microsoft, Facebook і Telegram, були досліджені на предмет надання комплексних інструментів для розробки і розповсюдження ботів. Вони надають спеціалізовані ресурси для вирішення конкретних завдань зі створення та розширення платформ для проектування чат-ботів і надають різноманітні програмні засоби, фреймворки, інструменти, програмні інтерфейси та додаткові функції.

Переваги використання ботів:

- простота в розробці: немає необхідності мати великі знання у програмуванні, щоб розробити свій бот:
- Дешевизна: для розробки робота немає необхідності виділяти велику кількість ресурсів;
- простий та інтуїтивно зрозумілий інтерфейс: основними елементами управління ботом є команди, прописані у вихідному коді бота, при введенні яких бот відтворює функціонал, вказаний для тієї чи іншої команди, введеної користувачем, так само як управління саме в Telegram-боті можуть використовуватись спеціально запрограмовані графічні елементи, наприклад, кнопки:
- високий рівень автоматизації: в роботі є можливість автоматизувати безліч різних процесів, це робить використання роботів таким популярним:
- підтримуються в багатьох месенджерах і соціальних мережах: на даний момент практично в будь-якому месенджері є можливість впровадження бота, це також підвищує популярність ботів:
- Доступність 24/7: так як сервер, на якому розміщений бот, повинен працювати в режимі 24/7, то і бот може безупинно виконувати функціонал, закладений у ньому розробником:
- можливість розробки бота зі штучним інтелектом: дана можливість дозволяє розробити більш потужні боти, які побудовані на основі нейронних

мереж, це дозволяє зробити бот розвивається і поступово навчати його. тим самим роблячи його більш корисним та самостійним;

- Миттєві результати на запити: на даний момент багато мов програмування підтримують багатопоточність, це дозволяє працювати боту одночасно з великою кількістю людей, тим самим у бота відсутній час простою, тобто. користувач не повинен очікувати поки бот обслужить іншого користувача, так як для кожного користувача виділятиметься власний потік.

Недоліки використання роботів:

- не повне розуміння запитів: оскільки боти - це машини і їм необхідно взаємодіяти з людиною, часто відбувається недорозуміння ботом запитів людини, дана проблема вирішується за рахунок розвитку бота та додавання до нього нового функціоналу. Боти ефективні, коли користувач звертається до бота з певними статичними командами, як тільки користувач починає впроваджувати сарказм або скорочення запитань у запити при взаємодії з ботом, бот починає негативно і не повноцінно реагувати, а часто і зовсім не реагує на подібні запити;

- погана імпровізація: боти добре працюють, поки розмова не виходить за межі алгоритму:

- примітивний інтерфейс: незважаючи на те, що інтерфейс роботів досить простий і зрозумілий, він має досить мізерні графічні можливості.

РОЗДІЛ 2.

МЕТОДИ ТА ЗАСОБИ ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАЛІЗНИЧНИХ КВИТКІВ НА ПРИКЛАДІ ТЕЛЕГРАМ-БОТУ

2.1. Опис існуючих моделей та сервісів

Завдяки Телеграму чат-боти отримали надзвичайну популярність. Вони стали тим інструментом, завдяки якому користувач може виконувати різні дії, не виходячи з програми і маючи всі можливості в одній програмі.

Вимоги до сервісу повинні містити найефективніші для цього завдання рішення. З цього випливає, що при вивченні та порівнянні сервісу з аналогами основними завданнями можна виділити чітке формулювання вимог до свого проекту та знаходження найуспішніших рішень у вже існуючих розробках.

Вивчення вже розроблених конкретних аналогів, розгляд їх переваг та недоліків, маркетингових моделей, технічних рішень, допоможе краще зрозуміти, яким має бути створюваний сервіс, які елементи необхідно включити, а яких немає потреби, сформулювати вимоги до проекту.

Для цього було проведено аналіз різних сервісів на підставі власного позиціонування на власних платформах, статей та відгуків про них простих користувачів, особистого досвіду використання. Основними критеріями були готова версія товару, затребуваність у цільовій аудиторії споживачів, різні технічні чи маркетингові рішення сервісів вивчення кількох підходів до реалізації товару.

Бот-акаунт в додатку здається іншим і відрізнити його від реального користувача не складно. У них немає можливості почати розмову першим, немає відміток про статус «online». Їхні імена закінчуються «bot», і від них можна отримати відповідь в будь-який час. У груповий чат можна додати бота.

Слід підкреслити, що можливості у ботів - індивідуальні. Серед них такі:

- @telerobot_bot може доомогти перекласти інтерфейс Telegram на російську або українську мови,

- @ImageSearchBot допоможе знайти будь-яку картинку за запитом,

- @weatherman_bot відображає погодні умови за містом,

- @Instasave_bot допоможе зберегти фото з Instagram або відео з YouTube,

- @ tap2bot багатофункціональний бот, який має змогу знайти фільм на вечір, знайти квиток на літак.

Існує також вбудований тип ботів (Inline). Їх не потрібно додавати в груповий чат або починати з ними бесіду. Вони працюють опціонально і, як правило, виконують одну задачу - досить набрати ім'я бота в рядку введення повідомлення і відповідний запит.

Прикладами таких ботів можуть служити @gif, @pic, @wiki, @imdb, @bold та інші (Рис. 2.1).

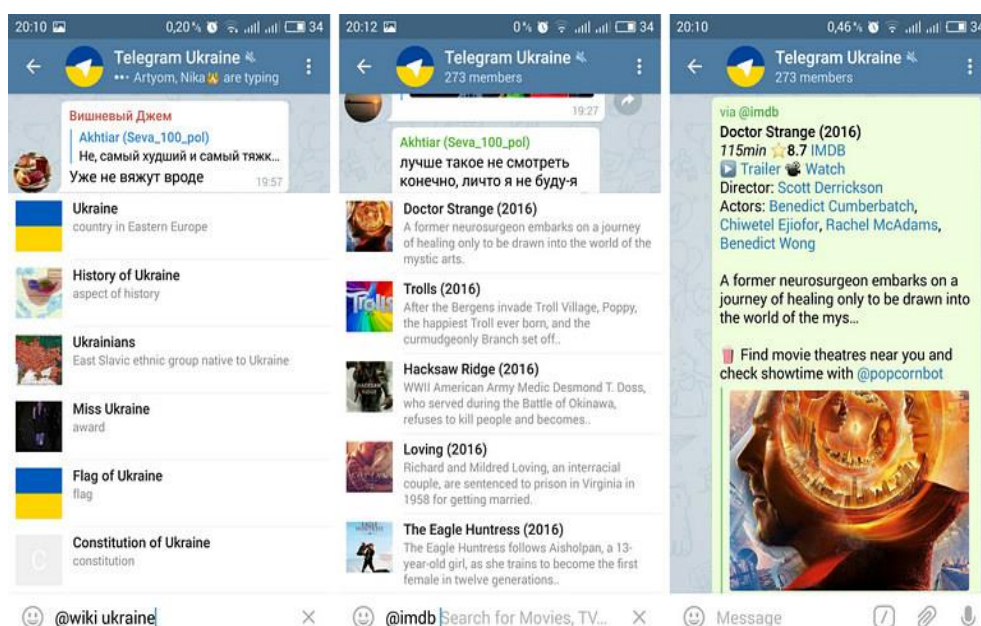


Рисунок 2.1. – Інтерфейс Telegram-бота

Всі телеграм-боти можна розділити на дві широкі категорії, залежно від технології, для ведення діалогу [37]:

1. програми з обробкою текстів на природних мовах NLP (NaturalLanguageProcessing);

2. програми, засновані на формуваннях міркувань на основі конкретних випадків CBR (Cased-BasedReasoning).

Чат-бот який обробляє текстові запити природною мовою аналізує повідомлення від користувачів за допомогою синтаксичного аналізу, що призводить до поділу пропозицій на термінальні та нетермінальні символи, які знайомі або незнайомі з програмою. У чатботів, які використовують технологію NLP, немає баз даних з чітко заданими шаблонами відповідей, реакції введених користувачем пропозицій синтезуються кожен раз відповідно до правил використаної граматики, а також структура тексту, раніше введених користувачами, ключові слова, знайдені в ній. Деякі з цих ботів спочатку не мають інтегрованої граматики, але їх можна навчати при спілкуванні з користувачами і «регулюватись і підлаштовуватись» за манерами користувача.

Найбільш відомими представниками кейс-заснованих дискусійних програм є програма A.L.I.C.E., яка була заснована в кінці 90-х рр. Р. Воллесом. В A.L.I.C.E. розробка велась на мові розробка велась на мові AIML, яка була під це заточена. AIML є зміненою версією XML.

За мету розробки мови AIML було взято забезпечення необхідних функцій для вилучення та обробки знань відповідно до конкретної шаблонної структури, а також для генерації вихідних сигналів за схемою «стимул-відповідь».

Зараз специфікації та перекладачі для перекладу мов AIML і текстів програм на стандартні мови програмування, орієнтовані на об'єкт, надаються компанією A.L.I.C.E. Вони поширюються безкоштовно на офіційному сайті. Фонд AI.

Програма містить так звані «типові» відповіді, які використовуються, коли система не знаходить потрібну модель. Успіх цих операцій залежить тільки від винахідливості розробника і завзятості користувача [25, р. 64].

Якщо відповідь використовує стандарт, складений з урахуванням психологічних аспектів діалогу і повторюється не дуже часто, користувач може не помітити змін у темі розмови, а до того ж не визнати це «слабкістю» системи.

Однак використання цих програм в системах електронної торгівлі залишається вкрай обмеженим, і в даний час ці програми використовуються лише в якості експериментальних прототипів, які широко не використовуються.

Щодо видів ботів, то на даний час є не офіційна класифікація[35]:

- бот-автоматизатор,
- бот-помічник,
- вбудований бот,
- бот-помічник, діяльність якого це оформлення замовлень,
- бот-збирач лідів.

Бот-автоматизатор розроблений для виконання декількох рутинних дій для користувача в автоматичному режимі.

Помічники здатний зрозуміти питання клієнтів і відповісти, навіть без втручання окремої людини.

Вбудовані боти - це боти з елементами штучного інтелекту, які можуть шукати інформацію швидше, ніж люди. Це корисно, коли потрібно переміщатися по великій кількості даних, таких як страхові або юридичні документи.

Бот, щоб допомогти зробити замовлення – він може знайти потрібний продукт і замовити через інтерфейс месенджера швидше, ніж можна через сайт [13, Р. 33].

Lead бот збирає дані про клієнтів, зацікавлених у замовленні, і уточнює інформацію про свої особисті дані та продукти/послуги. Він також полегшує процеси адміністраторів підприємства від регулярної роботи. Це корисно, якщо потрібно зібрати точно такий же тип даних від користувачів.

Найчастіше боти вживаються в робочих групах для автоматизації проблем, які можуть виникати і які вимагають активного залучення людей. Однак відмова від часткової взаємодії може обмежити творчість і продуктивність людей.

Тому ботів переважно використовувати для сприяння контактам і координації виконання робіт.

Наприклад, боти можуть інформувати про хід виконання завдання, сповіщати про цілі групи, пов'язувати експертів з новачками і служити справі зміцнення довіри і командного духу. При цьому неякісна реалізація або надмірне використання ботів можуть привести до інформаційного перевантаження або зайвого кількості переривань.

В ідеалі взаємодія користувача з ботом має бути комфортною і відбуватися без запинок. Досягти цього можна шляхом ретельного планування схеми бесіди, особливо для діалогових спамерських пошукових робіт.

Зокрема, бот зміг повторити вхідні команди, щоб користувачі могли правильно зрозуміти. Боти також повинні мати можливість визначати, коли діалог перервався, і давати підказки щодо можливості продовження взаємодії. Боти можуть мати елементи GUI для прискорення взаємодії.

Чатботи залежать від завдання, яке необхідно вирішити. Тому, наприклад, для комерційних чатботів, які повинні працювати під відповідальністю певних діалогових просторах, чатботам не потрібно використовувати технологію машинного навчання [26].

Цей тип чат-бота використовує базову технологію обробки мови.

- вибір ключового слово або іменованих об'єктів
- сегментація (це поділ на частини);
- stemming/лематизація.

Аналогічно методиці створення звичайної граматики за типом введення патернів, це шаблон відповідної реакції, приклад AIML [35]:

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

Якщо у вас багато статистичних даних, рекомендується використовувати методи машинного навчання, тобто методи глибокої нейронної мережі, для створення більш складних чат-ботів.

Раніше для створення чат-ботів були потрібні серйозні навички програмування та, відповідно, чат-боти в основному використовувалися у великих компаніях, оскільки на розробку чат-бота були потрібні хороші знання та кошти. Зараз існує великий вибір платформ, які дозволяють створити свій чат-бот без знання мов програмування, при цьому його можна інтегрувати у свій сайт або будь-який месенджер.

Платформи для створення чат-ботів можуть використовувати розпізнавання запитів природною мовою або штучний інтелект.

Також багато платформ пропонують інтеграцію з багатьма популярними платформами – Facebook, Текаташ, CRM системи, власні сайти.

Розглянемо декілька популярних платформ для створення чат-ботів.

1. Платформа Flow XO - дозволяє створити свого чат-бота без використання навички програмування природною мовою.

Платформа має такі переваги:

- Шаблони для ботів;
- Можливість діалогу за допомогою електронної пошти;
- Зберігання атрибутів користувача (ім'я, номер);
- можливість отримання файлів (зображення, місцезнаходження та інше)
- Створення тригерів;
- Виконання JavaScript.

Головною перевагою є інтеграція з величезною кількістю платформ - більше ста.

2. Aimylogic - це конструктор для створення чат-ботів, який заснований одночасно на штучному інтелекті та розумінні природної мови. Поєднання штучного інтелекту та розпізнавання природної мови дозволяє ефективно використовувати потенціал чат-бота (рисунок 12)

-розробники стверджують, що за допомогою їх чат-бота можна закрити до 80% питань користувачів без спілкування з оператором.

Bot API - це HTTP-інтерфейс для використання ботів у Telegram. Кожен бот - це спеціальний акаунт, створений для автоматичної обробки і відправки повідомлень. Є два способи отримати оновлення бота, логічно протилежні.

- long pulling - програма автоматично виконує зчитування з сервера Telegram для оновлення бота. Типовим значенням є 100 мс;

- webhook - як тільки на самому сервері Telegram з'явиться оновлення, програму на сервері буде повідомлено.

Майбутні оновлення будуть зберігатися на сервері до обробки, але до 24 годин. Незалежно від того, як надходять оновлення, у відповідь надсилається об'єкт послідовного оновлення JSON.

Всі запити до Telegram Bot API повинні здійснюватись через HTTPS у наступному вигляді: <https://api.telegram.org/bot<token>/НАЗВА> МЕТОДУ.

Наприклад, «/help» - команда реалізована для багатьох ботів, яка зазвичай виводить список доступних команд чи іншу необхідну користувачу інформацію для початку роботи із ботом[24].

Малюнок 2.1 ілюструє приклад відповіді на таку команду ботом BotFather, який може створити бота для Telegram.



Рисунок 2.1 – Діалогове вікно після введення команди «/help» ботом Bot Father
[36]

У месенджерах існує багато чатботів для вирішення різних проблем, деякі з них будуть відображені в малюнках.

- Paupb bot відповідальний за переказ грошей у Telegram, розроблений PrivatBank.
- «Мої платежі» - бот для оплати держпослуг через Facebook в Приватбанку.
- Зоряна - бот «Київстар» від Telegram, який надає консультації з питань зв'язку і допомагає визначати і обрати тарифи і послуги (рис. 2.2).
- Whereisnp_bot - неофіційний телеграм-бот, що трекає місцезнаходження посилки.
- телеграм-бот Q Tracker для відстеження пакетів компаній «Укрпошта», «Нова пошта», «Делівері».
- Paralegal_bot - реєстратор корпоративних юридичних фірм (Рис. 2.3).

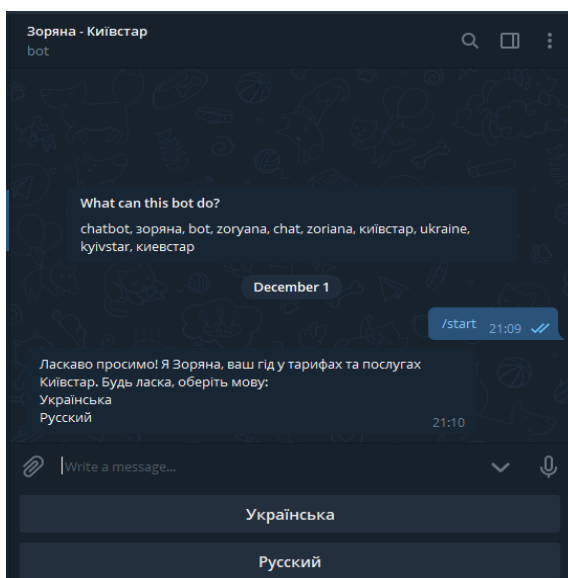


Рисунок 2.2 - Інтерфейс бота Зоряни за розробкою компанії «Київстар»

Paralegal Bot - це перше і єдине рішення для отримання процесуальних документів без реєстрації, абсолютно безкоштовно. Усний помічник Paralegal Bot може скоротити час, присвячений технічній частині написання питань і рухів і залишити більше інтелектуальної діяльності для юристів і їх колег.

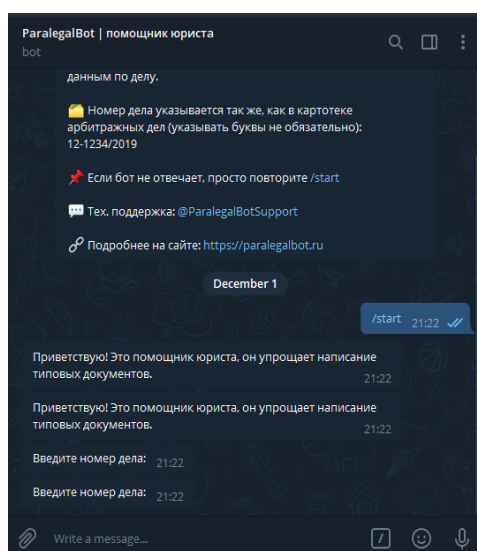


Рисунок 2.3 – Діалогове вікно бота Paralegalbot

- iGovBot - телеграм-бот, який здійснює пошук на порталі державних послуг, пошук інформації про компанії за прізвищем, кодом ОКРО або прізвищем директора, а також інформує користувачів про зміни в компаніях, на які вони підписуються.
- OpenDataBot - багатоплатформний бот, який відстежує записані дані українських компаній та судові справи.

Перевагами використання чат-ботів є наступні[4]:

1. Невисока вартість.

Для компаній економічно вигідніше використовувати чат боти, ніж наймати людські ресурси. Як і люди, різні чат-бот можуть виконувати широке коло завдань від більш простого надання консультацій в США і здійснення покупок і замовлень замість клієнтів.

Компаніям дешевше використовувати бот-чат замість найму персоналу. Як і люди, різноманітні чат-бот можуть виконувати широкий спектр завдань легше, від порад до покупки і замовлення замість користувачів.

Роблячи цю роботу, чат-бот не втомлюється і не хворіє. Тому він не вимагає тривалих відпусток, лікарняних і за роботу в нічні не потребує зайвих витрат, на відміну від працівників Людей

Роблячи все це, чат-бот ніколи не втомлюються і не хворіють. Таким чином, вони не вимагають відплатних відпусток, лікарняних і доплат за роботу в нічні зміни на відміну від працівників людей.

2. Швидкість та оперативність, а також відсутність такого явища як «черга»

Чат-боти швидко генерують точні відповіді. Це займає менше секунди, щоб відобразити необхідну інформацію з бази знань і генерувати результати. Вони не витрачають занадто багато часу на роздуми, пошук, згадування і відкриття багатьох вкладок браузеру.

Крім того, чат-боти можуть вести багато розмов одночасно, тому клієнтам не потрібно чекати чогось, що вони ненавидять. Швидка відповідь на запити забезпечує великий користувацький досвід.

3. Цілодобова доступність.

Чат-боти компанії завжди будуть підтримувати зв'язок з клієнтами і ніколи не втратять можливості заробити. Для ботів не існує перерв на обід і неробочий час. Споживачі також раді, що їхні питання як і раніше актуальні і інтерес не зник, і вони можуть отримати допомогу в будь-яку пору року, кожен день, тиждень.

API для створення чат-бота і сервісів для їх використання надають іноземні компанії, такі як Pandorabots і Syntheticcaption Network. Перший надає доступ до свого API на основі досить простої XML скриптові мови AIML, яка реалізує всі технології, необхідні для задоволення простих комерційних потреб. Вартість пропонувала від \$9 до \$999 в залежності від кількості підтримуваних спам-пошукових роботів Synthetic Intel Ligence Network пропонує концептуально

подібну і AIML-мову SIML, а також його середовище розробки ботів - Syn Chatbot Studio. NET

Поряд з тим не менш відомим фреймворками для створення ботів можна по праву вважати[8]:

- Errbot на Python,
- Hubot під Node.js.

Всі ці боти можуть бути перенесені на Slack, Telegram, IRC.

Структура обробки природної мови. Багато технологій з відкритим вихідним кодом можуть бути придатними для обробки природної мови на рівні, необхідному для створення чат-ботів, щоб полегшити роботу людських консультантів в будь-якому сервісі. Також можна створювати подібні на основі Apache Open NLP тощо. Не забувайте про існуючі API, які надають призначення сутностей для використання з чат-ботом.

Обмежене використання реалізації чат-ботів можна знайти в скриньці «Yandex SpeechKit Cloud Toolbox», який забезпечує доступ до технології голосового зв'язку і розпізнавання таких типів позначених одиниць: ПІБ, адреси, дати і часу.

Створення комерційного чат-бота не вимагає складних технологій - достатньо базових мовних прийомів.

Достатньо фреймворків і API, які використовуються для створення додаткових дуже розумних чат-ботів.

Крім того, фреймворки обробки мовлення чат-ботів можуть створюватися самостійно на базі різних бібліотек програмного забезпечення з відкритим кодом [12].

Отже, чат-боти – це дуже перспективний напрямок. Їх активне використання в месенджерах і в якості цифрових асистентів в смартфонах з великою ймовірністю призведе до популяризації UX-парадигми messaging-as-an-interface.

Так що чатбот - це доволі перспективний напрямок. Їх активне використання як цифрових помічників для миттєвих обмінів повідомлень в месенджерах в смартфонах може поширити парадигму UX message-as-an-interface.

2.2. Підходи до проектування чат-боту

На початку 2020р., у квітні Telegram представив оновлення платформи Bot 2.0, головним нововведенням якої стала інтегрована клавіатура зі попередньо заготовленими кнопками - командами. Додатково, боти обрели можливість відправляти різні типи файлів, використовувати номер телефону і розташування користувача.

Це розширило можливості ботів і спростило спілкування з ними. Ініціювати розмову з роботом дуже просто. Для цього достатньо ввести команду / start для його запуску або натиснути відповідну кнопку. У відповідь ви отримаєте інформацію про бота і список доступних для нього команд. Всі боти в месенджері безкоштовні. Однак деякі з них все ж не соромляться просити пожертви або пропонувати платні послуги [27].

Це покращило можливості бота і полегшило спілкування з ботом. Дуже легко почати розмову з чат-ботом. Для цього введіть команду/start або натисніть відповідну кнопку. В результаті ви побачите список інформації про бота і його доступні команди. Всі бот-месенджери безкоштовні. Однак деякі з них потребують пожертвувань або просять оплатити послуги [27].

Розглянемо недоліки використання чат-ботів:

Шкідливі чат-боти часто використовуються для наповнення чатів спамом і рекламою, імітуючи людську поведінку і розмови або спонукаючи людей розкрити особисту інформацію на зразок телефонних номерів банківських рахунків. Їх зазвичай можна знайти на Yahoo! Messenger, Windows Live Messenger, AOL Instant Messenger та інші протоколи обміну миттєвими повідомленнями. Також був опублікований звіт про використання чат-бота в підробленої особистої реклами на сайті служби знайомств.

Якщо алгоритм відправки тексту може видавати себе за людину, а не за чат-бота, його повідомлення буде більш достовірним.

Отже, чат-боти, схожі на людей з добре продуманою мережевий ідентифікацією, можуть почати розсилати фейковий новини, які здаються правдоподібними, наприклад, робити неправдиві заяви під час президентських виборів. При наявності достатньої кількості чат-ботів можна навіть добитися штучного соціального доказу [15].

Обмеження чат-ботів

Створення та впровадження чат-ботів все ще є що розвивається областю, в значній мірі пов'язаної зі штучним інтелектом і машинним навчанням, тому пропонувані рішення, володіючи очевидними перевагами, мають деякі важливі обмеження з точки зору функціональності і варіантів використання. Однак з часом це змінюється.

Оскільки база даних, яка використовується для генерації виведення, є фіксованою і обмеженою, чат-боти можуть давати збій при обробці незбереженого запиту.

Ефективність чат-боту значною мірою залежить від мовної обробки і обмежується неточностями і помилками.

Також чат-боти не можуть вирішити відразу кілька завдань, тому вони обмежені. Навчання чатботу вимагає великої кількості діалогових даних.

З іншого боку, чат-ботам важко керувати нелінійними розмовами, які повинні йти туди-сюди на різні теми з користувачами.

Як і технологічні зміни в існуючих послугах, деяким споживачам незручно спілкуватися з чат-ботами через їх обмежене розуміння, в основному, старшими поколіннями, і очевидно, що їхні вимоги обробляються машинами.

Можливості роботів [29]:

- можливість збирати коментарі (збір фідбеку);
- сучасна технологія обробки мов дозволяє вести складний організаційний діалог з людьми.
- інтеграція і парсинг з зовнішніми джерелами даних;

- реалізація на мобільних платформах;
- зручний і юзабіліті інтерфейс для системи діалогу;
 - вирішення бізнес-завдань, таких як технічна підтримка, допомога та просування.

Відмінності між автоматизованим і природним діалогом [32, с. 88]:

- ступінь і спосіб розуміння відповіді співрозмовника віртуальним співрозмовником;
- загальний принцип і глибина діалогу;
- характер, стилістика та інші специфічні характеристики формування відповіді.

Ці відмінності обумовлені природною обмеженістю методик NLP в порівнянні з мовними навичками людини, адже віртуальний співрозмовник має можливості, якими людина не може скористатися.

В межах рівня аналізу вхідних текстів [35]:

1. семантично-лексичні дані;
2. тематична специфіка робота;
3. жаргон, неологізми;
4. помилки.

Важливими аспектами розробки чат-бота, які потребують уваги, включають в себе:

- тестування, що призводить до відмови зв'язку;
- взаємодія віртуального співрозмовника;
- ряд цілей нарад, будь то розваги, консультації, рекомендації або реклама;
- глибина володіння конкретною темою (цільовою областю)
- мінливість, структура стилю та інші вимоги для віртуальних співрозмовників;
- комунікаційна організація діалогу.

Віртуальні співрозмовники завжди повинні швидко реагувати і відповідати, мати великі обсяги даних, не відволікатися, використовувати повторні формати, замінювати і підставляти дані з баз даних і сервісів і фіксувати позиції коректної комунікації: не намагатися симулювати людей, використовувати підказки і питань, коригувати відповіді в межах коректності на комунікативні проблеми: прохання переформулювати питання, чи не відхилятися від теми, побудова розмови відповідно до функціонального призначення конкретного бота.

Основними мінусами чат-ботів є [22]:

- Багато людей не знають, хто такий віртуальний помічник.
- Немає усталеного майданчика, для розміщення мало можливостей;
- Потужні боти недоступні для простих користувачів.
- Неповний інтерфейс.

Нерозуміння сарказму. Чат-боти призначені для взаємодії з людьми, проте останні, на відміну від ботів, схильні висловлюватися неоднозначно і часто вкладають у свої слова прихований сенс. Тому, незважаючи на те, що чат-боти прекрасно справляються з прямолінійними питаннями, генеруючи швидкі і точні відповіді, розпізнати сарказм, іронію і гумор складно навіть для ботів зі штучним інтелектом. Вони приймають всі слова за чисту монету і не вміють читати між рядків. Вчені невпинно працюють над вдосконаленням штучного інтелекту, намагаючись олюднити комп'ютерні програми, проте на даний момент їх можливості щодо цього аспекту залишають бажати кращого.

Відсутність розуміння іронії. Чатботи призначені для взаємодії з людьми, але останні, як правило, говорять неоднозначно, навпаки, від ботів, і часто додають у свої слова приховані значення. Тому, незважаючи на те, що чатботи повною мірою ставлять прості питання і дають швидкі і точні відповіді, навіть боти зі штучним розумом важко розпізнати іронію і гумор. Вони приймають всі слова в прямому значенні і не можуть читати між рядків. Розробники невпинно працюють над вдосконаленням штучного інтелекту для гуманізації комп'ютерних програм, але в даний час їх здібності в цьому аспекті залишаються недоступними.

Погана імпровізація. Чат-боти добре працюють, поки розмова не перевищує запрограмований алгоритм.

Однак поведінка користувачів без шаблонів зупиняє бота, оскільки розробникам неможливо передбачити мільярд сценаріїв. Щоб дізнатися, що мають на увазі співрозмовники, чат-боти задають неправильні питання. Якщо додаткові відповіді замовника залишаться невідомими, програма продовжить повторювати ті ж нечіткі питання.

В результаті клієнти, які засмучені і незадоволені сервісом, залишають сайт або додаток без отримання необхідної інформації.

Не для всякого роду бізнесу. Чат-бот не є унікальним рішенням і не підходить для будь-якого виду діяльності.

Його використання може бути доцільним для компаній, що пропонують такі послуги, як таксі, доставка їжі, бронювання готельних номерів, продаж квитків, роздрібна торгівля тощо.

Однак для досягнення задоволеності клієнтів і впевненості багатьом B2B компаніям необхідно надати поглиблені консультації зі своїми індивідуальними підходами і адекватно реагувати на ситуацію на місцях. На жаль, чат-боти не можуть забезпечити якісне обслуговування клієнтів у таких випадках.

Слід взяти до уваги такі нюанси, як [36]:

- є складні системи фільтрів та налаштувань;
- є досить багато різноманітних сценаріїв використання продукту, з якими один бот не впорається;
- в складних питаннях все одно необхідно підключення до діалогу оператора;
- не використовує методи самонавчання;
- є сфери, в яких програма не впорається із запитом (наприклад, складна бізнес послуга).

Для коректної роботи чат-боту необхідно мати власний сертифікат-SSL(SecureSocketsLayer), оскільки веб-хуки в Telegram працюють тільки по HTTPS (Рис. 2.4).

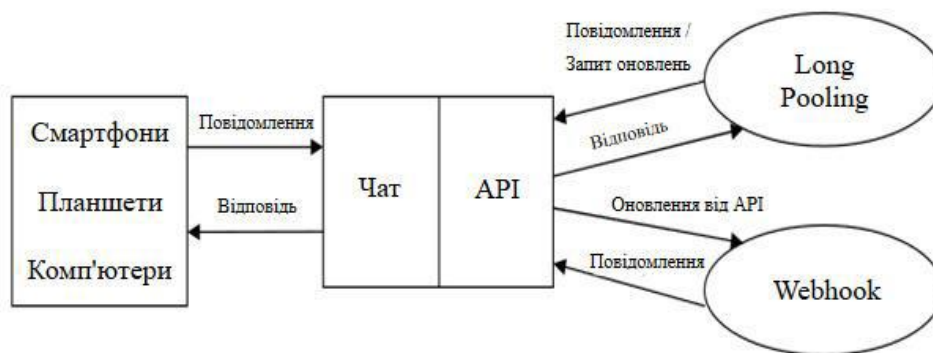


Рисунок 2.4 – Принцип роботи чат-бота на платформі Telegram

Для отримання токена необхідно написати спеціальному боту @BotFather.

Нижче опишемо приклади доступних методів для API [35]:

а) get Updates – це метод, який використовується для отримання оновлень за технологією long polling;

б) set Webhook – цей метод прив'язує доботу url домену, де міститься поточний запущений бот;

в) send Message – метод, який відправляє текстове повідомлення в клієнтській частині Telegram;

г) send Location – метод, який відправляє повідомлення з координатами місцезнаходження пристрою;

д) get File – метод, який повертає вкладені файли по їх імені.

Допускаються POST і GET запити.

Існують 4 способи передачі параметрів в Bot API:

1. запит в URL;
2. application / x-www-form-urlencoded;
3. application / json;
4. multipart / form-data (підходить для завантаження файлів).

Для роботи з TelegramBotAPI була вивчена документація, в якій описані всі методи і параметри, було встановлено, що всі відповіді приходять в JSON-форматі. В ході написання чат-бота були протестовані і використані наступні методи і типи [37]:

- Метод `get Updates` використовується для отримання оновлень через `longpolling`. Відповідь повертається у вигляді масиву об'єктів `Update`.

- Метод `set Webhook` необхідний для завдання URL веб-хука, на який бот буде відправляти оновлення. Кожного разу при отриманні оновлення на цю адресу буде відправлений `HTTPS POST` з серіалізованим в `JSON` об'єктом `Update`.

При помилці запиту до сервера спроба буде повторена декілька разів.

Для більшої безпеки рекомендується включити токен в URL веб-хука, наприклад, так: `https://your web hook server.com/<token>`. Ми можемо бути впевнені, що запити до веб-хуку відправляє саме Telegram, оскільки ніхто сторонній не знає токена.

Метод `send Message` використовується для відправлення повідомлень.

Метод `send Photo` використовується для відправки фото.

Метод `edit MessageText` використовується для редагування текстових повідомлень, які були відправлені ботом або через бота.

Об'єкт типу `User` надає інформацію про користувача Telegram.

Об'єкт типу `Chat` являє собою інформацію про чат.

Об'єкт типу `Message` являє собою інформацію про повідомлення.

Об'єкт типу `Reply Key board Mark up` являє собою клавіатуру з опціями відповіді.

Об'єкт типу `Key board Button` є ще однією кнопкою в клавіатурі відповіді.

Для звичайних текстових кнопок цей об'єкт може бути змінений на рядок, що містить текст на кнопці.

`Mongo DB` – це база даних, яка є документо-орієнтована з відкритим вихідним кодом і вона не вимагає опису схем таблиць. Головні риси та Можливості [23]:

а) підтримка відмовостійкості і масштабованості;

б) асинхронна реплікація, набір реплік і розподілу бази даних на вузли;

в) `JSON`-подібна схема зберігання даних;

- г) ефективно зберігання великих об'єктів, адміністративний інтерфейс, серверні функції, Map / Reduce і інше;
- д) використання Javascript в якості мови для формування запитів;
- е) профілювання запитів;
- ж) широкий набір атомарних операцій над даними (умовний пошук, складна вставка / оновлення тощо);
- з) різні типи даних (в тому числі підтримка масивів); підтримка індексів (B-Tree);
- и) повнотекстовий пошук, в тому числі українською мовою, з підтримкою морфології;
- к) журнал операцій, що змінюють дані в базі даних.

Mongo DB пропонує документо-орієнтовану модель даних на відміну від реляційних баз даних, завдяки цьому вона працює в рази швидше, має кращу масштабованість та її легше використовувати.

Однак, враховуючи всі недоліки традиційних баз даних і переваги Mongo DB, важливим розумінням є те, що завдання бувають різні і методи їх вирішення бувають також різні. В певній ситуації Mongo DB дійсно здатна поліпшити продуктивність програми, для прикладу візьмемо ситуацію, що існує потреба зберігати складні за структурою дані. В іншій же ситуації краще буде використовувати традиційні реляційні бази даних.

MongoDB може представляти не тільки єдину базу даних, що знаходиться на одному фізичному сервері. Функціональність Mongo DB дозволяє розташувати декілька баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними і зберігати цілісність [22].

Однак, враховуючи переваги і недоліки баз даних і Mongo DB, важливо розуміти, що завдання відрізняються і методи їх вирішення теж різні. У деяких ситуаціях Mongo DB дійсно може поліпшити продуктивні програми, наприклад, взяти ситуацію, коли виникає необхідність структурно збережених даних. В іншій ситуації краще використовувати реляційну базу даних.

MongoDB може представляти більше однієї бази даних на одному фізичному сервері. Функція Mongo DB дозволяє розміщувати кілька баз даних на декількох фізичних серверах, дозволяючи їм легко спілкуватися і підтримувати цілісність. [22]

У процесі створення телеграм-бота була розроблена схема взаємодії користувача з базою даних для відповіді на її повідомлення (рисунок 2.5).

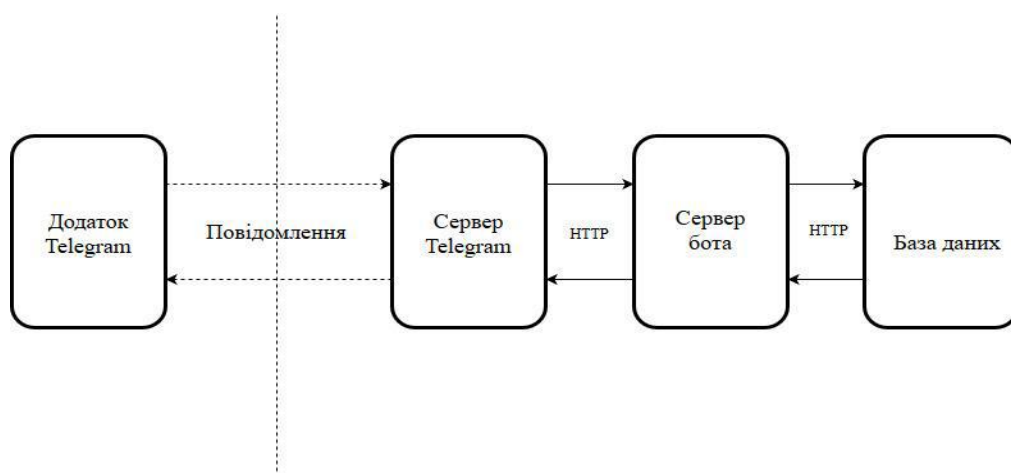


Рисунок 2.5- Схема взаємодії користувача з БД

Слід уточнити етап створення нового бота і використання API, що надається Telegram. Згідно інструкції з офіційного сайту, для створення бота мінімально необхідні такі пункти:

1. додати бота «Bot Father» досебе в чат, який дозволяє керувати створенням і налаштуванням нових ботів всередині Telegram;
2. надіслати боту команду «/newbot», після чого той запросить повне ікоротке ім'я створюваного бота (повне ім'я використовується для відображення користувачеві назви бота в заголовку чату, коротке ім'я для посилання швидких команд) і згенерує унікальний токен авторизації, який буде потрібен для ідентифікації бота і взаємодії через TelegramAPI при відправці запитів.

2.3. Типовий алгоритм створення системи

Алгоритм - це детальний набір інструкцій для виконання завдань або усунення неполадок. Технічно ваш ПК видає детальні інструкції для виконання завдань за допомогою алгоритмів. З точки зору ефективності, різні алгоритми можуть виконувати операції легко і швидко або вирішувати проблеми.

Повноцінна розробка нового програмного рішення є доволі ресурсовитратною та вимагає залучення додаткових фахівців, щонеухильно тягне за собою надмірні витрати часу.

Вищевикладені вимоги можуть бути реалізовані в рамках існуючих програмних продуктів.

В даному проекті раціональніше за все буде використовувати вже наявні системи.

Впровадження готового рішення дозволить скоротити час, оскільки відсутні етапи розробки та налагодження програми.

Таким чином, поставлена задача зводиться до вибору системи (комплексу систем), що задовольняє заданим вимогам.

Запропонуємо загальний алгоритм створення чат-боту для мобільного додатку Telegram виглядає наступним чином:

1. Розробка блок-схеми та визначення ключової мети створення боту.
2. Проектування системи управління з конструктором сценарію чат- боту.

Складність розробки системи управління ботом залежить від завдань бота і його функціоналу. Також, складність системи залежить від наявності або відсутності в ній таких функцій, як: конструктор моделювання сценаріїв бота, можливість підключення оператора.

3. Створення чат-боту.

На етапі розробки скрипта чат-бота (на мові PHP), реалізується весь необхідний функціонал і логіка бота згідно із затвердженою блок-схемою. Попередньо створюється прототип чат-бота на конструкторі спамерських пошукових роботів.

Розробка бота реалізується під різні месенджери, зокрема, такі як: telegram, viber, fbmessenger, whatsapp і інші.

Незважаючи на те, що написання ботів під різні месенджери вимагає написання різного програмного коду, управління ними відбувається через одну панель управління. Також, створюються шаблони для чат-бота.

4. Синхронізація.

Чат-боти, які вимагають синхронізації зі сторонніми базами і системами - це масштабні проекти.

Під них пишеться API, за допомогою якого відбувається процес передачі даних. API може синхронізувати бота безпосередньо або через систему управління.

Для створення алгоритму необхідно спочатку вивчити основну структуру побудови компонента чат-бота і його взаємодії, для цього можна використовувати загальну схему.

В алгоритмі розробки бота на першому кроці потрібно дізнатися яка платформа використовується користувачем, далі проводиться дослідження обраної платформи, перевірка чи є вже готова API, чи все ж потрібно запрограмувати самостійно (або переконатися в тому, що є вже готова API на обраній мові програмування).

У алгоритмі розробки бота на першому кроці потрібно з'ясувати, яку платформу використовує користувач. Після цього вибрана платформа розглядається чи є готовий API. Чи прийдеться програмувати її самостійно (можливо, API вже готова на обраній мові програмування).

Це залишається проблемою з типом нейронної мережі, що використовується для тренування умовних чат-ботів. Послідовність важлива в людських мовах. Давайте зробимо деякі висновки і прийнемо рішення на основі того, що ми сказали в преамбулі і параграфі [21].

Поточна нейронна мережа (RNN) добре підходить для цих критеріїв, але в міру збільшення відстані між спорідненими частинами тексту розмір RNN повинен збільшуватися, що знизить якість обробки інформації. LSTM (Довга короткострокова пам'ять) вирішує цю проблему. Мається на увазі одну важливу особливість - стан клітини, який може залишатися постійним, або змінюватися

при необхідності. Таким чином, інформація в ланцюзі не втрачається, що є критичним для природної обробки мови.

Поточна нейронна мережа (RNN) відповідає цим критеріям, але у міру збільшення відстані між відповідними частинами тексту збільшується розмір RNN і знижується якість обробки інформації. LSTM (Long Short Term Memory) може вирішити цю проблему. Маємо на увазі важливу функцію - стан клітини, залишається постійним, або може бути змінена при необхідності. Тому інформація в ланцюжку не втрачається, що є доволі критичним.

Сьогодні існує багато бібліотек для обробки природних мов. Якщо говорити про мову Python, яка часто використовується для аналізу даних, це NLTK та Spacy. Великі компанії також беруть участь у розробці бібліотек NLP, таких як NLP архітектор Intel або дослідники Facebook та Uber в PyTorch. Незважаючи на такий великий інтерес до нейронних мережевих методів обробки мови великими компаніями, взаємопов'язані діалоги будуються переважно на основі класичних методів, а мережа відіграє допоміжну роль у вирішенні проблеми попередньої обробки та класифікації мови.

Вибір правильного алгоритму машинного навчання вважається важливим моментом у розвитку систем передбачення. У більшості випадків цей вибір залежить від деталей аналізованих даних та умов аналізу.

Всі алгоритми машинного навчання поділяються на три типи: те, яким можна управляти, поза контролем і навчання з підкріпленням.

Контрольне навчання застосовується при наявності таблиці даних з елементами з конкретними явними властивостями, але в іншій таблиці ці властивості неявні і повинні надаватися під час роботи. Неконтрольоване навчання, придатне для неявної ідентифікації

Підтримка навчання - це симбіоз вищезазначених категорій, а його суть полягає у наявності певного зворотного зв'язку, який доступний для кожного кроку чи дії, але не має будь-якого повідомлення про розмітку та помилку.

У разі аналізу запитань існує база питань та відповідей, а також чіткі ознаки у вигляді наявності певних слів у питанні.

У зв'язку з цим було прийнято рішення використовувати один із керованих алгоритмів [9].

Далі наведені основні завдання, які можна вирішити за допомогою таких алгоритмів.

Підтримка з навчання - це поєднання вищевказаних категорій, суть яких полягає в наявності конкретного зворотного зв'язку, який доступний для кожного кроку або дії, але не має повідомлення маркування та помилки.

Для аналізу питань є чіткі ознаки у вигляді бази питань і відповідей і наявності конкретного слова в питанні.

У зв'язку з цим було вирішено використовувати один з керованих алгоритмів[9].

Основні проблеми, які можна вирішити за допомогою цих алгоритмів:

1. Класифікація.

Ця операція визначає, чи належить об'єкт, що досліджується, певному класу, підмножини. Ця класифікація називається двочленною/двокласовою якщо визначення відбувається за наявністю двох критеріїв.

Наприклад, при пошуку серед багатьох подібних питань, коли існує багато категорій, класифікація називається багатоскладовою.

2. Регресія.

Це завдання базується на прогнозуванні масштабів певних атрибутів у певних записах наборів даних.

3. Виявлення аномалії.

У такому випадку завдання полягає у визначенні точки даних, яка була вилучена зі спільного набору кількома атрибутами. Приклади таких атрибутів: виявлення шахрайства з банківською карткою - це, наприклад, платіжна операція з дуже різними функціями, будь то за місцем виведення, часом або кількістю транзакцій та сумой.

Звичайно, класифікація є оптимальним рішенням, оскільки вимагає занадто великої обробки даних для вирішення проблем регресії в поточному проекті і

далеко не найточніші результати, оскільки багато питань можуть мати однакову участь в даній темі.

Наступний крок - вибір відповідного алгоритму класифікації класу, а головну роль відіграє те, що наявність конкретного слова є ознакою питання. Важливо, щоб більш короткі і значущі слова відігравали важливу роль у класифікації, а недоречні слова вживалися пізніше.

Алгоритм дерева рішень – майже повністю відповідає заданій моделі. Вона заснована на ознаках більш високої ваги і близька до кінця гілок більш низької ваги.

Одна з перших версій проекту базувалася на цьому алгоритмі, але незабаром стало зрозуміло, що якість передбачення дуже низька. Це спочатку пов'язано з тим, що набір неважливих слів, про які йде мова, створює велике відволікання в вибірці, а по-друге, невеликий масштаб навчального зразка призводить до явища, званого переключенням.

По-перше, здійснюється так звана обробка бази діалогу та її продовження, після чого залишається лише набір, що відповідає поточній фазі діалогу. Було запропоновано розробити чат-бота для мобільних додатків Telegram для перегляду підходу до розробки чат-ботів.

Запропоновано розробити чат-бота для мобільних додатків Telegram для візуалізації підходу до розробки чат-ботів. Ми будемо використовувати[11

- JSON для обміну інформацією (так як бот повертає дані в цьому вигляді);
- Node.js для серверної частини;
- API APP ChatBot.

Розглянемо дані інструменти для розробки чат-боту детальніше.

1. JSON - стандартний текстовий формат відображення структурованих даних на основі синтаксису об'єктів JavaScript. Як правило він використовується для передачі даних у веб-додатки (наприклад, для відправки даних з сервера на клієнт, щоб його можна було переглянути на веб-сторінці або навпаки).

з сервера на клієнт, тому їх можна відображати на веб-сторінці або навпаки).

2.Node.js - це платформа, побудована на JavaScript, що дозволяє легко створювати швидкі, масштабовані мережеві додатки. Node.js використовує модель, керовану подіями, яка не блокує введення-виведення, що робить її ідеальною для використання в режимі реального часу додатків, що працюють на розподілених пристроях.

3. 3. APP ChatBot - це система розуміння мови, яка дозволяє всім службам бути розумним чатом. Легко інтегруйте ваші улюблені програми обміну повідомленнями зі створеними ботами і продовжуйте надавати послуги клієнтам.

Кожен запит API вимагає аутентифікації для визначення ліцензії, відповідальної за подання заявки. Аутентифікація надається маркеру доступу (Мал. 2.7).

```
var http = require("https");

var options = {
  "method": "GET",
  "hostname": "api.chatbot.com",
  "port": null,
  "path": "/stories",
  "headers": {
    "authorization": "Bearer ${DEVELOPER_ACCESS_TOKEN}"
  }
};

var req = http.request(options, function (res) {
  var chunks = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });

  res.on("end", function () {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });
});

req.end();
```

*Рисунок 2.7 – Зовнішній вигляд інтерфейсу запиту аутентифікації
(за кодом Nodejs)*

Отже, складність розробки автоматизованої системи групового навчання з використанням мобільних технологій на прикладі проектування телеграм-боту залежить від завдань бота і його функціоналу. Також, складність системи залежить від наявності або відсутності в ній таких функцій, як: конструктор моделювання сценаріїв бота, можливість підключення оператора.

2.4 Порівняльний аналіз мов програмування

Python - передова загальна мова програмування, яка також використовується для розробки веб-додатків. Мова призначена для підвищення продуктивності розробників і читабельності коду.

Python підтримує різноманітні парадигми програмування: структурну, об'єктно-орієнтовану, функціональну, імперативну та аспект-орієнтовану. Мова включає динамічні типи, автоматичне управління пам'яттю, повне самонавчання, обробку винятків, багатопотокову підтримку комп'ютера та практичні передові структури даних високого рівня.

Переваги Python:

- відкрита технологія;
- вчитися дуже легко, особливо на ранніх стадіях
- функція синтаксису стимулює програмістів добре писати читабельний код;
- забезпечує засоби швидкого прототипування та динамічної семантики.
- має велику спільноту, яка є позитивною для початківців.
- безліч корисних бібліотек та розширень мови можна легко використовувати у своїх проектах завдяки гранично уніфікованому механізму імпорту та програмним інтерфейсам;
- механізми модульності добре продумані та можуть бути легко використані;
- завдяки дуже уніфікованому механізму імпорту та програмному інтерфейсу, багато зручних бібліотек та мовних розширень можуть бути легко використані в проекті.

Недоліки Python:

- підтримка багатопотоковості виявилася невдалою;;
- на Python створено небагато високоякісних програмних проектів порівняно з іншими поширеними мовами програмування, такими як Java;

- відсутність комерційної підтримки інструментів розробки (хоча з тих пір це змінилося);
- спочатку обмежене фінансування унеможливлювало використання бази даних;
- стандарти показують продуктивність Python нижче базової Java VM, що робить репутацію мови повільною.

JavaScript - мова програмування з декількома парадигмами. Підтримує об'єктно-орієнтовані, обов'язкові та функціональні стилі.

Переваги JavaScript:

- Останній браузер підтримують JavaScript..
- Навіть не фахівці можуть впоратися з використанням плагінів і скриптів, написаних на JavaScript.
- Зручні налаштування функцій
- Завжди вдосконалюється мова - бета-версія проекту розробляється JavaScript 2.
- Взаємодія з програмою також може здійснюватися через текстовий редактор - Microsoft Office і Open Office.
- Погляд на використання мови в процесі викладання програмування та інформатики.

Недоліки JavaScript:

- Зниження безпеки за рахунок широкого і вільного доступу до вихідного коду загальних скриптів.
- Багато маленьких прикрих помилок на всіх етапах роботи. Більшість з них легко видозмінюються, і їх існування дозволяє вважати цю мову менш професійною.
- Повсюдно використовується. Унікальним недоліком можна вважати той факт, що деяких активно використовуваних програм (особливо додатків) більше не буде.

Щоб зробити вибір мови програмування для чат-бота, потрібно зрозуміти, що таке чат-бот. Чат-бот – це програма, яка з'ясовує потреби користувачів, а потім допомагає задовольнити їх.

За статистикою та моїм особистим аналізом та пошуком в інтернеті, я з'ясувала, що найчастіше в якості мови програмування для чат-бота вибирають Python. Наведемо кілька прикладів переваг створення бота на Python:

- Ваші можливості практично безмежні.
- Економія грошей.
- Здобуття навичок, які можуть стати в нагоді - як мінімум, ви можете заробляти на створенні «роботів».
- Недоліки:
- Витрата часу та сил на вивчення мови програмування.

Підіб'ємо підсумки, у кожної мови програмування є свої плюси та мінуси, кожна мова хороша для конкретних цілей, наприклад якщо створювати сайт, то за описаними критеріями потрібно вибрати JavaScript, а для написання чат-бота в месенджері Telegram більше підійде Python так як він більш простий у написанні, має багато документації та має великий вибір бібліотек, які допоможуть створити гарний сервіс.

В підсумку, кожна мова програмування має свої сильні сторони і недоліки, і кожна мова підходить для певної мети. Наприклад, при створенні сайту JavaScript буде гарним помічником, але Python добре підходить для написання чат-ботів в месенджері Telegram, за умови поки він простіший в написанні, з великою кількістю документів і великим вибором бібліотек, він може допомогти створити хороший сервіс.

Висновки за розділом 2

Підсумовуючи другий розділ, можемо зробити такі висновки:

1. Проаналізовано існуючі боти та наведено приклади їх застосування. В результаті, було виявлено необхідність популяризації використання розпізнавання команд із повідомлень користувачів при використанні ботів.

2. Окреслено підходи до проектування телеграм-боту, визначено їх переваги та недоліки.

3. Формалізовано типовий алгоритм вибору інструментів для моделювання чат-боту під час проектування автоматизованої системи для моніторингу залізничних квитків.

РОЗДІЛ 3.

ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАЛІЗНИЧНИХ КВИТКІВ НА ПРИКЛАДІ ТЕЛЕГРАМ- БОТУ

3.1. Створення Telegram-боту

Бот Telegram створюється за допомогою іншого бота під назвою BotFather. Для цього необхідно надіслати йому команду `/newbot`, вибрати ім'я, яке відобразатиметься у списку контактів, та адресу.

Якщо адреса не зайнята, а ім'я введено правильно, BotFather надішле у відповідь повідомлення з токеном - "ключом" для доступу до створеного бота.

Коли бот зареєстрований, його потрібно наділити функціоналом. І тому можна використовувати різні методи. Існують різні послуги, за допомогою яких можна наділяти робота певними функціями, але для більш гнучкої налаштування знадобиться робота з мовою програмування.

BotFather. Налаштування бота. Спочатку потрібно зареєструвати саму оболонку робота в самому месенджері, отримати присвоєний їй ID, який є також токеном. У телеграмі допомогу при такій реєстрації надає віртуальний помічник @BotFather.

1. Введіть команду `/start` для запиту інструкції.
2. Надсилання установки `/newbot`.

3. @BotFather пропонує вигадати ім'я для створюваного бота. Воно повинне закінчуватися словом "bot".

4. Далі помічник надсилає токен нового віртуального користувача та url для додавання його до списку контактів.

За допомогою BotFather можна додати опис створеного чат-бота, додати зображення профілю, змінити ім'я, присвоїти текст, який відображається при першому відкритті чат-бота та ін.

Після налаштувань Telegram та отримання токена можна розпочати розробку програмної частини бота. Для розробки використовувалися необхідні бібліотеки мовою програмування Python.

В першу чергу бібліотеки необхідні для підключення чат-бота до бібліотеки. Під час розробки код вказується токен створеного чат-бота.

Отриману комбінацію бажано протестувати цим посиланням: api.telegram.org/bot/getMe.

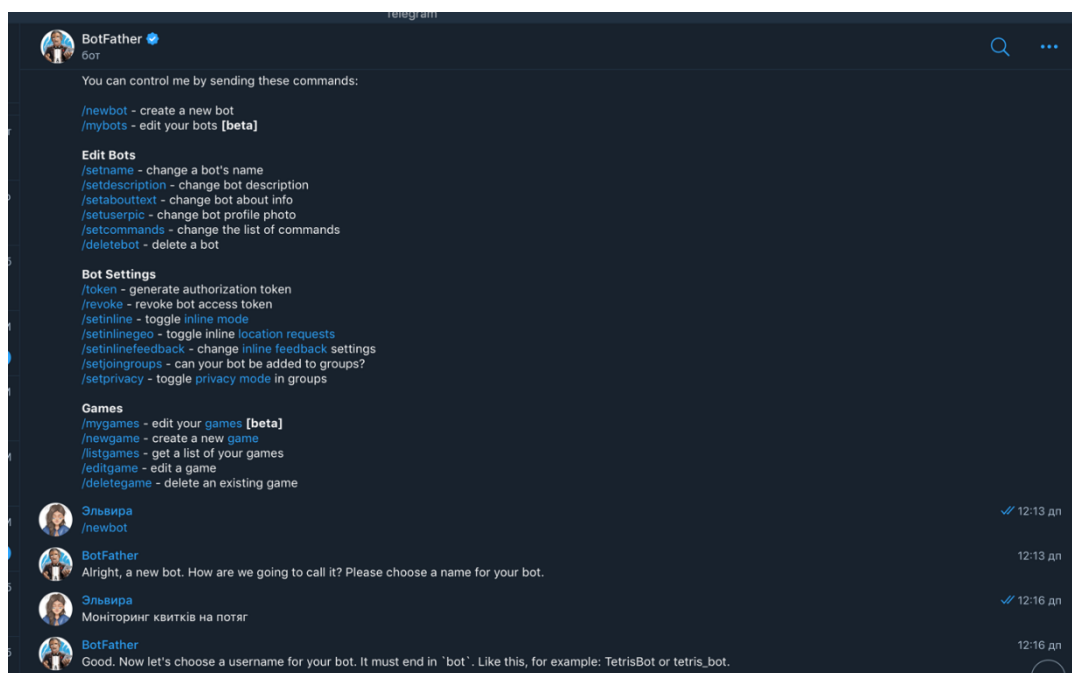


Рисунок 3.1 – Реєстрація робота

3.2 Реалізація клієнтської частини чат-бота

Користувачі Telegram можуть взаємодіяти з чат-ботами кількома способами: використовуючи команди з параметрами або вбудовані клавіатури.

Для зручності користувачів було вирішено зробити інтерфейс із вбудованою клавіатурою. Для реалізації поставленої мети необхідно реалізувати кілька меню зі своїми параметрами. На малюнку зображено початок роботи з ботом, після того, як йому тільки дали опис і встановили зображення профілю.

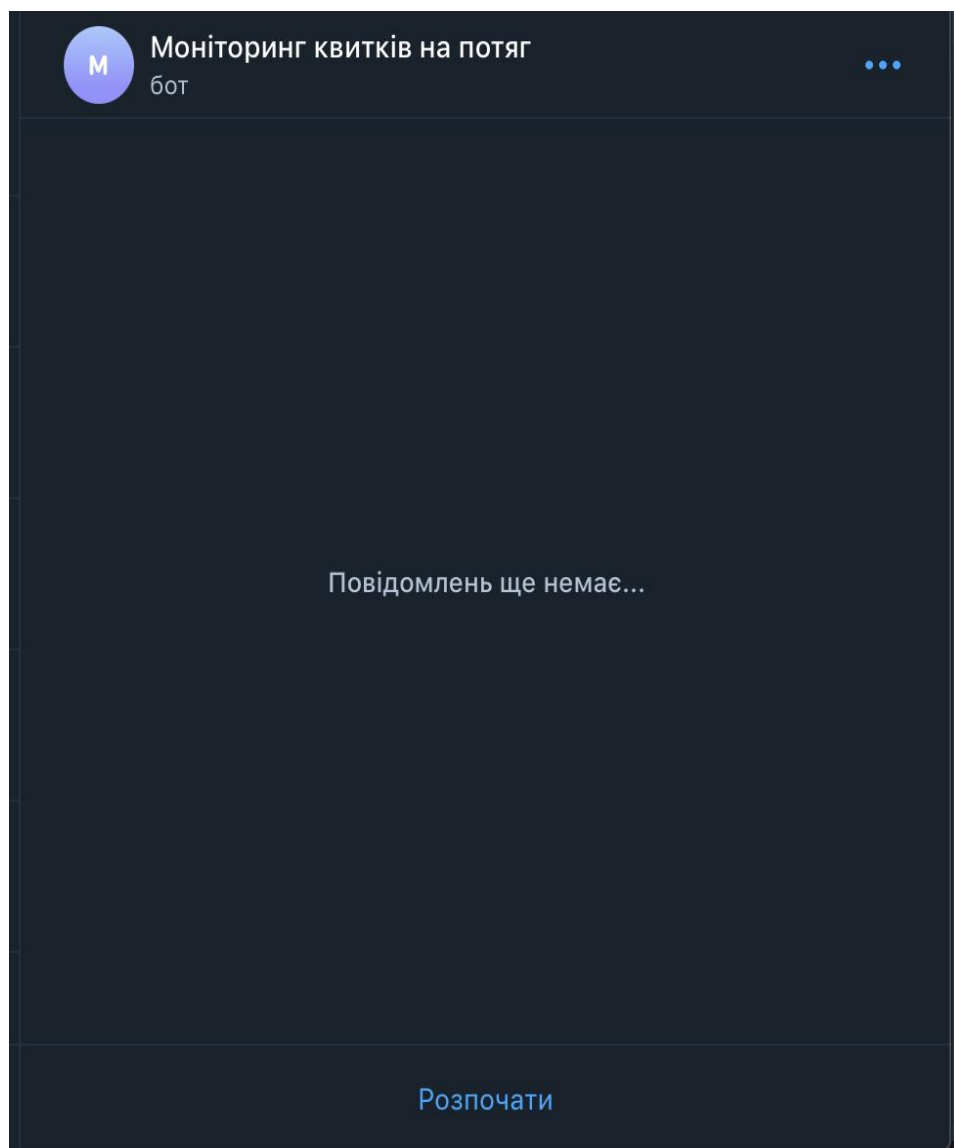


Рисунок 3.2 – Початковий екран

- Основне меню;
- Меню типу вантажу;
- меню додаткової інформації;
- Меню зворотнього зв'язку.

Початок роботи з чатом починається з команди /start. Ця команда активує команду і бот починає діалог із користувачем. Код початку роботи представлений малюнку 3.2.

```

tickets_parser = TicketsParser()
bot = Bot(token='5969845737:AAEa38KiJLCLW9J8p1nejY0FQ6IxFautIIY')
storage = MemoryStorage()

dp = Dispatcher(bot=bot, storage=storage)

async def monitoring_sending_data():
    users_list = tickets_parser.monitoring_database.get_all_users()
    for user in users_list:
        tickets_info = \
            tickets_parser.get_tickets(departure_station=user[1], arrival_station=user[2], departure_date=user[3])
        if tickets_info:
            user_id = user[0]
            data = tickets_info
            await bot.send_message(chat_id=user_id, text=data)
            tickets_parser.monitoring_database.delete_user(user_id=user_id)

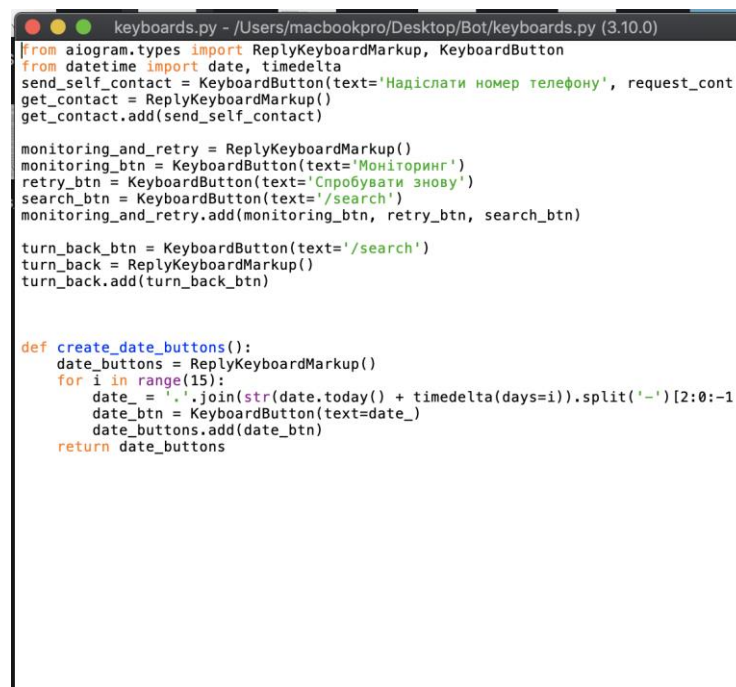
@dps.message_handler(commands=['start'])
async def greeting(message: types.Message):
    await bot.send_message(chat_id=message.from_id, text='Привіт, я допоможу купити та відстежити квитки на поїзд.'
        ' А також розповім про розклад з маршрутами.')

    time.sleep(0.6)
    await bot.send_message(chat_id=message.from_id, text='Щоб почати необхідна активація бота через номер телефону.')
    time.sleep(0.6)
    await bot.send_message(chat_id=message.from_id,
        text='Натисніть кнопку "■ Активувати" нижче або просто напишіть номер 📞',
        reply_markup=get_contact())

```

Рисунок 3.3 - Код привітання

Далі слідує реалізація меню, через яке відбуватиметься взаємодія з створеним чат-ботом. Нижче наведено основне меню (рисунок 3.4). За допомогою підключення деяких бібліотек до меню можна додавати значки, які є у списку месенджера Telegram. Піктограми потрібні для оформлення деяких деталей меню.



```

keyboards.py - /Users/macbookpro/Desktop/Bot/keyboards.py (3.10.0)
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton
from datetime import date, timedelta
send_self_contact = KeyboardButton(text='Надіслати номер телефону', request_contact=True)
get_contact = ReplyKeyboardMarkup()
get_contact.add(send_self_contact)

monitoring_and_retry = ReplyKeyboardMarkup()
monitoring_btn = KeyboardButton(text='Моніторинг')
retry_btn = KeyboardButton(text='Спробувати знову')
search_btn = KeyboardButton(text='/search')
monitoring_and_retry.add(monitoring_btn, retry_btn, search_btn)

turn_back_btn = KeyboardButton(text='/search')
turn_back = ReplyKeyboardMarkup()
turn_back.add(turn_back_btn)

def create_date_buttons():
    date_buttons = ReplyKeyboardMarkup()
    for i in range(15):
        date_ = '.'.join(str(date.today() + timedelta(days=i)).split('-')[2:0:-1])
        date_btn = KeyboardButton(text=date_)
        date_buttons.add(date_btn)
    return date_buttons

```

Рисунок 3.4 – Програмна реалізація активного меню

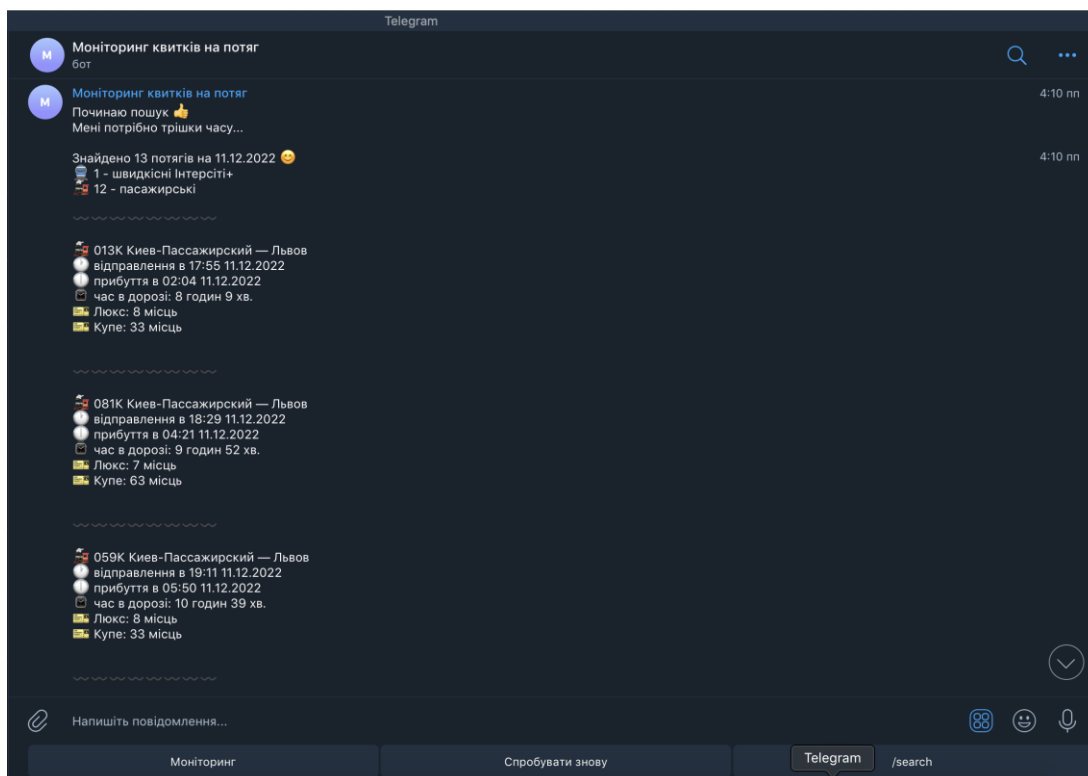


Рисунок 3.5 – Можливості моніторингу квитків ботом після надання даних для поїздки користувачем

Після створення основного меню починається реалізація меню пошуку квитків, інформаційного табла та моніторингу (рисунок 3.5). У цьому меню і відбувається вибір параметрів, завдяки яким чат-бот допомагатиме клієнту визначити параметри вибору пошуку

Важлива частина роботи це консультація по розкладу відправлень поїздів. Саме в цьому меню користувач дізнається більшу частину необхідної інформації, яка допомагає йому визначитися у своєму виборі. Вибір типу поїзду починається із вказівок необхідної дати та напрямлення(рисунок 3.5). Після вибору параметрів, які допомагають роботу отримати більш детальну інформацію, бот видає список рейсів, про які можна дізнатися більш детально. Зі списком можна взаємодіяти або повернутися назад і повторити вибір заново (рисунок 3.5).

Так як в першу чергу клієнта цікавить наявність квитків, то бот пише кількість актуальну наявність, а також клас місць.

Після клієнт може детальніше вивчити список та підібрати той, що йому більше підходить.

Дуже часто стаються випадки, коли на рейси обаного напрямку дуже великий попит, а часу і ресурсів може не вистачати на постійне оновлення даних на сайті саме через це в цьому боті ми маємо можливість поставити обраний маршрут на моніторинг. На рисунку можна побачити модуль, який відповідає за дану функцію.

```
monitoring_database.py - /Users/macbookpro/Desktop/Bot/monitoring_database.py (3.10.0)
import sqlite3

class MonitoringDatabase:
    def __init__(self):
        conn = sqlite3.connect("db_sql_main.db", check_same_thread=False)
        cursor = conn.cursor()

        cursor.execute(
            "CREATE TABLE IF NOT EXISTS monitoring_info ("
            "user_id BIGINT NOT NULL,"
            "departure_station TINYTEXT NOT NULL,"
            "arrival_station TINYTEXT NOT NULL,"
            "departure_date TINYTEXT NOT NULL"
            ");"
        )
        self.connection = conn
        self.cursor = cursor

    def append_user(self, user_id: int, data) -> None:
        departure_station = data['departure_station']
        arrival_station = data['arrival_station']
        departure_time = data['departure_date']
        self.cursor.execute(f"INSERT INTO monitoring_info"
            f"({user_id}, {departure_station}, {arrival_station}, {departure_time})"
            f"VALUES ({user_id}, '{departure_station}', '{arrival_station}', '{departure_time}');"
        )
        self.connection.commit()

    def delete_user(self, user_id: int) -> None:
        self.cursor.execute(f"DELETE FROM monitoring_info WHERE user_id={user_id};")
        self.connection.commit()

    def _user_exists(self, user_id: int) -> None:
        try:
            self.cursor.execute(f"EXISTS (SELECT user_id from users_information WHERE user_id={user_id});")
            self.connection.commit()
        except sqlite3.Error:
            pass

        return self.cursor.fetchone()

    def get_all_users(self):
        self.cursor.execute(f"SELECT user_id, departure_station, arrival_station, departure_date from monitoring_info;")
        users_data = self.cursor.fetchall()
        self.connection.commit()
        return users_data

    def exit(self):
        self.connection.commit()
        self.cursor.close()
```

Рисунок 3.6 – Програмна реалізація модуля «моніторингу»

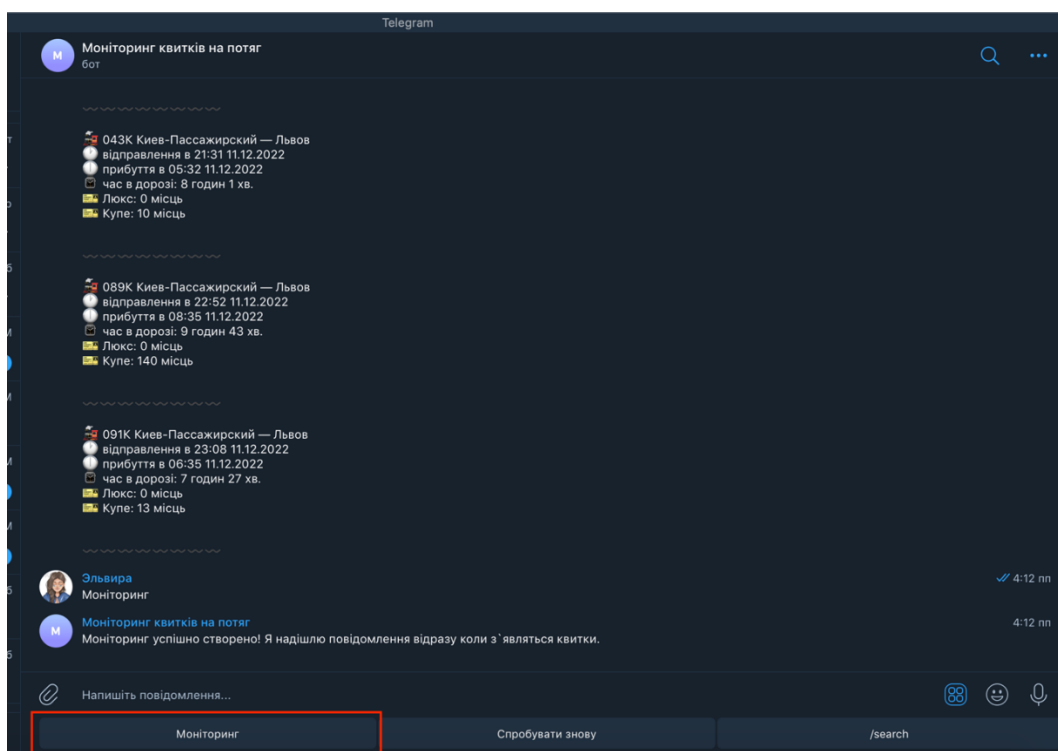


Рисунок 3.7 – Відображення кнопки «Моніторинг»

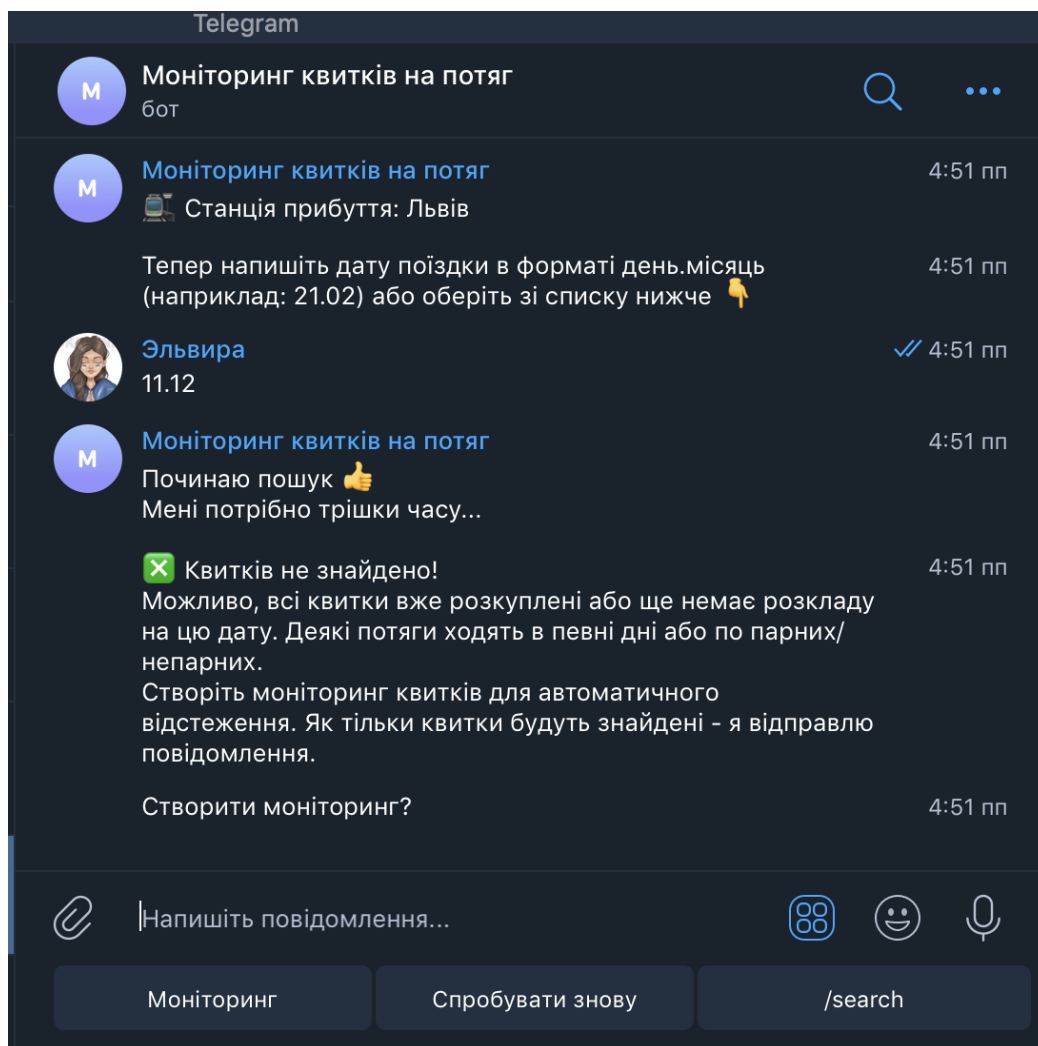


Рисунок 3.8 – Моніторинг білетів в ситуації, коли немає наявних місць

Бот відображає таку інформацію, як:

- Маршрут;
- Дату;
- Кількість наявних потягів;
- Кількість білетів;
- Час руху;

Необхідну інформацію бот бере із підключеної бази даних. Оновлення даних відбувається шляхом внесення до бази даних нових вантажних засобів чи іншої інформації.

3.3 Тестування чат-боту

Тестування створеного чат-боту проводилось особисто. На кожному етапі створення проводилось тестування на працездатність, за складеним до цього тестовими точками згідно з цілями розробки і наявними інструментами, що використовувалися

Оцінка роботи з Telegram-ботом:

- а) швидкість відповіді чат-бота на повідомлення на початку діалогу;
- б) правильна і доречна обробка запитів в клієнта за різними категоріями;
- в) цільне відображення діалогу;
- г) можливість взаємодіяти з клавішами;
- д) швидкість запитів і роботи бази даних.

Тестування боту проводилося на мобільному пристрої MacBook Pro з наступними технічними характеристиками.

- а) екран: 13,3 дюйма, IPS, 2560×1600, 227 ppi;
- б) процесор: Intel Core i5-8257U (4 ядра, 8 потоків, 1,4 ГГц, Turbo Boost до 3,9 ГГц);
- в) операційна система: macOS Sierra;
- г) оперативна пам'ять: 8;
- д) вбудована пам'ять: 128 ГБ;
- е) навігація: GPS;
- ж) Telegram: Telegram v 5.7.

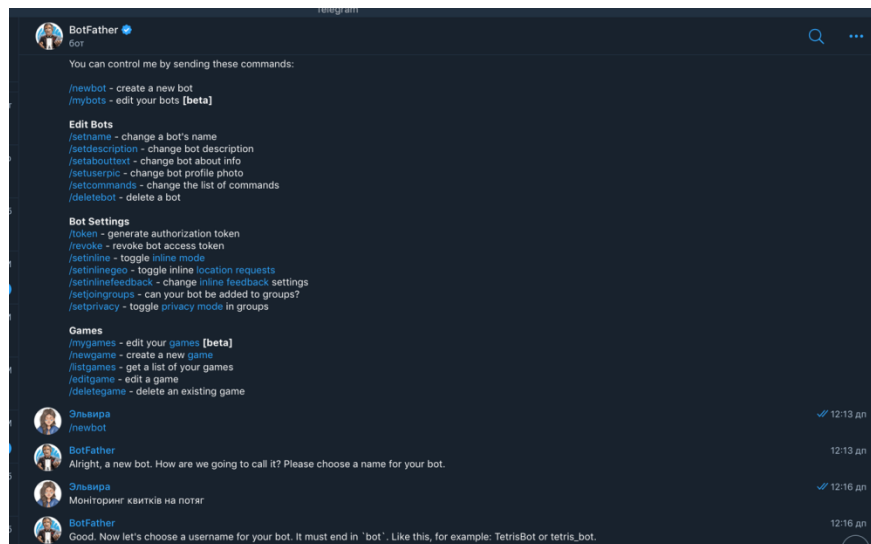


Рисунок 3.1 – Загальний вигляд кристовацького інтерфейсу розробленого чат-боту

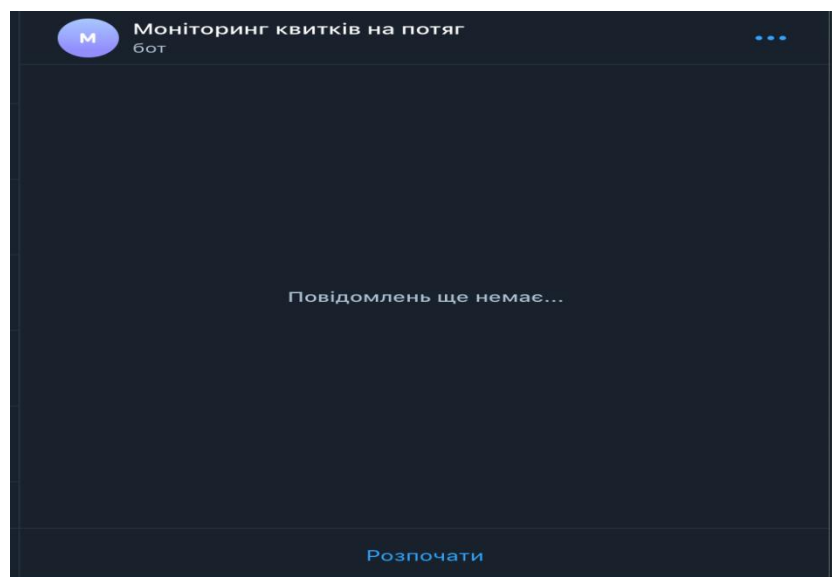


Рисунок 3.2 – Початок роботи з ботом

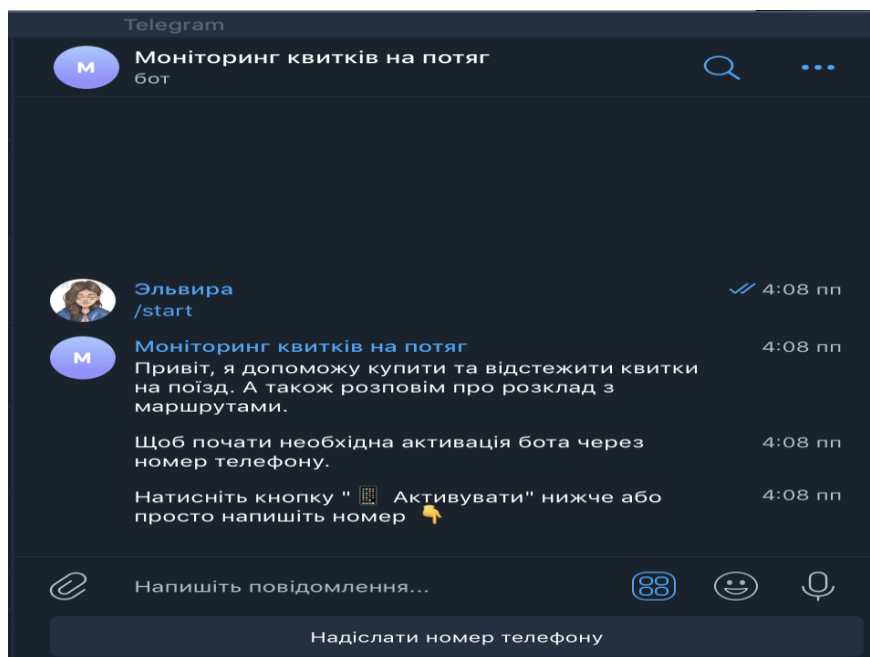


Рисунок 3.3 – Привітання та опис можливостей боту після вводу функції /start

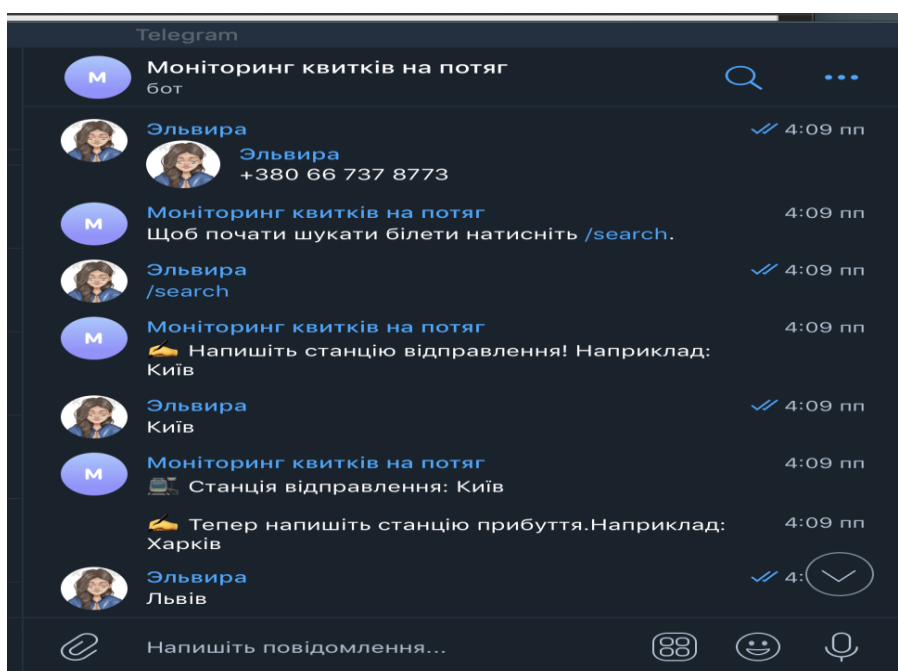


Рисунок 3.4 – Активізація функції пошуку телеграм-боту

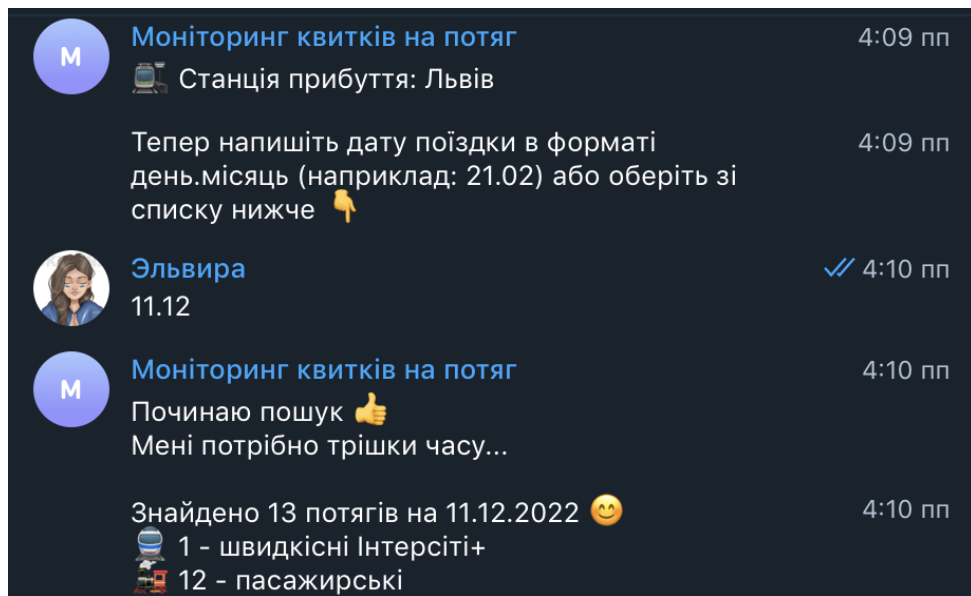


Рисунок 3.5(а) – Можливості моніторингу квитків ботом після надання даних для поїздки користувачем

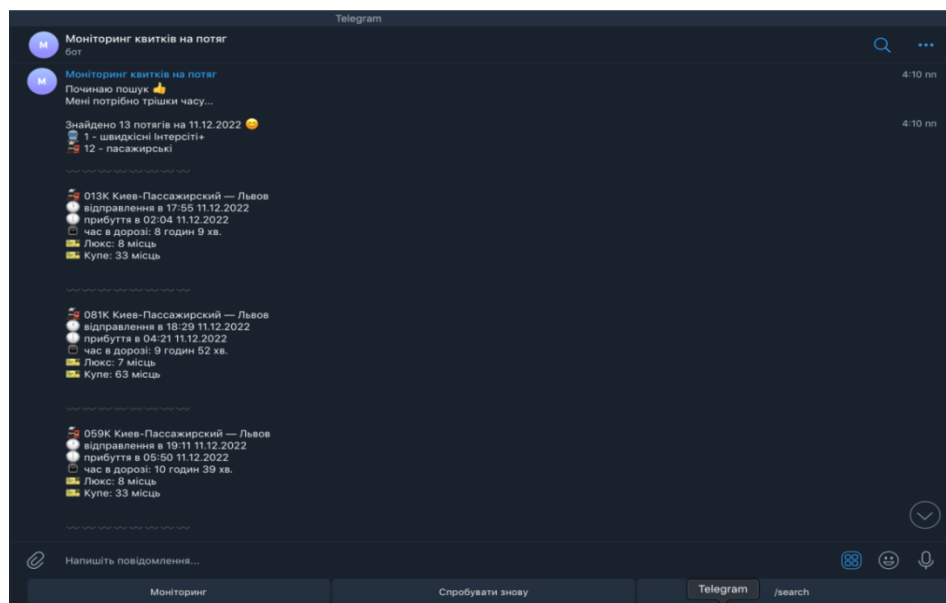


Рисунок 3.5(б) – Можливості моніторингу квитків ботом після надання даних для поїздки користувачем

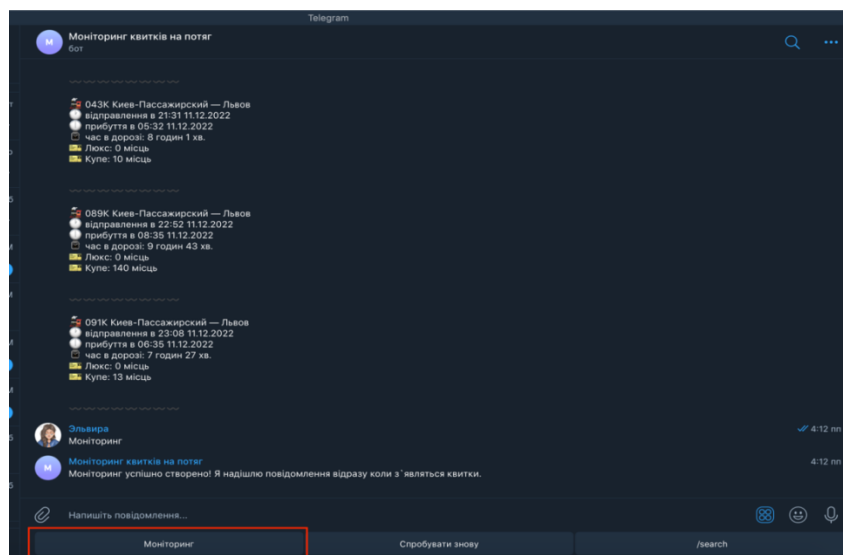


Рисунок 3.5(в) – Можливості моніторингу квитків ботом після надання даних для поїздки користувачем

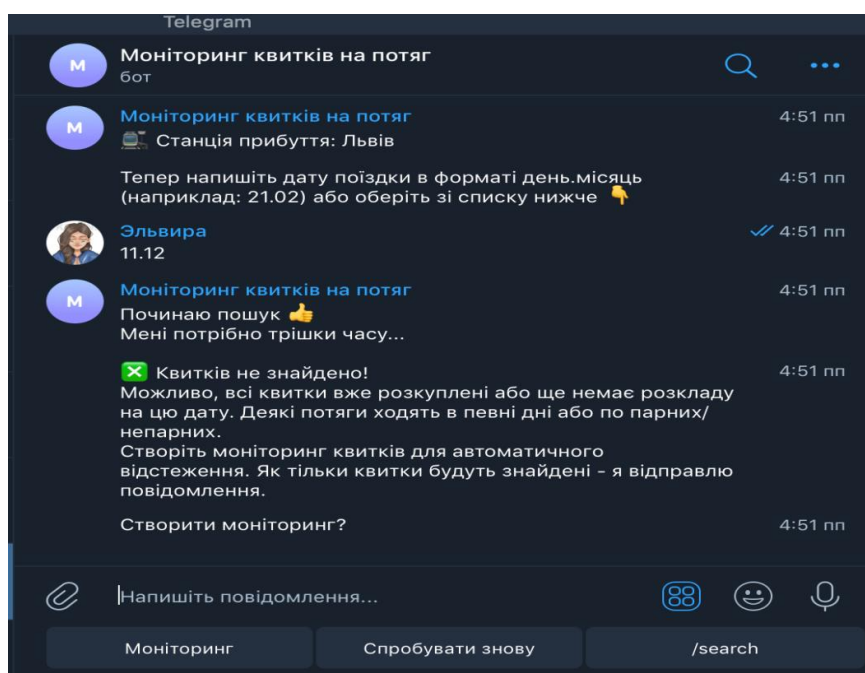


Рисунок 3.5(г) – Можливості моніторингу квитків ботом після надання даних для поїздки користувачем

Отже, розроблений бот по запиту шукає наявні квитки, показує кількість квитків по заданому маршруту і повну інформацію про кількість місць, час в дорозі і класи місць.

В разі, коли місць немає, цей бот дає можливість поставити моніторинг на заданий маршрут.

3.2. Аналіз програмної реалізації додатку

Згідно із завданням дипломного проекту та відповідно до проаналізованих підходів було створено телеграм-бота, який надає необхідну інформацію для пошуку залізничних квитків. Основні завдання, які були реалізовані в ході виконання проекту наступні:

- а) отримання запитів від клієнта;
- б) обробка запитів;
- в) за необхідністю уточнити деякі аспекти запиту;
- г) надання коректної необхідної інформації відповідно до наданого запиту;
- д) за делегів передбачена відповідь на невідповідні запити;

Також в доповнення були реалізовані наступні процеси:

- а) створення бази даних наявних поїздів по заданому маршруту;
- б) отримання і обробка точок координат місцезнаходження користувача;
- в) запропонувати варіанти маршрутів і інформаційне табло.

Для більш детального опису. Можна впевнено сказати, що користувачу буде легше дізнаватися інформацію із запропонованих ботом заготовлених запитів, після цього запит буде передан модулем по API, після завершення всіх процесів буде надана необхідна інформація.

Якщо ж клієнт ввів запит, який не був передбачений сценарієм розвитку, то бот повідомить, щоб користувач вводив запити тільки по запропонованим питанням.

Висновки за розділом 3

Підсумовуючи третій розділ, можемо зробити такі висновки:

Проведено тестування функціональних можливостей створеного telegram-боту:

1. отримання запитів від клієнта;
2. обробка запитів;
3. за необхідністю уточнити деякі аспекти запиту;
4. надання коректної необхідної інформації відповідно до наданого запиту;
5. заделегіть передбачена відповідь на невідповідні запити;

Отже, розроблений бот по запиту шукає наявні квитки, показує кількість квитків по заданому маршруту і повну інформацію про кількість місць, час в дорозі і класи місць.

В разі, коли місць немає, цей бот дає можливість поставити моніторинг на заданий маршрут.

6. Здійснено аналіз програмної реалізації додатку. Результатом проведеного дослідження стало створення телеграм-бота для моніторингу залізничних квитків. Система є сучасною та зручною, задовольняє всі вимоги з точки зору функціональності та логіки.

ВИСНОВКИ

Підсумовуючи загальний зміст дипломної роботи, можемо зробити такі висновки:

1. Визначено, що проектування та впровадження сучасних бот-систем із застосуванням передових мобільних технологій видається дуже важливим, адже розумний бот призначений для виконання одноманітної та рекурсивної роботи та, на відміну від комп'ютерних програм, на даний момент має величезні перспективи в різноманітних галузях суспільної діяльності.

2. Досліджено, що чат-бот являє собою програмний додаток, що використовується для проведення онлайн-чат розмови за допомогою тексту або тексту в мову, замість забезпечення прямого контакту з живим людським агентом. Створені для переконливого моделювання поведінки людини в якості співрозмовника, системи чат-ботів зазвичай вимагають безперервного налаштування та тестування.

3. З'ясовано, що чат-боти є одним з найбільш перспективних нововведень для впровадження в автоматизовану систему управління будь-якою установою. В тому числі використання новітніх мобільних технологій економічно обґрунтоване для процесу здійснення інтерактивного навчання з вивчення різноманітних дисциплін. такі компанії як Microsoft, Facebook, Telegram надають комплексні засоби розробки і дистрибуції програм-роботів. Пропонують спеціалізовані ресурси для вирішення конкретних задач для створення та розповсюдження платформи для проектування чат-ботів, надають різні програмні засоби, фреймворки, інструментарії, інтерфейси програмування і додаткові можливості.

4. Виявлено, що всі існуючі чат-боти можна розділити на дві великі категорії в залежності від технології, що використовується для ведення інтелектуального діалогу:

- програми з обробкою текстів на природних мовах NLP (Natural Language Processing);

- програми, засновані на формуваннях міркувань на основі конкретних випадків CBR (Cased-BasedReasoning).

5. Були проаналізовані основні існуючі аналоги чат-ботів. Також був проведений аналіз технологій та моделювання програмного забезпечення. Сервіс функціонує на сервері, який представлений платформою Node JS. Для вводу та виводу даних використовується платформа месенджера Telegram. На основі функціональних та нефункціональних вимог до ПЗ було створено функціонал, що має набір системних функцій. Кожна системна вимога має свій унікальний функціонал, до якого не мають доступу інші групи. Було здійснено зв'язок серверної частини з базою даних Mongo DB.

6. Отже, було проведено дослідження технологій для побудови телеграм-бота. В результаті дослідження були обрані наступні мови програмування та технології: Python, модуль Tele Bot, модуль Cherry Py, модуль Requests, Telegram API та база даних Mongo DB. Результатом проведеного дослідження стало створення телеграм-бота для моніторингу залізничних квитків. Система є сучасною та зручною, задовольняє всі вимоги з точки зору функціональності та логіки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Антонюк В. С., Вислоух С. П., Аверченков В. І. - Автоматизоване проектування технологічних процесів. – Київ, УМК ВО, 1989 – 116 с.
2. Антонюк В. С. Методологія наукових досліджень: [Текст] : навч. посіб./ В. С. Антонюк, Л. Г. Полонський, В.І. Аверченков, Ю.А. Малахов. – К.: НТУУ «КПІ», 2015. – 286 с.
3. Бідюк П. І. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень / П. І. Бідюк, Л. О. Коршевніук; – К.: ННК «ІПСА» НТУУ «КПІ», 2010. – 340 с.
4. Бот для Telegram на Python. Heroku сервер [Електронний ресурс]. – Режим доступу: <https://tproger.ru/translations/telegram-bot-create-and-deploy>.
5. Вентцель Е. С. Теорія ймовірностей: Учеб.для вузів. - 6-е вид. стер. - М.: Вища. шк., 2015. – С. 12-54.
6. Вороновський Г. К., Махотило К. В., Петрашев С. Н., Сергєєв С. А. Генетичні алгоритми, штучні нейронні мережі і проблеми віртуальної реальності. - Заповне. - Х.: ОСНОВА, 1997. - С. 112.
7. Вудс Р., Гонсалес Р. Цифрова обробка зображень //М.: Техносфера. - 2005. – С. 44-48.
8. Глибовець М. М. Штучний інтелект / М.М. Глибовець, О.В. Олецкий // К.: Вид. дім «КМ Академія», 2016. – 366 с.
9. Глушков В. М., Амосов М. М., Артеменко І. О. Енциклопедія кібернетики. Том 2. Київ, - 1974. – С. 33-54.
10. Головкін Б. О. Машинне розпізнавання та лінійне програмування. - М.: Радянське радіо. 2019. - 100 с.
11. Дейч А. М. Методи ідентифікації динамічних об'єктів. – М: Енергія, 1979 – 240 с.
12. Згуровський М. З. Основи системного аналізу / М. З. Згуровський, Н. Д. Панкратова. – К.: Видавнича група ВНУ, 2017. – 544 с.

13. Комашинський В. І. Смірнов Д. О. Впровадження нейроінформаційних технологій. / В. І. Комашинський, Д.А. Смірнов - СПб, 2014. – С. 33-48.
14. Круглов В. В., Борисов В. В. Штучні нейронні мережі. Теорія та практика. - 1-е. - М.: Гаряча лінія - Телекому, 2010. - С. 382.
15. Левикін В.М., Кудрявцева М.С. Візуальні мови та середовища розробки застосувань: Навч. посібник. – Харків: ХНУРЕ, 2011. – 168 с.
16. Офіційний сайт Telegam [Електронний ресурс]/ API – Режим доступу: <https://core.telegram.org/api>.
17. Поспелов Г. С Штучний інтелект - основа нової інформаційної технології - М.: Вища школа, 2019. – 280 с.
18. Роберт Каллан. Основні концепції нейронних мереж = The Essence of Neural Networks First Edition. - 1-е. - «Віл'ямс», 2017. - С. 288.
19. Рутковська Д. Нейронні мережі, генетичні алгоритми та нечіткі системи / Д. Рутковська, М. Пилинський, Л. Рутковський. - М. Гаряча лінія-Телеком 2014. – С. 34-46.
20. Самовчитель Python [Електронний ресурс]: Режим доступу: <https://pythonworld.ru/samouchitel-python>.
21. Советов Б.Я. Інформаційні технології / Б.Я. Советов, В.В Цехановський - М.: Вища школа, 2019 - 263 с.
22. Соколов В. Ю. Інформаційні системи і технології: Навчальний посібник – Київ ДУІКТ. - 2020. – С. 33-49.
23. Тархов Д. О. Нейронні мережі. Моделі та алгоритми. – М.: Радіотехніка, 2010. – 82 с.
24. Терехов В. А., Єфімов Д. В., Тюкін І. Ю. Нейромережні системи керування. - 1-е. - Вища школа, 2012. - С. 180-184.
25. Тестування телеграм-бота. - [Електронний ресурс]. – Режим доступу: [https:// habr.com/ru/post/322816](https://habr.com/ru/post/322816).

26. Технології створення та застосування чат-ботів. - [Електронний ресурс]/ Нові технології – Режим доступу: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chatbotov>.
27. Типові архітектури нейронних мереж. - [Електронний ресурс] – Режим доступу: <https://studfiles.net/preview/5740125/page:4>.
28. Уосермен Ф. Нейрокомп'ютерна техніка: Теорія і практика. Переклад українською І. Ю.Юрчак, 2016. – С. 88-94.
29. Чат-боти: історія, сучасність та перспективи. [Електронний ресурс]. – Режим доступу: <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460>.
30. Чат-боти в маркетингу та бізнесі: функції, ролі, можливості. [Електронний ресурс]. – Режим доступу: <http://www.epochta.ru/blog/articles/chat-bots/>.
31. Як створити чат-бота з нуля на Python: детальна інструкція [Електронний ресурс] – Режим доступу: <https://www.dcstyling.ru/robots/kak-sozdat-chat-bota-s-nulia-na-python-podrobnaia-instruktsiia>.
32. Ясницкий Л. Н. Введение в штучный интеллект. - 1-е. – Вид.-й центр «Академія», 2015. - С. 170-176.
33. Botan.io - експериментальний інструмент аналітики ботів Telegram від Яндекс [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/yandex/blog/264121>.
34. Chatbot and Communication Platform for Hotels. [Електронний ресурс]. – Режим доступу: <https://hijiffy.com/>.
35. Dewitt David J , Gray Jim. Parallel database systems: the future of high performance database systems. Communications of the ACM, Volume 35, Number 6, June, 1992. – P. 12-26.
36. Telegram Bot API. - [Електронний ресурс] – Режим доступу: <https://core.telegram.org/bots/api>.
37. Telegram FAQ. - [Електронний ресурс]. – Режим доступу: <https://telegram.org/faq>.

38. Telegram-bot. - [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/442800>

ДОДАТКИ

Додаток А

Лістинг коду телеграм-чату для моніторингу залізничних квитків

```
import re
import time
import asyncio
import aiogram.scheduler
from datetime import date
from aiogram import Dispatcher
from aiogram import Bot, types
from aiogram import executor

from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.dispatcher import FSMContext
from aiogram.types import ContentType

from keyboards import get_contact, monitoring_and_retry, turn_back,
    create_date_buttons
from FSM import FindTicket
from tickets_parser import TicketsParser

tickets_parser = TicketsParser()

bot = Bot(token='5969845737:AAEa38KiJLCLW9J8p1nejYOFQ6IxFautIIY')

storage = MemoryStorage()
```

```
dp = Dispatcher(bot=bot, storage=storage)
```

```

    async def monitoring_sending_data():
        users_list = tickets_parser.monitoring_database.get_all_users()
        for user in users_list:
            tickets_info = \
tickets_parser.get_tickets(departure_station_=user[1], arrival_station_=user[2],
                             departure_date=user[3])
            if tickets_info:
                user_id = user[0]
                data = tickets_info
                await bot.send_message(chat_id=user_id, text=data)
            tickets_parser.monitoring_database.delete_user(user_id=user_id)

@dp.message_handler(commands=['start'])
async def greeting(message: types.Message):
    await bot.send_message(chat_id=message.from_id, text='Привіт, я допоможу купити
        та відстежити квитки на поїзд.'
        ' А також розповім про розклад з маршрутами.')
        time.sleep(0.6)
    await bot.send_message(chat_id=message.from_id, text='Щоб почати необхідна
        активація бота через номер телефону.')
        time.sleep(0.6)
    await bot.send_message(chat_id=message.from_id,
text='Натисніть кнопку "  Активувати" нижче або просто напишіть номер  ',
        reply_markup=get_contact)

```

```

@dp.message_handler(content_types=ContentType.CONTACT)
    async def contact_getting(message: types.Message):
awaitbot.send_message(chat_id=message.from_id, text='Щоб почати шукати білети
    натисніть /search.',
    reply_markup=turn_back)

```

```

@dp.message_handler(commands=['search'])
    async def searching(message: types.Message, state: FSMContext):
awaitbot.send_message(chat_id=message.from_id, text='🔍 Напишіть станцію
    відправлення! Наприклад: Київ')
    await state.set_state(FindTicket.departure)

```

```

@dp.message_handler(state=FindTicket.departure)
    async def set_departure_station(message: types.Message, state: FSMContext):
        departure_station = message.text
        station_encoding = tickets_parser.encode_station(departure_station, 'departure')
        if station_encoding:
            async with state.proxy() as state_data:
                state_data['departure_station'] = station_encoding
            awaitbot.send_message(chat_id=message.from_id, text=f'📍 Станція відправлення:
                {departure_station}')
            awaitbot.send_message(chat_id=message.from_id,
                text=f'🔍 Тепер напишіть станцію прибуття. Наприклад: Харків',
                reply_markup=turn_back)
            await FindTicket.next()
        else:
            awaitbot.send_message(chat_id=message.from_id,

```

```

text='Станцію не знайдено, напишіть назву коректно, будь-ласка!')

@dp.message_handler(state=FindTicket.arrival)
async def set_arrival_station(message: types.Message, state: FSMContext):
    arrival_station = message.text
    station_encoding = tickets_parser.encode_station(arrival_station, 'arrival')
    if station_encoding:
        async with state.proxy() as state_data:
            state_data['arrival_station'] = station_encoding
    await bot.send_message(chat_id=message.from_id, text=f'📍 Станція прибуття:
    {arrival_station}')
    await bot.send_message(chat_id=message.from_id, text='Тепер напишіть дату поїздки
    в форматі день.місяць'
    ' (наприклад: 21.02) або оберіть зі списку нижче 📅',
    reply_markup=create_date_buttons())
    await FindTicket.next()
    else:
        await bot.send_message(chat_id=message.from_id,
        text='Станцію не знайдено, напишіть назву коректно, будь-ласка!')

@dp.message_handler(state=FindTicket.departure_date)
async def set_departure_time(message: types.Message, state: FSMContext):
    # TODO add date validity checking
    departure_date = message.text
    if bool(re.match(r"\d{2}.\d{2}", departure_date)):
        departure_date = message.text + '.' + str(date.today()).split('-')[0]
    async with state.proxy() as state_data:
        # TODO departure date correction

```

```

state_data['departure_date'] = departure_date
awaitbot.send_message(chat_id=message.from_id, text='Починаю пошук 🐾\nМені
    потрібно трішки часу...')
    tickets_info = tickets_parser.get_tickets(
departure_station_=state_data['departure_station'],
    arrival_station_=state_data['arrival_station'],
    departure_date=departure_date)
    if tickets_info:
awaitbot.send_message(chat_id=message.from_id, text=tickets_info,
    reply_markup=monitoring_and_retry)
    else:
        awaitbot.send_message(
            chat_id=message.from_id,
            text='❌ Квитків не знайдено!\n'
'Mожливо, всі квитки вже розкуплені або ще немає розкладу на цю дату.'
' Деякі потяги ходять в певні дні або по парних/непарних.\n'
'Створіть моніторинг квитків для автоматичного відстеження.'
' Як тільки квитки будуть знайдені - я відправлю повідомлення.',
            reply_markup=monitoring_and_retry)
awaitbot.send_message(chat_id=message.from_id, text='Створити моніторинг?', )
        await FindTicket.next()
    else:
awaitbot.send_message(chat_id=message.from_id, text='Оберіть, будь-ласка,
    коректну дату!')

```

```

@dp.message_handler(state=FindTicket.monitoring)
async def monitoring(message: types.Message, state: FSMContext):
    if message.text == 'Моніторинг':
        async with state.proxy() as state_data:

```

```
tickets_parser.monitoring_database.append_user(user_id=message.from_id,  
                                                data=state_data)  
awaitbot.send_message(  
    chat_id=message.from_id,  
    text='Моніторинг успішно створено! Я надішлю повідомлення відразу коли  
        з`являться квитки.')
```