МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка програми віддаленого управління комп'ютером

Виконав студент 4 курсу групи К-42
Напрям 6.050101 комп'ютерні науки,
Зубрицький Богдан Миколайович

Керівник ст.викладач
Вохменцева Тетяна Борисівна

Консультант д.т.н., професор
Мещеряков Володимир Іванович

Рецензент к.ф.–м.н., доцент
Ткач Тетяна Борисівна

Одеса 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук управління та адміністрування
Кафедра Інформаційних технологій
Рівень вищої освіти бакалавр
Напрям підготовки 6.050101 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри_____
С.Д. Кузніченко _____
"3" квітня 2019 р.

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Зубрицькому Богдану Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програми віддаленого управління комп'ютером

керівник роботи ст.викл. Вохменцева Тетяна Борисівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від "01 "квітня 2019 р. № 49-с
2. Строк подання студентом роботи 28.05.2019
3. Вихідні дані до роботи Ресурси мережі Інтернет

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз предметної області
Вибір та обгрунтування засобів реалізації
Розробка алгоритмів
Опис роботи програми

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Аналіз предметної області	Мещеряков В.І., д.т.н., проф.	20.04	20.04

7. Дата видачі завдання "18" квітня 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Аналіз предментної області. Постановка завдання	24.04		
2	Вибір обґрунтування засобів реалізації	26.04		
3	Розробка алгоритмів	03.05		
4	Створення програми	13.05		
5	Рубіжна атестація	13.05 – 19.05		
6	Тестування	20.05		
7	Оформлення пояснювальної записки	22.05		
8	Перевірка на плагіат	23.05		
9	Оформлення презентації	25.05		
10	Рецензування, нормоконтроль	28.05		

Студент _____ Зубрицький Б.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Рольщиков В.Б.
(підпис) (прізвище та ініціали)

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI (Artificial intelligence) – штучний інтелект.

API (Application Programming Interface) – програмний інтерфейс програми.

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту.

HTTPS (HyperText Transfer Protocol Secure) – захищений протокол передачі гіпертексту.

IP (Internet Protocol) – міжмережевий протокол.

JSON (JavaScript Object Notation) – позначення об'єкта JavaScript.

REST (Representational State Transfer) – передача стану уявлення.

TCP (Transmission Control Protocol) – протокол управління передачею.

ВСТУП

Чат-боти аж ніяк не новина: перший хто їх створив був Джозеф Вейценбаум [1][1)] аж у 1966 році. Він називався «Еліза» [2][2)] і розмовляв, як психотерапевт: перепитував і просив продовження. Деякі при першому контакті навіть не здогадувалися, що їх співрозмовник – машина.

Дуже довго чат-боти тільки імітували філософські розмови: довго підтримували розмову, але без практичного сенсу. Але ось коли шаблони з спілкування змінили на рекомендації, сталася справжня революція. Тепер вони не просто відповідали «Як настрій», а й могли підказати, чи варто взяти парасольку. Найбільш просунутих навіть навчили сумніватися і перенаправляти питання людині. З тих пір чат-боти не імітують спілкування, а пророкують найкращу відповідь. Так навіщо наймати зайвих співробітників, якщо програма правильно розуміє співрозмовника і здатна до виразного спілкування? Все одно живі працівники діють по завченим інструкціям, але їм при цьому треба платити і давати лікарняні з вихідними.

Розробити чат-бота – недорого, а після цього він зможе без зупинок допомагати людям.

Чат-боти – це віртуальні співрозмовники, програми, що імітують живу людину. В основу роботи покладено алгоритм штучного інтелекту. Взаємодія зазвичай проходить через інтернет-чат. Найбільш часто зустрічаються електронні менеджери, які консультують користувача при відвідуванні сайту.

Однак, функції чат-ботів постійно розширюються. Зараз такі програми вже нагадують особистих секретарів, в коло їхніх обов'язків входять інформування про погоду, про знижки, замовлення квитків і багато іншого. Існує думка, що чат-боти незабаром витіснять соціальні мережі і замінять цілий ряд сайтів. На їхньому боці зручність і ефективність.

[1)] [1] Джозеф Вейценбаум стаття з «Вікіпедії» вільної енциклопедії. URL: https://is.gd/ZicOUC. (дата звернення 12.06.19).

[2)] [2] «Еліза» стаття з «Вікіпедії» вільної енциклопедії. URL: https://is.gd/Slym7h. (дата звернення 12.06.19).

У список можливостей входять консультування з питань: медицини; закону і права; страхування; покупок і продажів; інвестування та інших областей. Фактично сучасний бот здатний відповісти на будь-яке питання, досить лише завантажити в нього інформацію.

Така программа працює швидко і майже завжди безпомилково, повністю замінюючи людину. Однією з популярних функцій є органайзер, коли чат-бот самостійно складає розклад, аналізуючи отриманий код і надаючи кілька варіантів розподілу часу.

Це дуже зручно при щільному графіку. Фактично бот виконує роль секретаря, допомагаючи економити час і гроші. Чат-боти зайняли місце помічників в пошукових системах і меседжерах. Так «Яндекс» [3][1)] і Телеграм [4][2)] давно активно пропонують своїм користувачам допоміжні програми. У коло їхніх обов'язків входять як простий пошук, так і аналітика. У секторі розважальних програм чат-боти використовуються як співрозмовник або навчальна програма.

В онлайн іграх застосовуються боти, які автоматизують процесс гри, коли фактично за юзера грає комп'ютер. Гравець лише віддає основні команди, що значно економить час. Більш серйозне застосування – це біржові чат-боти. Їх завдання відстежувати і аналізувати всю інформацію по торговому майданчику та давати рекомендації брокеру. Звичайно, такий бот не замінить людини, але значно полегшить його працю. Частка яку виконують боти досягає 30%, що не мало.

Сучасні можливості чат-ботів постійно удосконалюються і розширюються. Зараз їх можна зустріти і в соціальних мережах і в серйозному бізнесі. Штучний інтелект не помиляється, при цьому значно економить час. У ці технології зараз вкладаються великі гроші, і коло поширення дуже широке.

[1)] [3] Система пошуку «Яндекс». URL: https://yandex.ua/ (дата звернення 12.06.19).

[2)] [4] Месенджер, що дозволяє обмінюватися повідомленнями і медіафайлами багатьох форматів. URL: https://telegram.org/ (дата звернення 12.06.19).

Ціль моєї роботи – створення програми для управління комп’ютером та програмами, до яких буде надано доступ, для полегшення користування комп’ютером на великих відстанях та за неможливістю особистої присутності.

Обсяг пояснювальної записки становить 52 сторінок, також присутні 2 таблиці та 18 рисунків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Чат-боти з'явилися вже давно, проте широкого вжитку набули відносно недавно. Коло застосування таких ботів дуже широке і великі компанії в них зацікавлені, і в технологію будуть вкладатися великі гроші. Тому досвідчені програмісти вже зараз досліджують цю нішу, оскільки база штучного інтелекту стрімко вдосконалюється, і подібні розробки добре вписуються в концепцію роботизації людства в майбутньому.

1.1 Види і функції чат-ботів

Виділяють кілька видів чат-ботів: консультанти – від простих менеджерів в інтернет-магазинах до медичних і юридичних послуг. Їх обов'язки – спілкуватися з потенційним і реальним клієнтом, відповідаючи на всілякі питання. Помічники – можна зустріти в месенджерах і пошукових системах. Допомагають при пошуку інформації, також проводять її первинний аналіз. Розваги – часто віртуальний співрозмовник, який відповідає на питання. Програми навчаються на залишених повідомленнях, і з часом можуть замінити реального співрозмовника.

Соціальні мережі гідно оцінили можливості чат-ботів. Відомо, що в популярній мережі майже половину акаунтів займають боти і віртуали. У соц. мережах чат-боти часто несуть корисні функції: інформація про курс валют, останні новини, замовлення квитків, будь-якого товару, складання гороскопу, переклад слова. Часто ефективно замінюють собою мобільні додатки, оскільки легше встановлюються і коштують дешевше.

При правильному підході при відсутності фантазії чат-бот допоможе знайти нових друзів, взявши на себе функцію парламентера. Зараз вже є дуже «розумні» боти, здатні бути максимально схожими на людину.

Ще один бостонський стартап – «AdmitHub» [5][1]. Цей чат-бот надає допомогу абітурієнтам при вступі до університетів, а потім – зі складнощами університетського життя.

Бот «Oli» [6][2] – це посібник з життя в коледжі, що надається студентам безкоштовно. Стандартні питання, на які він відповість – коли закінчиться реєстрація на курс, де найближчий гуртожиток або їдальня. З його допомогою адміністрація залучає тільки що вступивших в навчальний процес і описує перспективи для них.

Чат-ботів впровадила також відома мовна платформа «Дуолінго» [7][3], призначена для спілкування з іноземцями та навчання їх мов. Автоматичне ПО спілкується з користувачем на мові, що вивчається, причому воно першим починає розмову: запитує про життя, наприклад. Якщо ви не знаєте, що відповісти, вони підкажуть варіанти, а потім виправлять помилки. На початкових стадіях спілкування дуже просте і під силу новачкам. У міру практики заняття ускладнюються. Більш того, кожен бот володіє власним стилем спілкування і своїми варіантами відповідей на ваші репліки.

Куди ви лізете в першу чергу, якщо щось заболіло? В Інтернет, дивитися – до чого симптом. Знаючи про це, в китайському пошукачі запустили медичного чат-бота для діагностики. І якщо людина сама собі за інформацією з мережі поставить купу діагнозів один страшніший за інший, то Melody (імя чат-бота) зв’язується з лікарем і записує вас на прийом. Бот аналізує симптоми і обчислює, до якого з шестисот тисяч реальних докторів відправити пацієнта. Він задає питання, пропонує варіанти відповіді – дуже прискіпливо. Таким чином, у медиків ще до візиту збирається необхідна інформація. Ця програма діє лише в КНР, але його автори збираються

---

[1] [5] Компанія, яка використовує штучний інтелект і мобільні повідомлення для допомоги студентам у коледжі. URL: https://www.admithub.com/ (дата звернення 12.06.19).

[2] [6] Чат-бот компанії «AdmitHub» для допомоги студентам. URL: http://www.olibot.com/ (дата звернення 12.06.19).

[3] [7] Дуолинго (англ. Duolingo) - безкоштовна платформа для вивчення мови та краудсорсингових перекладів. URL: https://ru.duolingo.com/ (дата звернення 12.06.19).

розширюватися. А в США і Англії вже запускають такі ж автоматизовані системи.

Your.MD збирає інформацію про симптоми і анамнез, після чого видає ймовірні причини стану звернувся і ставить оцінки, наскільки це може бути серйозно.

З'явилася навіть віртуальна медсестра Моллі – її розробили в Сан-Франциско. Вона розпізнає жести і мову, аналізуючи з їх допомогою стан. Вона допоможе стежити за перебігом хронічного захворювання, скласти індивідуальний план лікування, нагадає про прийом медикаментів – зовсім як медсестра, яка за вами наглядає. Тим часом за даними ВООЗ в найближчі два десятиліття в світі не вистачатиме близько 13 мільйонів медпрацівників – чим не вирішення проблеми?

Дух старої школи ще не вимер в сучасних чат-ботах, хоча вони все більше схожі на консультантів. Наприклад, це «AnzacLiveWizeline» [8][1)] – величезний проект, оцифровуючий і перероблючий щоденникові записи австралійського військового Арчі Барвика. Це ветеран Першої Світової, який вже давно загинув, але інформація подається так, ніби він на фронті прямо зараз.

Його щоденники зберегли історію того далекого періоду. У записах – всі думки і почуття юного солдата: тут і випивка, і спорт, і їжа, і спогади про будинок, і страх смерті в насуваючийся битві.

Таким чином у Арчі з'явилося друге життя. Проект став синтезом відразу декількох напрямів: традиційної друкованої журналістики, соціальних каналів і історіографічних досліджень. Додаток до цього дня відповідає на питання і відправляє меседжі з французького поля битви, де війна припинилася більше сотні років тому.

[1)] [8] Сайт компанії «AnzacLiveWizeline».
URL: http://www.anzaclive.com.au/feature.html (дата звернення 12.06.19).

Ботів просто навчити продажу товарів, але з послугами більше нюансів. Страховики зазвичай продають потрібний вид послуг, а не пропонують вибрати. Цей процес складно піддається автоматизації.

Одна з компаній-страховиків представила свого бота «A.I. Jim», який займається обробкою претензії. З його допомогою поставили рекорд з оперативності реагування: він обробив заяву про вкрадене пальто за три секунди.

За цей термін банк отримав платіжні інструкції, а клієнт – повідомлення про успіх розгляду справи. Причому розробники взагалі не страховики: два технічних працівника звернули увагу, що в галузі непочатий край роботи, яку могли б виконувати автоматичні програми.

Бот може обробити тільки прості випадки, але справляється з цим за секунди замість днів і тижнів.

Це не перше подібне технічне рішення в страхуванні. Додаток Spixii на п'яти мовах пропонує страхування, причому якщо помітить, що ваш пакет не покриває все, чим ви займаєтеся (наприклад, ви любитель екстриму), то запропонує розширений – дорожчий. Зовсім як людина.

Нові технології швидко розвиваються, штучний інтелект впроваджується в наше життя. Роботи навчилися повноцінно спілкуватися з користувачами. Чат-бот – віртуальний помічник, який вбудовується в месенджери і допомагає підприємцям стати ближче до клієнтів. Такий робот – це автоматизована система спілкування з користувачами, що дозволяє ефективно вирішувати проблеми і скорочує витрати бізнесменів за рахунок зменшення числа менеджерів і працівників call-центрів.

Бізнес чат-боти, завдання яких – оптимізувати роботу, зробивши її більш ефективною, відмінно зарекомендували себе для збору статистичних даних про ту чи іншу онлайн-компанії. Не рідко використовуються як звичайний щоденник. Функція чат-ботів лежить в площині оператив нихпослуг, коли треба швидко і в будь-який момент відповісти на питання або зробити аналіз ситуації. Чат-боти користуються популярністю у брокерів

і трейдерів, коли потрібно дізнатися стан біржі. Поширені види бізнес чат-ботів: тижневик з функцією органайзера, особистий секретар, лічильник обміну валют з постійним оновленням, програми аналізу стану фінансового ринку.

Особливо перспективним бачиться майбутнє чат-ботів як навчальних програм для дітей. Вони не тільки здатні відповідати на багато питань, але і надавати навчальний матеріал в ігровому виді. Також цікаві чат-боти в вигляді електронних енциклопедій. Програма знаходить за ключовим словом необхідну інформацію і коментує її. Чат-боти популярні в розважальному секторі. Віртуальний співрозмовник не тільки відповідає на питання, але і навчається в процесі.

Чат-боти допомагають оптимізувати роботу і збільшити ККД. Завдяки чат-ботів досягається автоматизація рутинних процесів, за рахунок чого економиться час і гроші. Також цезручно і для клієнта, який отримує інформацію безпосередньо. Інша важлива функція чату ботів в бізнесі – це аналітика. Такі програми на фінансовому ринку користуються великою популярністю за рахунок швидкості реагування.

Крім підприємців, віртуальні помічники вигідні й клієнтам:

- економія часу – людям відповідають миттєво, а не на наступний день;
- людина може користуватися сервісом на тих майданчиках, до яких вона звикла;
- користувачі отримують тільки ту інформацію, яка їм цікава - ніхто не нав'язує щось купити.

1.3 Постановка задачі

При пошуку можливих рішень для написання чат-бота було встановлено, що для розробки бота необхідно знати хоча б одну з мов серверного програмування: Python, Ruby, Node.JS, PHP. Потрібно було

визначитися яку саме мову вивчати і використовувати. Так само важливо вміти працювати з REST (Representational State Transfer) [9][1] API(Application Programming Interface) [10][2], який надають месенджери, а саме Telegram BotAPI.

Після визначитися з напрямком, оскільки боти бувають двох типів:

– Ті, що навчаються (розуміють природні мови), які використовують логіку при побудові діалогу або обробку природної мови (NLP) і машинне навчання (ML) для формування відповідей на повідомлення. Або і те, і інше.

– Скриптові боти (не розуміють природні мови), в них весь діалог – це заздалегідь сформований шаблон, а «скрипт» – це дерево рішень, в якому відповідь на питання відкриває новий, заздалегідь запрограмований сценарій. Діалоги в них зазвичай лінійні і структуровані.

І, в кінцевому рахунку, для чого його використовувати, так як у бота повинна бути певна задача, тому що інакше він не представляє ніякої цінності.

В якості мови для написання сервера краще обрати Python по ряду причин.

---

[1] [9] Representational State Transfer стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/REST (дата звернення 12.06.19).

[2] [10] Application Programming Interface стаття з «Вікіпедії» вільної енциклопедії. URL https://ru.wikipedia.org/wiki/API (дата звернення 12.06.19).

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ

2.1 Вибір мови програмування

Python це мова програмування загального призначення, націлена в першу чергу на підвищення продуктивності самого програміста, ніж коду, який він пише. Говорячи простою мовою, на Python можна написати практично що завгодно (веб / настільні додатки, ігри, скрипти по автоматизації, комплексні системи розрахунку, системи управління життєзабезпеченням і багато іншого) без відчутних проблем. Більш того, поріг входження низький, а код багато в чому лаконічний і зрозумілий навіть для того, хто ніколи на ньому не писав. За рахунок простоти коду, подальший супровід програм, написаних на Python, стає легше і приємніше. А з точки зору бізнесу це тягне за собою скорочення витрат і збільшення продуктивності праці співробітників .

Безсумнівною перевагою є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах. Першою такою мовою була C, однак її типи даних на різних машинах могли займати різну кількість пам'яті і це служило деякою перешкодою при написанні дійсно переносної програми. Python не має такого недоліку.

Так само, важлива риса – розширюваність мови, цьому надається велике значення і, як пише сам автор Гвідо ван Россум, мова була задумана саме як розширювана. Це означає, що є можливість вдосконалення мови усіма зацікавленими програмістами. Інтерпретатор написаний на C і вихідний код доступний для будь-яких маніпуляцій. У разі необхідності, можна вставити його в свою програму і використовувати як вбудовану оболочку. Або ж, написавши на C свої доповнення до Python і скомпілювавши програму, отримати "Розширений" інтерпретатор з новими можливостями.

Наступна перевага – наявність великого числа підключаємих до програми модулів, що забезпечують різні додаткові можливості. Такі модулі пишуться на C і на самому Python і можуть бути розроблені усіма досить кваліфікованими програмістами. Як приклад можна привести такі модулі:

– Numerical Python – розширені математичні можливості, такі як маніпуляції з цілими векторами і матрицями;
– Tkinter – побудова додатків з використанням графічного призначеного для користувача інтерфейсу (GUI) на основі широко розповсюдженого на X-WindowsTk-інтерфейсу;
– OpenGL – використання великої бібліотеки графічного моделювання дво- і тривимірних об'єктів OpenGraphicsLibrary фірми SiliconGraphicsInc[11][1]. Даний стандарт підтримується, в тому числі, в таких поширених операційних системах як Microsoft Windows 95 OSR 2,98[12][2] і WindowsNT 4.0[13][3].

Єдиним недоліком, поміченим автором мови, є порівняно невисока швидкість виконання Python-програми, що обумовлено її інтерпретованістю. Однак, це з надлишком окупається перевагами мови при написанні програм не дуже вимогливих до швидкості виконання.

З використаних модулів можна виділити: TeleBot, CherryPy, Requests.

2.2 Вибір модуля для спрощення написання коду

Цей модуль є оболочкою над запитами до Telegram Bot API, використовується для спрощення та мінімізації написаного коду.

---

[1] [11] SiliconGraphicsInc стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Silicon_Graphics (дата звернення 12.06.19).

[2] [12] Microsoft Windows 95 OSR 2,98 стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Windows_95 (дата звернення 12.06.19).

[3] [13] WindowsNT 4.0 стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Windows_NT_4.0 (дата звернення 12.06.19).

Всі типи визначені в types.py[14] [4]. Всі вони повністю відповідають визначенню типів API Telegram, за винятком from поля Message, яке перейменовано в from_user (оскільки from це зарезервований токен Python). Таким чином, до таких атрибутів як message_id можна звертатися безпосередньо, наприклад: message.message_id. Варто звернути увагу, що атрибут message.chat може належати як певному користувачеві, так і групового чату.

Об'єкт Message також має content_type атрибут, який визначає тип повідомлення. Атрибут content_type може бути одним з наступних рядків: text, audio, document, photo, sticker, video і так далі. Можна використовувати декілька типів в одній функції.

Всі методи API розташовані в класі TeleBot. Вони перейменовані, щоб слідувати загальним угодами про імена Python. Наприклад: sendMessage перейменований в send_message, editMessageText в edit_message_text.

Оброблювач повідомлень – це функція, прикрашена декоратором примірника TeleBot. Обробники повідомлень складаються з одного або декількох фільтрів. Кожен фільтр повертає True або False для певного повідомлення, і якщо повертається True, то обробник отримує дозвіл на обробку повідомлення. Приклад оголошення обробника повідомлень приведено нижче:

```
@bot.message_handler(filters)
def function_name(message):
bot.reply_to(message, "This is a message handler")
```

Ім'я функції не має ніяких обмежень, крім тих, що встановлені мовою. Функція повинна приймати не більше одного аргументу, яким є повідомлення, яке повинна обробляти функція. Поле filters – список аргументів, ключових слів. Фільтр оголошуються в такий спосіб:

[4] [14] Веб-сторінка визначень типів мови Python. URL: https://svn.python.org/projects/stackless/trunk/Lib/types.py (дата звернення 12.06.19).

ім'я = аргумент. Один обробник може мати кілька фільтрів. Підтримувані TeleBot фільтри можна побачити в табл. 1.

Таблиця 1 – Підтримувані фільтри

Ім’я	Аргументи	Опис
content_types	Список допустимих типів (за замовчуванням [«’text’])	Повертає True, якщо message.content_types знаходитьс я в списку.
Regexp	Регулярний вираз у вигляді рядка	Повертає True, якщо функція re.search (regexp) повертає True і message.content_type =’text’ .
Commands	Список допустимих команд	Повертає True, якщо message.content_type == 'text' і message.text починається з команди, яка знаходиться в списку.
Func	Функція (lambda або довільна функція)	Повертає True, якщо lambda або функція повертає True.

Важливо розуміти що, всі обробники перевіряються в тому порядку, в якому вони були оголошені. Нижче наведено кілька прикладів використання фільтрів і обробників повідомлень.

#Обробляє всі текстові повідомлення, що містять команди

```
'/ start' или '/ help'.
@ bot.message_handler ( commands = [ ' start ' , ' help ' ])
def handle_start_help ( message ):
    pass
```

Обробляє всі відправлені документи і аудіофайли

```
@ bot.message_handler ( content_types = [ ' document ' ,
' audio ' ])
def handle_docs_audio ( message ):
pass
```

Обробляє всі текстові повідомлення, які відповідають регулярному виразу

```
@ bot.message_handler ( regexp = " SOME_REGEXP " )
def handle_message ( message ):
pass
```

Обробляє всі повідомлення, для яких лямбда повертає True

```
@ bot.message_handler ( func = lambda message : mes-
sage.document.mime_type == ' text / plain ' ,
content_types = [ ' document ' ])
def handle_text_doc ( message ):
pass
```

В оновленні bot2.0 з'явилися так звані запит зворотного дзвінка вбудовуються в inline-кнопки. Для обробки таких об'єктів використовується спеціальний декоратор, приведений нижче.

```
@bot.callback_query_handler(func=lambda call: True)
def test_callback(call):
logger.info(call)
```

2.3 Вибір модуля для розробки веб-додатків

CherryPy – це об'єктно-орієнтований веб-фреймворк, написаний на мові програмування Python. Спроектований для швидкої розробки веб-додатків

для мережі Інтернет. Являє собою надбудову над HTTP-протоколом, але залишається на низькому рівні і не виходить за рамки вимог RFC 2616[15][1)].

Встановити CherryPy зовсім нескладно, досить завантажити його і виконати кілька коротких інструкцій з файлу README. Модуль CherryPy встановлюється точно так же, як і будь-який інший модуль Python, тому в даному випадку не доводиться говорити про каталог установки. На відміну від інших, більш складних серверних технологій, модуль CherryPy просто стає доступним для використання щоразу, як тільки буде імпортований за допомогою інструкції import, як і будь-який інший модуль Python. Насправді CherryPy – це не більше ніж самостійний додаток на мові Python, який управляє своїм власним багатопотоковим веб-сервером, завдяки цьому виконати «сценарій на стороні сервера» так само просто, як запустити просту команду у вікні терміналу.

За замовчуванням сервер запускається локально на 8080 порту, і виконує передану йому функцію, але так само можна задати деякі параметри вручну, такі як: IP-адреса, порт, відкритий і закритий ключі SSL. Приклад завдання параметрів і запуск веб-сервера приведений нижче:

```
bot.remove_webhook()
bot.set_webhook(url=WEBHOOK_URL_BASE + WEBHOOK_URL_PATH, certif-
icate=open(WEBHOOK_SSL_CERT, 'r'))
cherrypy.config.update({
'server.socket_host': WEBHOOK_LISTEN,
'server.socket_port': WEBHOOK_PORT,
'server.ssl_module': 'builtin',
'server.ssl_certificate': WEBHOOK_SSL_CERT,
'server.ssl_private_key': WEBHOOK_SSL_PRIV
})
cherrypy.quickstart(WebhookServer(), WEBHOOK_URL_PATH,
{'/': {}})
```

[1)] [15] RFC 2616 – Протокол Передачи Гипертекста – HTTP/1.1 URL: https://rfc2.ru/2068.rfc (дата звернення 12.06.19).

Наведений приклад використовується для того, щоб після установки вебхуків «слухати» коли прийде повідомлення від сервера Telegram.

2.4 Вибір модулю для виконня HTTP-запитів

Requests – бібліотека Python, яка елегантно і просто виконує HTTP-запити (HyperText Transfer Protocol)[16][1)]. Нижче приведений приклад отримання інформації про веб-сторінку через запит import requests

```
resp = requests.get('htthps://api.github.com/events')
```

Після виконання цього коду, в об'єкті resp зберігається вся необхідна інформація про цей об'єкт.

Простий API Requests означає, що всі форми HTTP запитів є очевидними. Наприклад, нижче приведено, як можна зробити HTTP POST запит.

```
resp = requests.post("http://httpbin.org/post")
```

Інші HTTP-запити (PUT, DELETE, HEAD, OPTIONS) робляться за аналогією.

2.5 Отримання інформації від користувача

У документації Telegram Bot API виділяється два діаметрально протилежних способу отримання оновлень: с допомогою періодичних

[1)] [16] HyperText Transfer Protocol стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/HTTP (дата звернення 12.06.19).

запитів або установка вебхуків. Вхідні оновлення зберігається до тих пір, поки сервер не обробить його, але не довше 24 годин. Незалежно від способу отримання оновлень, у відповідь отримуємо об'єкт Update, серіалізований в JSON[17][2)].

Об'єкт Update представляє з себе що входить оновлення. Під оновленням мається на увазі дію, вчинене з ботом – наприклад, отримання повідомлення від користувача.

Тільки один з необов'язкових параметрів предтавлених на рис. 1 може бути присутнім в кожному оновленні.

Поле	Тип	Описание
update_id	Integer	The update's unique identifier. Update identifiers start from a certain positive number and increase sequentially. This ID becomes especially handy if you're using Webhooks, since it allows you to ignore repeated updates or to restore the correct update sequence, should they get out of order.
message	Message	**Опционально**. New incoming message of any kind — text, photo, sticker, etc.
inline_query	InlineQuery	**Опционально**. New incoming inline query
chosen_inline_result	ChosenInlineResult	**Опционально**. The result of an inline query that was chosen by a user and sent to their chat partner.
callback_query	CallbackQuery	**Опционально**. New incoming callback query

Рисунок 1 – Параметри класу Update

Метод getUpdates використовується для отримання оновлень через long polling. Відповідь повертається у вигляді масиву об'єктів Update. Параметри методу getUpdates показані на рис.2.

[2)] [17] JavaScript Object Notation стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/JSON (дата звернення 12.06.19).

Параметры	Тип	Обязательный	Описание
offset	Integer	Необязательный	Identifier of the first update to be returned. Must be greater by one than the highest among the identifiers of previously received updates. By default, updates starting with the earliest unconfirmed update are returned. An update is considered confirmed as soon as getUpdates is called with an *offset* higher than its *update_id*. The negative offset can be specified to retrieve updates starting from *-offset* update from the end of the updates queue. All previous updates will forgotten.
limit	Integer	Необязательный	Limits the number of updates to be retrieved. Values between 1—100 are accepted. Defaults to 100.
timeout	Integer	Необязательный	Timeout in seconds for long polling. Defaults to 0, i.e. usual short polling

Рисунок 2 – Параметри медоту getUpdates

Перший і найбільш простий варіант полягає в періодичному опитуванні серверів Telegram на предмет наявності нової інформації. Все це здійснюється через так званий Long Polling, тобто відкривається з'єднання на нетривалий час і всі оновлення тут же відправляються боту. Просто, але не дуже надійно. По-перше, сервери Telegram періодично починають повертати помилку 504 (Gateway Timeout), через що деякі боти впадають в ступор. По-друге, якщо одночасно запущено кілька ботів, ймовірність зіткнутися з помилками зростає.

Вебхукі працюють трохи інакше. Під установкою вебхука мається на увазі, що тепер якщо в чат приходить повідомлення, то Telegram сам говорить про це. Відпадає необхідність періодично опитувати сервери, тим самим, зникає причина падінь спамерських пошукових роботів. Однак за це доводиться платити необхідністю встановлення повноцінного веб-сервера на ту машину, на якій планується запускати ботів. Так само для роботи треба мати власний SSL-сертифікат (Secure Sockets Layer)[18][1)], тому що вебхукі в Telegram працюють тільки по HTTPS.

[1)] [18] Secure Sockets Layer стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/SSL (дата звернення 12.06.19).

В ході написання чат-бота були протестовані, і використані наступні методи і типи:

- метод getUpdates використовується для отримання оновлень через long polling. Відповідь повертається у вигляді масиву об'єктів Update;
- метод setWebhook необхідний для завдання URL вебхука, на який бот буде відправляти оновлення. Кожен раз при отриманні оновлення на цю адресу, буде відправлений HTTPS POST з серіалізованим в JSON об'єктом Update. При невдалому запиті до вашого сервера, спроба буде повторена помірне число раз. Для більшої безпеки рекомендується включити токен в URL вебхука, наприклад, так: https://yourwebhookserver.com/ <token>. Так як ніхто сторонній не знає вашого токена, ви можете бути впевнені, що запити до вашого вебхуку відправляє саме Telegram. Параметри методу setWebhook показані на рис.3;
- метод getWebhookInfo містить інформацію про поточний стан вебхука. Параметри методу getWebhookInfo представлені на рис. 4;
- метод sendMessage використовується для відправки повідомлень;
- метод sendPhoto використовується для відправки фото;
- метод editMessageText використовується для редагування текстових повідомлень, відправлених ботом або через бота;
- об'єкт типу User надає інформацію про бота або користувача Telegram;
- об'єкт типу Chat являє собою інформацію про чати;
- об'єкт типу Message є інформацією про повідомлення;
- об'єкт типу ReplyKeyboardMarkup є клавіатурою з опціями відповіді;
- об'єкт типу KeyboardButton є ще однією кнопкою на клавіатурі відповідей. Для звичайних текстових кнопок цей об'єкт може бути змінен на рядок, що містить текст на кнопці;
- об'єкт типу InlineKeyboardMarkup є вбудованою клавіатурою, яка з'являється під відповідним повідомленням;

– об'єкт типу InlineKeyboardButton є ще однією кнопкою вбудованої клавіатури. Ви обов'язково повинні задіяти рівно одне опціональне поле.

Параметры	Тип	Обязательный	Описание
url	String	Нет	HTTPS url для отправки запросов. Чтобы удалить вебхук, отправьте пустую строку.
certificate	InputFile	Нет	Загрузка публичного ключа для проверки корневого сертификата. Подробнее в разделе про самоподписанные сертификаты.

Рисунок 3 – Параметри методу setWebhook

Поле	Тип	Описание
url	String	URL вебхука, может быть пустым
has_custom_certificate	Boolean	True, если вебхук использует самозаверенный сертификат
pending_update_count	Integer	Количество обновлений, ожидающих доставки
last_error_date	Integer	*Опционально*. Unix-время самой последней ошибки доставки обновления на указанный вебхук
last_error_message	String	*Опционально*. Описание в человекочитаемом формате последней ошибки доставки обновления на указанный вебхук

Рисунок 4 – Параметри методу getWebhookInfo

2.6 Вибір середовища розробки

В якості середи розробки був обраний текстовий редактор кода «Atom»[19][1)] з підтримкою інтерпретатора мови Python, показаний на рис 5.

Atom – це безкоштовний текстовий редактор з відкритим вихідним кодом для Linux, macOS, Windows з підтримкою плагінів, написаних на

[1)] [19] Веб-сторінка текстового редактору коду «Atom». URL: https://atom.io/ (дата звернення 12.06.19).

Node.js[20][2)], і вбудованих під керуванням Git[21][3)]. Більшість плагінів мають статус вільного програмного забезпечення, розробляються і підтримуються спільнотою.

Atom заснований на Electron (раніше відомий як Atom Shell) – фреймворку крос-платформної розробки з використанням Chromium і io.js. Редактор написаний на CoffeeScript і LESS. Версія 1.0 була випущена 25 червня 2015 р.

Atom підтримує такі мови програмування:

C/C++, C#, Clojure, CSS, CoffeeScript, Markdown (GitHubFlavored), Go, Git, HTML, JavaScript, Java, JSON, Julia, Less, Make, Mustache, Objective-C, PHP, Perl, Property List (Apple), Python, Ruby on Rails, Ruby, Sass, Shell script, Scala, SQL, TOML, XML, YAML (Всі мови)

[2)] [20] Node.js стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Node.js (дата звернення 12.06.19).

[3)] [21] Git стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Git (дата звернення 12.06.19).

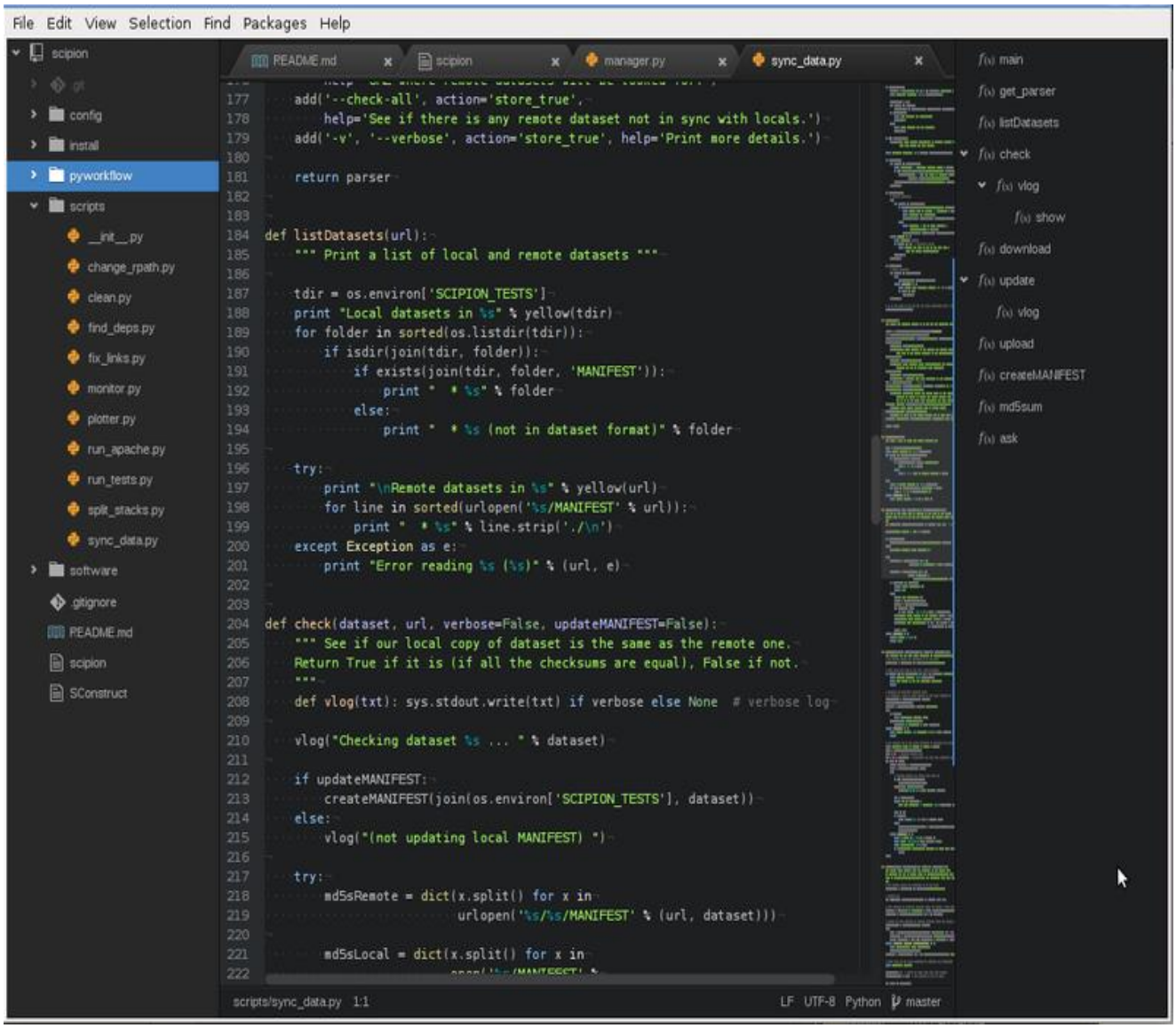

Рисунок 5 – Редактор коду «Atom»

3 ПРОЕКТУВАННЯ

3.1 Реалізація алгоритму за допомогою блок-схеми

Схема – графічне представлення визначення, аналізу або методу розв'язання задачі, в якому використовуються символи для відображення даних, потоку, обладнання та т. д.

Блок-схема – поширений тип схем (графічних моделей), що описують алгоритми або процеси, в яких окремі кроки зображуються у вигляді блоків різної форми, з'єднаних між собою лініями, що вказують напрямок послідовності. Правила виконання регламентуються ГОСТ 19.701-90 «Схеми алгоритмів, програм, даних і систем. Умовні позначення і правила виконання»[21][1)]. Стандарт зокрема регулює способи побудови схем і зовнішній вигляд їх елементів.

У автоматиці функціональна схема, або блок-схема САР, складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема зображаються більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму показані в табл. 2.

Таблиця 2 – Основі елементи схем алгоритму

Найменування	Позначення	Функція

[1)] [22] Електронна версія єдиної системи програмної документації. URL: http://www.pntd.ru/19.701.htm (дата звернення 12.06.19).

Термінатор		Елемент відображає вхід у зовнішнє середовище або вихід з нього (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.
Процес		Елемент відображає одну або кількох операцій, обробку даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.
Рішення		Елемент відображає обробку умови, рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижньої). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижньої) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Продовження таблиці 2

Зумовленний процес		Елемент відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і пе редані в нього дані.
Дані		Елемент відображає перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).
Цикл з параметром		Елемент відображає заголовок циклу з параметром. У ньому через крапку з комою вказуються ім'я змінної (параметра) з початковим значенням, граничне значення параметра (або умова виконання циклу), крок зміни параметра.
Межа циклу		Елемент складається з двох частин – відповідно, початок і кінець циклу – операції, що виконуються всередині циклу, розміщуються між ними. Умови циклу і збільшення записуються всередині символу початку або кінця циклу – в залежності від типу організації циклу. Часто для зображення на блок-схемі циклу замість цього символу використовують символ рішення, вказуючи в ньому умову, а одну з ліній виходу замикають вище в блок-схемі.

Кінець таблиці 2

З’єднувач		Елемент відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.
Коментар		Елемент використовується для детальнішої інформації про кроки, процесу або групи процесів. Опис поміщається з боку квадратної дужки і охоплюється нею по всій висоті. Пунктирна лінія йде до описуваного елементу, або групи елементів (при цьому група виділяється замкнутою пунктирною лінією). Також символ коментаря слід використовувати в тих випадках, коли обсяг тексту в будь-якому іншому символі (наприклад, символ процесу, символ даних та ін) перевищує його обсяг.

3.2 Вибір програми для створення блок-схем

Draw.io – це сервіс, призначений для формування діаграм і схем. Сервіс розділений на три частини – меню, панель об'єктів і сам документ.

За допомогою веб-сервісу Draw.io можна створювати:

– діаграми;

– UML-моделі;

– вставка в діаграму зображень;

– графіки;
– блок-схеми;
– форми.

Для початку користувач може вибрати об'єкт з панелі, переглянувши категорії, і перенести мишею об'єкт в документ. Для з'єднання об'єктів блок-схеми необхідно виділити другий об'єкт і навести покажчиком на перший, далі з'явиться зелений прапорець і за допомогою нього виконується перетягування. Блок-схема алгоритму показана на рис. 6.

В меню сервісу діаграму або схему можна відформатувати наступними настройками:

– стиль шрифту;
– колір фону документа або об'єктів;
– тіні і ступінь прозорості;
– колір і товщина ліній;
– заливка і градієнт.

Також доступний експорт готових схем в зображення (PNG, GIF, JPG, PDF), синхронізація отриманих документів з Google Диском.

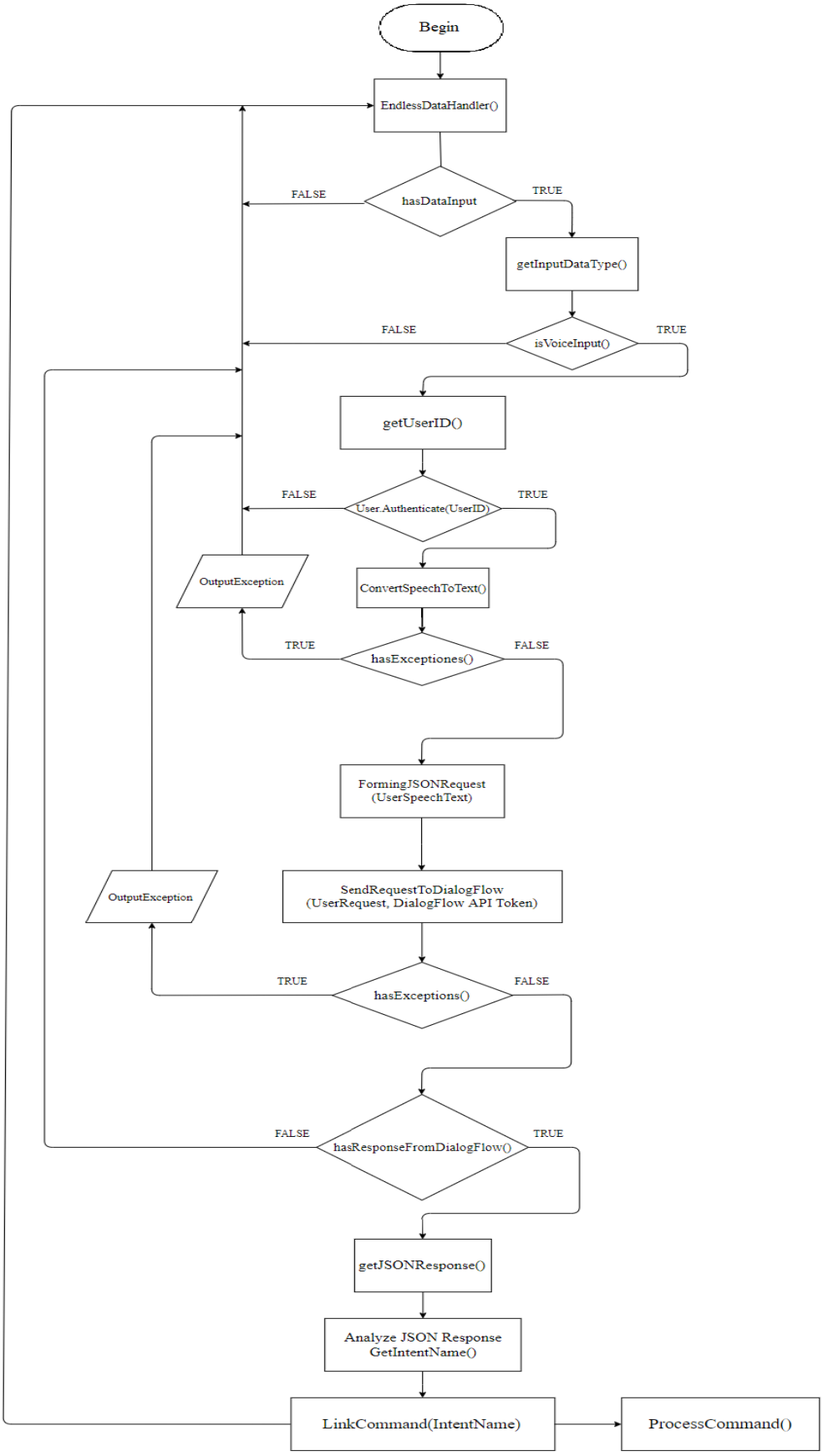

Рисунок 6 – Блок-схема алгоритму

4 РЕАЛІЗАЦІЯ

4.1 Особливості реалізації роботи бота

Даний програмний продукт був розроблений на базі мови програмування Pythonver.3.7.0. З використанням бібліотеки PythonTelegram-Bot.

Ця бібліотека спрощує використання відкритого APITelegram і дозволяє зусередитися лише на реалізації проекту.

Ідея роботи бота заключається в безпереривному очікуванні даних від користувача, який зареєстрований в базі та має відповідний рівень доступу. Якщо користувач незареєстрований, то він отримає повідомлення про це. Використання бота спрямоване на керування процессами компьютера, або інших заданих систем, тому доступ повинен бути лише у адміністратора та в певному колі користувачів, який обирає сам адміністратор .

Очікування реалізується за допомогою класу бібліотеки, називаємого telegram.ext.Updater :

```
Class telegram.ext.Updater
(tken = None, base_url = None, workers = 4, bot = None, private_
key = None, private_key_password = None, user_sig_handler = None
, request_kwargs = None)
Bases:object
```

4.2 Початок створення бота

На початку створюється об’єкт updater класу Updater, для подальшого використання методів:

```
updater = Updater(token=config['Tokens']['Telegram_Token'])
```

В конструктор класу аргументом передається ключ телеграм боту, який видається системою при створенні боту. Даної системою в Telegram виступає сам бот показаний на рис. 7 і 8.

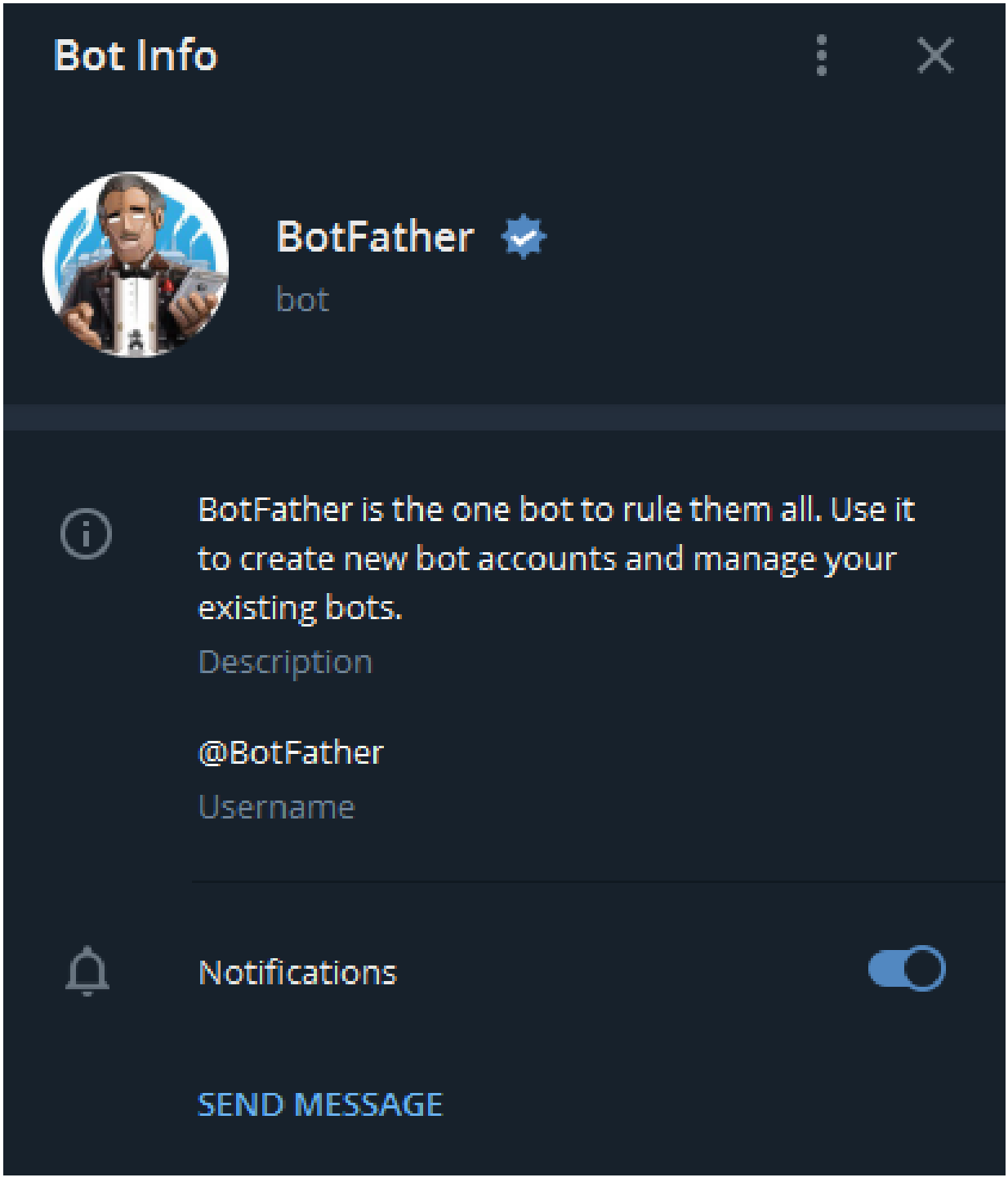

Рисунок 7 – Система реєстрації боту

Почавши з ним листування, користувач відразу ж побачить у діалоговому вікні командний список, що дозволяє розробникам виконувати наступні маніпуляції:

– / Newbot – створити новий чат-бот;
– / Setname – змінити ім'я;
– / Setdescription – змінити опис, в якому вказуються основні можливості робота;
– / Setabouttext – вказати коротку інформацію про даному аккаунті;
– / Setuserpic – встановити аватарку;
– / Setcommands – вказати список підтримуваних функцій, який буде виводитися в діалоговому вікні при введенні символу «/»;
– / Setjoingroups – можливість додавання бота в групи;
– / Deletebot – видалення роботизованого аккаунта.

Для створення чат-бота в діалоговому вікні необхідно ввести / newbot. Після цього BotFather запитає його повне ім'я (name), яке буде відображатися в чатах і контактах Telegram, і коротке ім'я (username), що складається з латинських букв і закінчується на «bot», наприклад image_bot.

Як зазначалося вище, кожен бот в телеграмі володіє унікальним функціоналом, що дозволяє виконувати ті чи інші завдання. Разом з тим, існує ряд простих команд, які розпізнаються практично всіма роботизованими акаунтами:

– / Start – початок спілкування з користувачем, вітання (при використанні додаткових аргументів можна розширити цю функцію);
– / Help – відображення допомоги (коротка інформація про акаунт, опис можливостей, перелік доступних функцій і інше);
– / Settings – надання списку можливих налаштувань, доступних користувачеві.

На даний момент назвати точне число роботів в Telegram практично неможливо. При правильній роботі з чат-ботами вони стають потужним

організаційним ресурсом, що дозволяє автоматизувати виконання повторюваних дій.

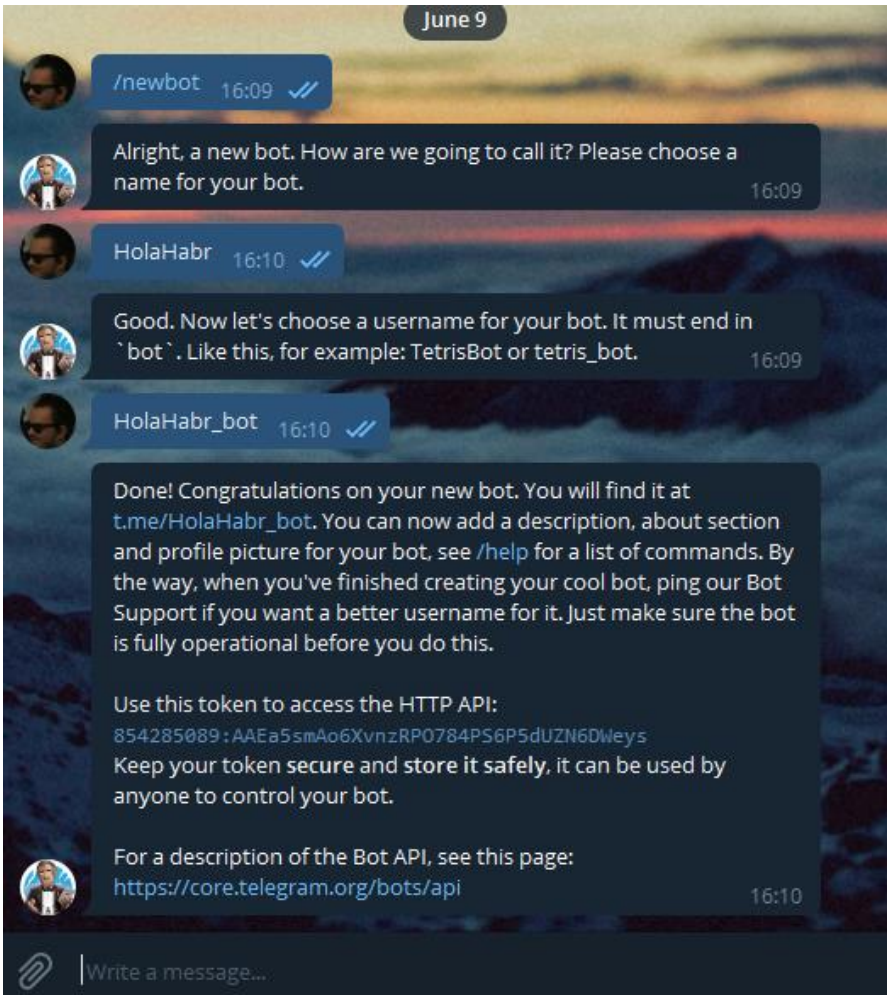

Рисунок 8– Процес створення боту

Ключ (токен) це набір символів типу:
«110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw»

Виступає унікальним та індивідуальним індентифікатором для взаємодії з HTTP-інтерфейсом BotAPI.

Далі створюється об’єктклассу dispatcher

```
dispatcher = updater.dispatcher
```

Цей клас посилає всі види оновлень своїм зареєстрованим обробникам.

Важливим методом цього класу є

```
add_handler(handler, group = 0)
```

За його допомогою реєструється обробник

Обробник повинен бути екземпляром підкласу telegram.ext.handler. всі обробники організовані в групи з числовим значенням. Групою за стандартом є 0. Всі групи будуть оцінені для обробки оновлення, але буде використано лише 0 або 1 обробник на групу. Якщо telegram.ext.dispatcherhandlerstop піднята від одного з обробників, додаткові обробники (незалежно від групи) не будуть викликані.

Пріоритет / порядок обробників визначається наступним чином:

- Пріоритет групи (нижчий номер групи == більш високий пріоритет);
- Буде використаний перший обробник у групі, який повинен обробляти оновлення, інші обробники з групи не будуть використовуватися порядок, в якому обробники були додані до групи, визначає пріоритет.

Параметри:

- handler (telegram.ext.Handler) – Екземпляр обробника;
- group (int , optional) - Ідентифікаторгрупи. Стандартно- 0.

Далі ми вказуємо яку процедуру треба виконувати при умові, що бот отримав саме голосове повідомлення від користувача

```
voice_message_handler = MessageHandler(Filters.voice,
Voice)
```

Після цього додаємо обротник з цими параметрами

```
- dispatcher.add_handler(voice_message_handler)
```

Voice – процедура, яка буде викликана при виникненні перервання за параметром голосового повідомлення

На початку цієї процедури йде перевірка користувача на факт реєстрації у системі.

```
autor(update.message.chat.id)
```

Аргументом є унікальний id користувача, який порівнюється з тими, що прописані у спеціальному файлі адміністратора, де вказані id, яким дозволяється користуватися ботом

Одним із параметрів процедури є голосове повідомлення, яке є лише звукою доріжкою, отже потрібно якимось чином отримати саме текстове повідомлення. З цим допоможе стороння бібліотека speech_recognition

Ця бібліотека є обгорткою багатьох відкритих API сервісів «texttospeech» таких як:

- Google Speech Recognition;
- Microsoft Bing Voice Recognition;
- Houndify API;
- IBM Speech to Text.

Якими можливо користуватися як при підключенні до мережі Інтернет так і оффлайн.

Основною ідеєю «оболонки» бота є нейронна мережа, яка може аналізувати поступаючий до неї текст, та давати відповіді на поступаючі запити.

В цієї ролі виступає DialogFlow від Google. DialogFlow – це сервіс, що дозволяє створювати чат-ботів для різних платформ і мов на різних пристроях.

Даний сервіс дозволяє інтегруватися з багатьома іншими сервісами, від яких можна отримувати повідомлення для обротки.

4.3 Налаштування сервісу DialogFlow

Для початку потрібно зареєструватися та створити нового агента:

Назву агента можна вказати будь-яке. Мову обрано російську. Але в подальшому можна додати і інші.

Часовий пояс важливий пункт, так як по ньому будуть визначаться такі фрази як «Сьогодні», «3 дні тому».

Реєстрація агента показана на рис. 9.

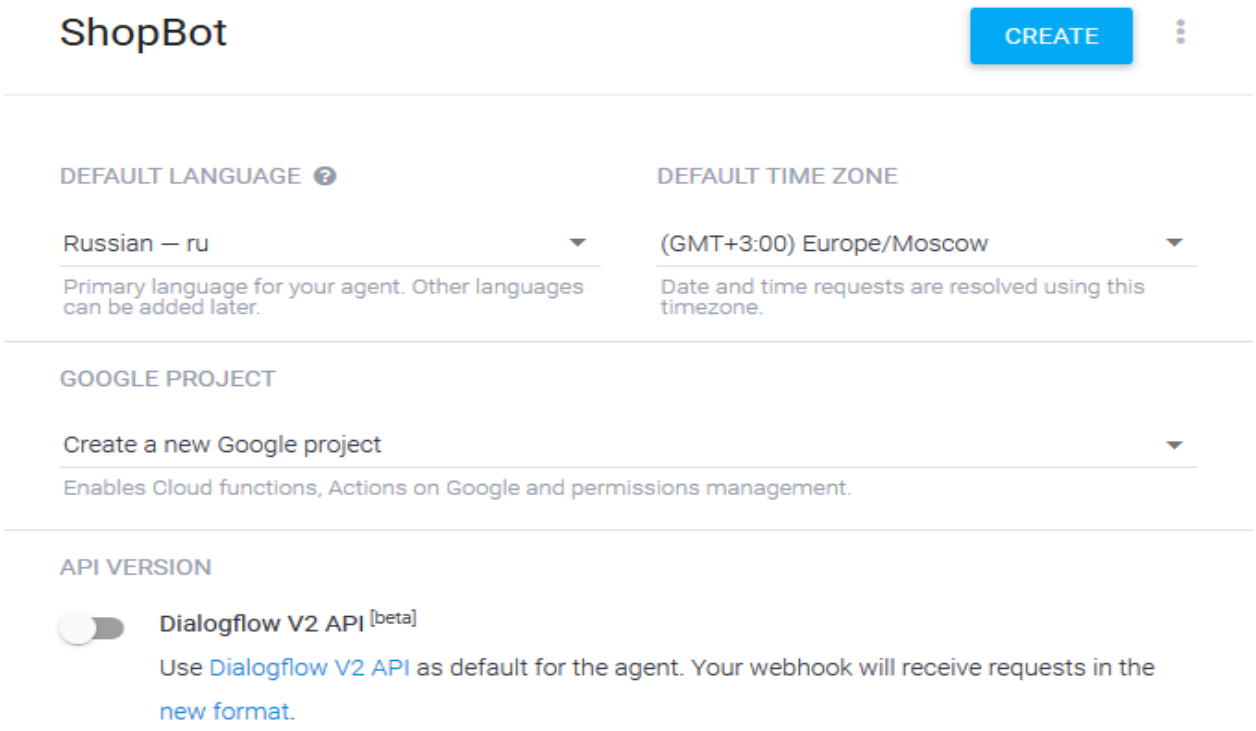

Рисунок 9 – Реєстрація агенту DialogFlow

Створимо новий Google проект. Будемо використовувати першу версію API.

Після створення агента переходимо на вкладку «Integrations», яка показана на рис. 10 і підключаємо ті платформи, які нам потрібні.

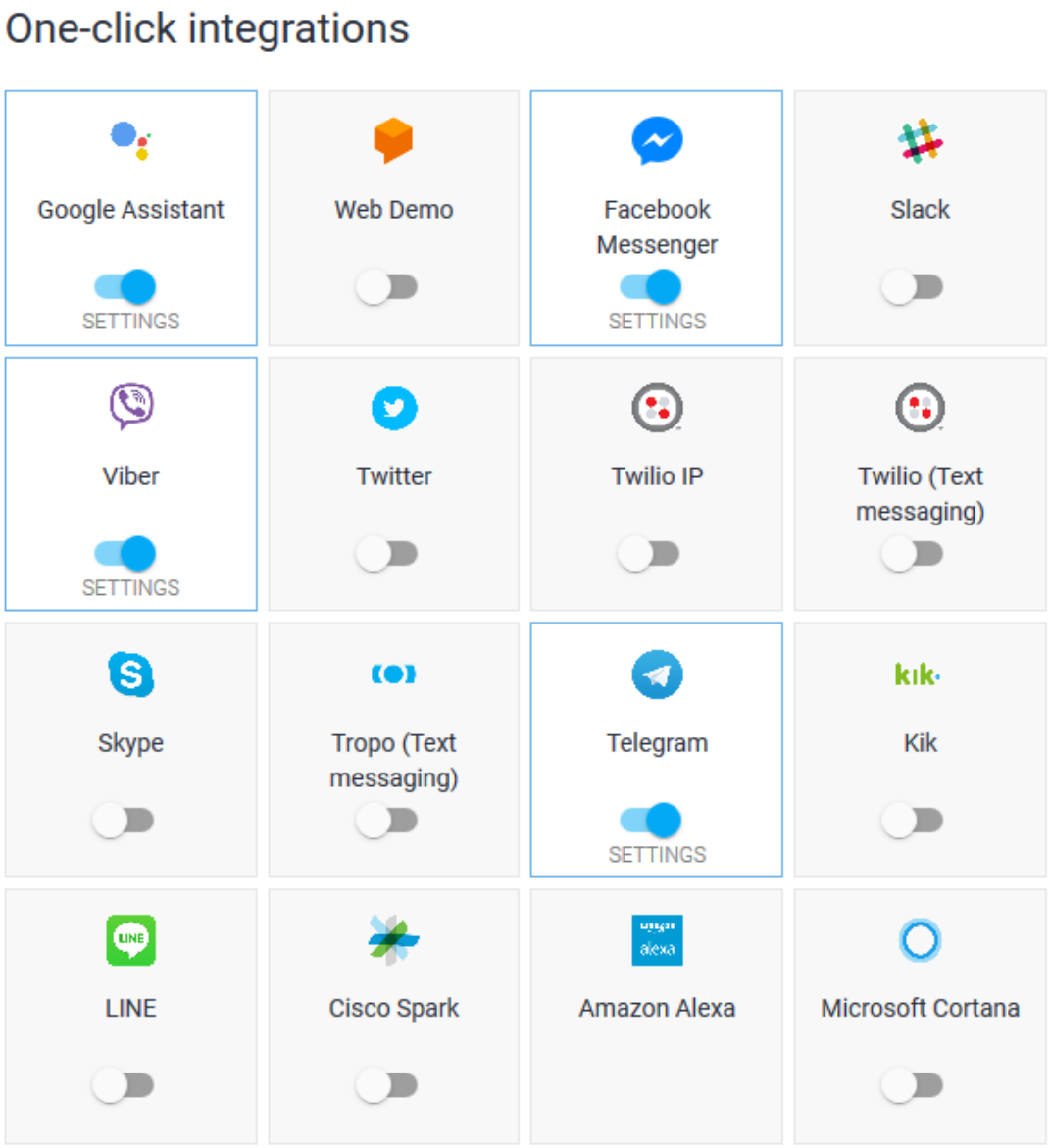

Рисунок 10 – Доступні сервіси для інтеграції

На цьому етапі наш бот уже запущений і може спілкуватися з користувачами. Тепер потрібно навчити його правильно розуміти фрази.

4.4 Створення призначень

Призначення - це оброблювач певного запиту від користувача.

Переходимо на вкладку «Intents». Після створення агента тут будуть два стандартних призначення:

– Default Welcome Intent – запускається для вітання користувача (відправить у відповідь вітання);
– Default Fallback Intent – запускається в тому випадку, якщо жодна з інших призначень не підходить. Відправить повідомлення, що користувача не зрозуміли («Ось ця остання фраза мені не зрозуміла», «Спробуйте, будь ласка, висловити свою думку по-іншому» і т.д.).

Далі ми реалізуємо наступне: коли користувач вітає бота, той питає у нього його ім'я та адресу проживання, щоб використовувати це в подальшому.

Додамо в вітання питання про користувача. Відкриваємо «Default Welcome Intent». Тут є кілька секцій:

– Contexts;
– User says;
– Events;
– Action;
– Response.

Зверніть увагу на секцію «Events»: тут вказано одне службове подія «WELCOME».

Призначення можна запустити двома способами: як реакцію на текст користувача або за подією.

У нашому випадку призначення «Default Welcome Intent» реагує не на якийсь текст від користувача, а на службове подія WELCOME. Наприклад, для Telegram це запуск бота командою / start. Детальніше про події.

Отже, розглянемо секцію «Response» показану на рис. 11. Тут перераховані повідомлення, які будуть відправлені у відповідь користувачеві. Кожне повідомлення може складатися з декількох рядків. Кожен рядок –

окремі варіанти повідомлення. Для відповіді буде обраний один з цих варіантів. І чим їх більше, тим рідше ваш бот буде повторюватися.

Додамо сюди повідомлення з проханням розповісти про себе.

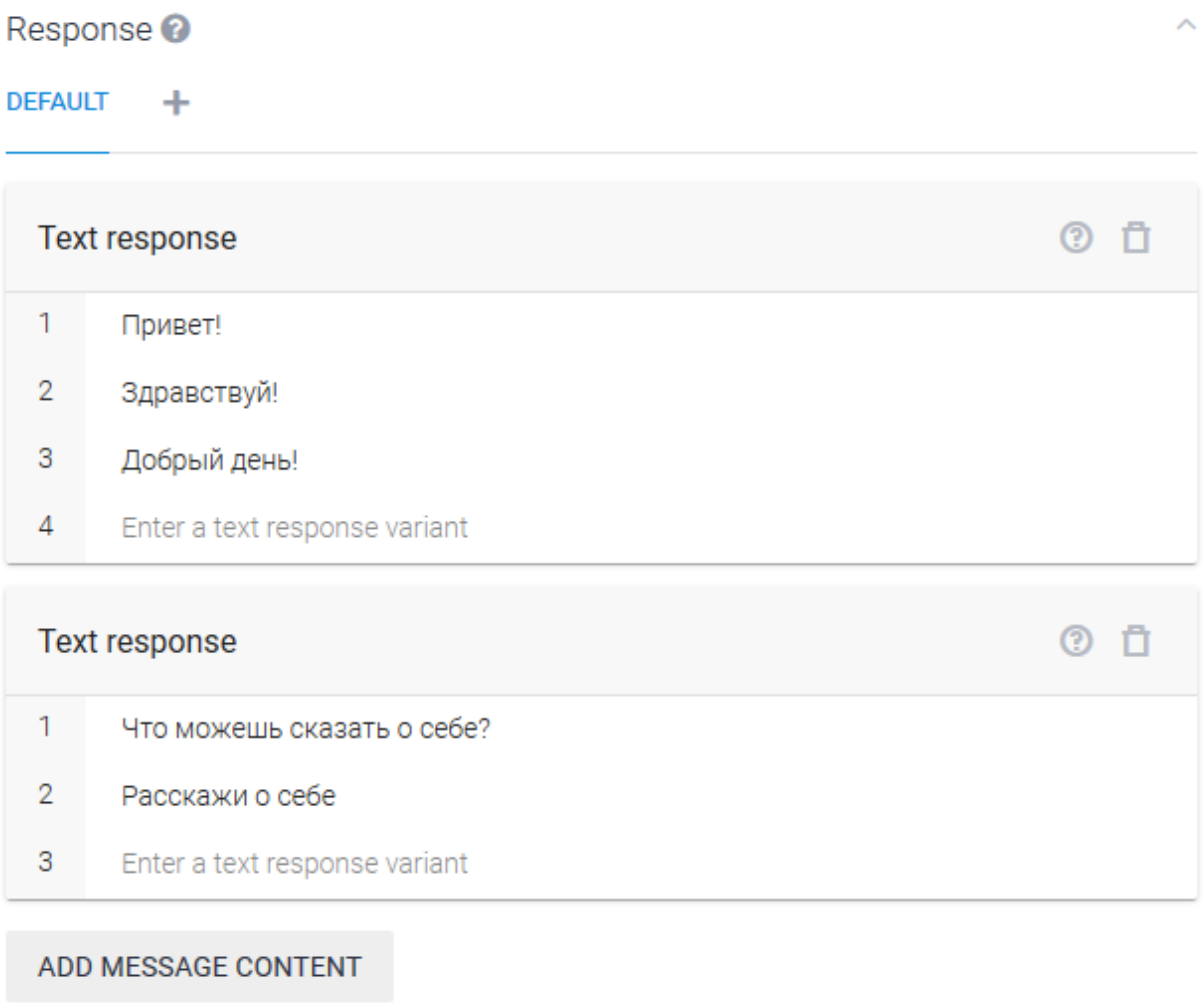

Рисунок 11 – Вікно додавання варінтів відповідей та запитанньь

Реакція на відповідь

Тепер створимо призначення, яке буде реагувати на подальше повідомлення.

На вкладці «Intents» наводимо курсор на «Default Welcome Intent» і натискаємо «Add follow-up intent». У випадаючому списку, нам

запропонують кілька встановлених варіантів. Нас вони не цікавлять - вибираємо «Custom»

Перейдемо до секції «User says» див. рис. 12 і додамо кілька варіантів повідомлень від користувача.

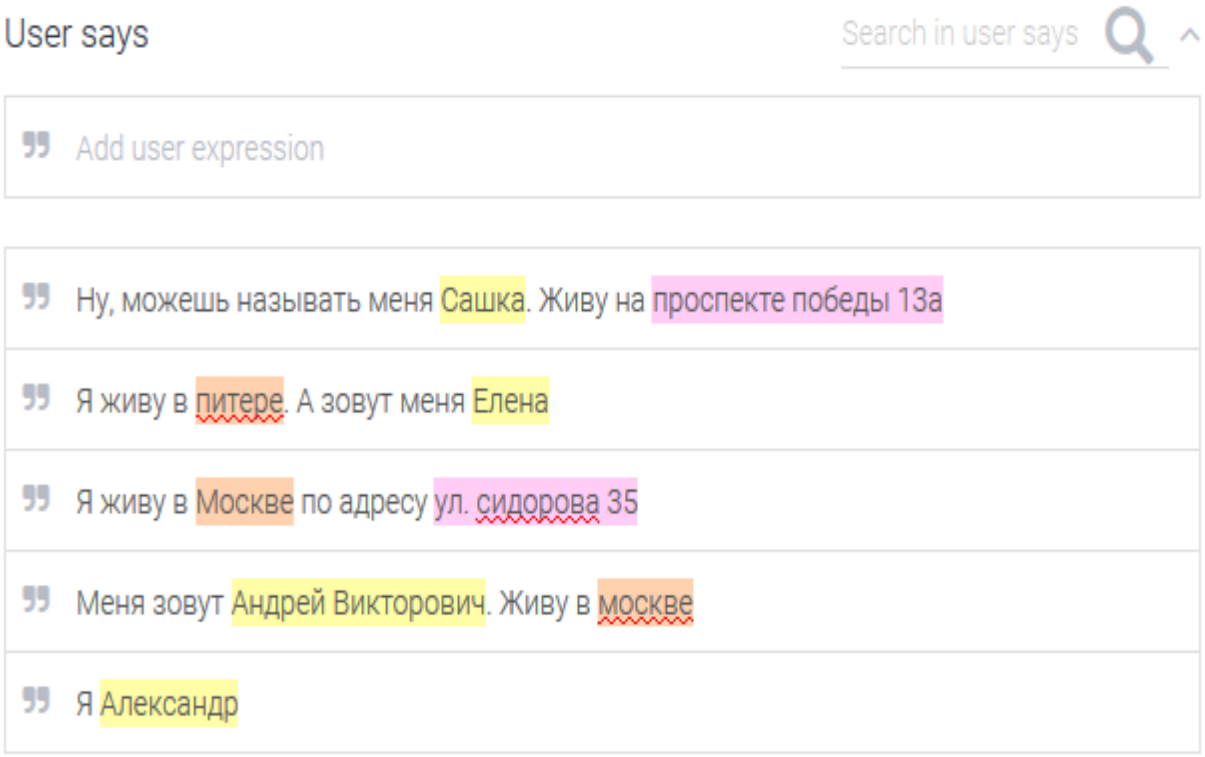

Рисунок 12 – Варінти звернень користувача

Тепер потрібно вказати, які дані в цих повідомленнях нам потрібні. Для цього слід вибрати фрагмент тексту і в випадаючому меню вибрати тип даних. У Dialogflow є ряд попередньо встановлених типів, зокрема для імен і адрес. Але ми можете створювати і власні.

Зараз звернемо увагу на секцію «Action» зображену на рис. 13. Тут перераховані всі параметри, які збирає це призначення. Параметри, якими ми розмічали текстові повідомлення, автоматично імпортувалися.

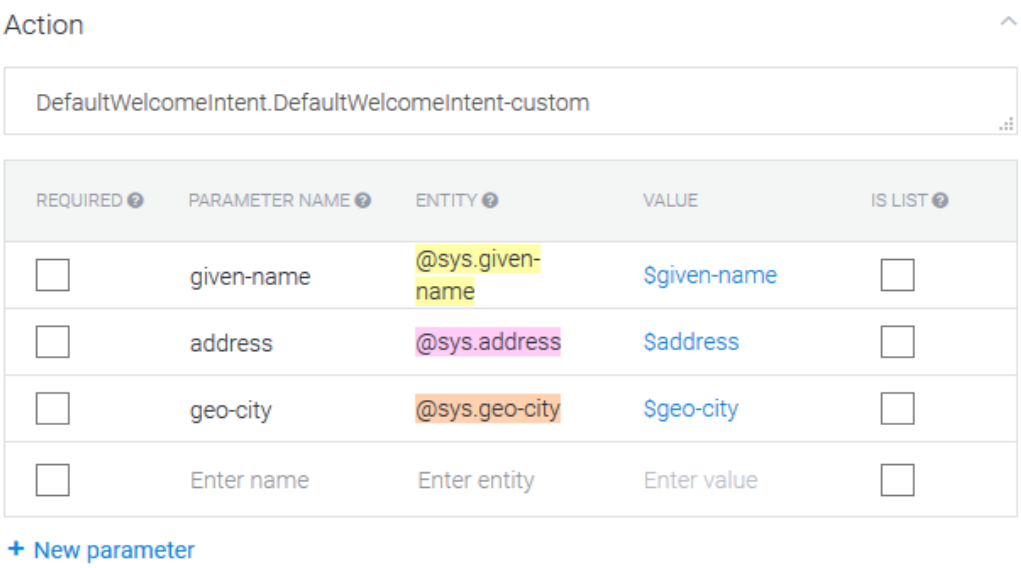

Рисунок 13 – Параметри текстових повідомлень

Позначимо ім'я користувача як обов'язковий параметр. Змінимо назву параметра з «given-name» на «name» для більшої зручності. І в колонці «Prompts» вкажемо питання «Як вас звати?» це зображено на рис. 14. Тепер, так як «name» це обов'язковий параметр, в разі якщо в повідомленні не буде цього параметра, користувачеві буде відправлений один з питань в «Prompts» для отримання імені.

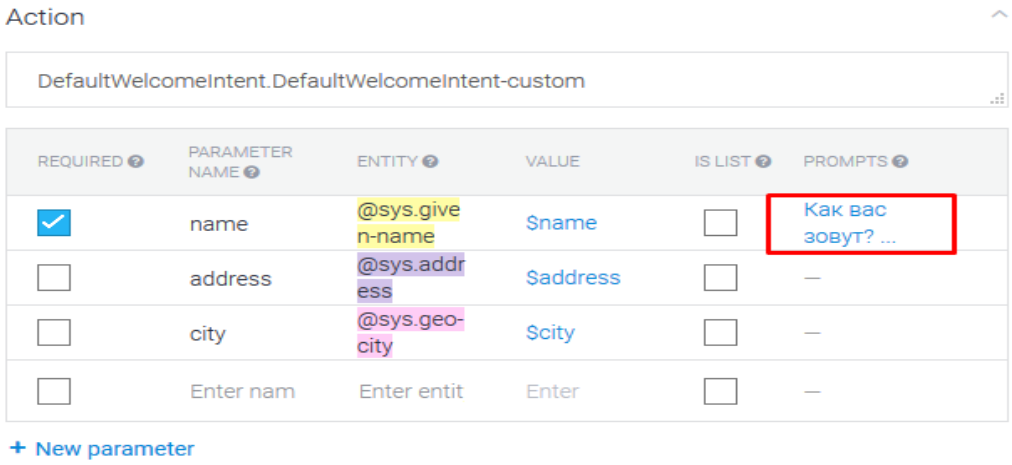

Рисунок 14 – Секція «Action» з запитом «Prompts»

І, нарешті, додамо відповідь користувачеві в секцію «Response» див. рис. 15.

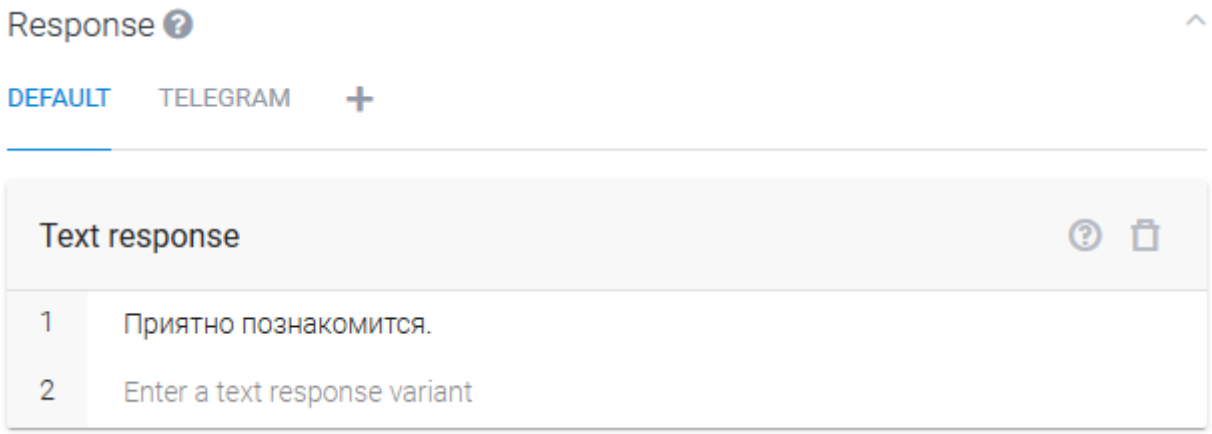

Рисунок 15 – Секція «Response»

Кожне призначення може працювати з вхідним контекстом і створювати вихідний контекст.

Якщо у призначення є вхідний контекст, то призначення буде запускатися тільки в тому випадку, якщо цей контекст існує. Якщо у призначення є вихідний контекст, то він буде створений і всі зібрані дані будуть записані в нього.

Розглянемо по порядку:

– Default Welcome Intent – входить контексту немає. Тому призначення може запуститися для будь-якого запиту. Призначення створює новий порожній контекст «DefaultWelcomeIntent-followup» (додасться автоматично);
– Default Welcome Intent-custom – має вхідний контекст «DefaultWelcomeIntent-followup» (додався автоматично). Це призначення може запуститься тільки в тих випадках, коли цей

контекст існує (коли користувач привітався). Вихідного контексту тут поки-що немає.

Додамо вихідний контекст для «Default Welcome Intent-custom» з назвою «UserData». Тепер, коли після привітання користувач привітається, інформація про нього буде записана в новий контекст «UserData».

Для даного проекту нам потрібен лише Telegram. Отримуємо текст за допомогою GoogleSpeechRecognition:

```
text = r.recognize_google(audio, language="ru-RU")
```

Далі формуємо запит до DialogFlow:

```
Request = apiai.ApiAI(config['Tokens']['DialogFlow_Token']).text
_request()
```

Задаємо мову запиту:

```
request.lang = 'ru'
```

Отримуємо JSON объект

```
responseJson                                                    =
json.loads(request.getresponse().read().decode('utf-8'))
```

З даного об’єкту беремо відповідь:

```
response = responseJson['result']['fulfillment']['speech']
```

Важливою функцією є DialogFlow Intent. За допомогою цієї функції можливо будувати діалоги з ботом, коли користувач прописує спеціальні фрази та також відповіді, які бот повинен надіслати показані на рис. 16.

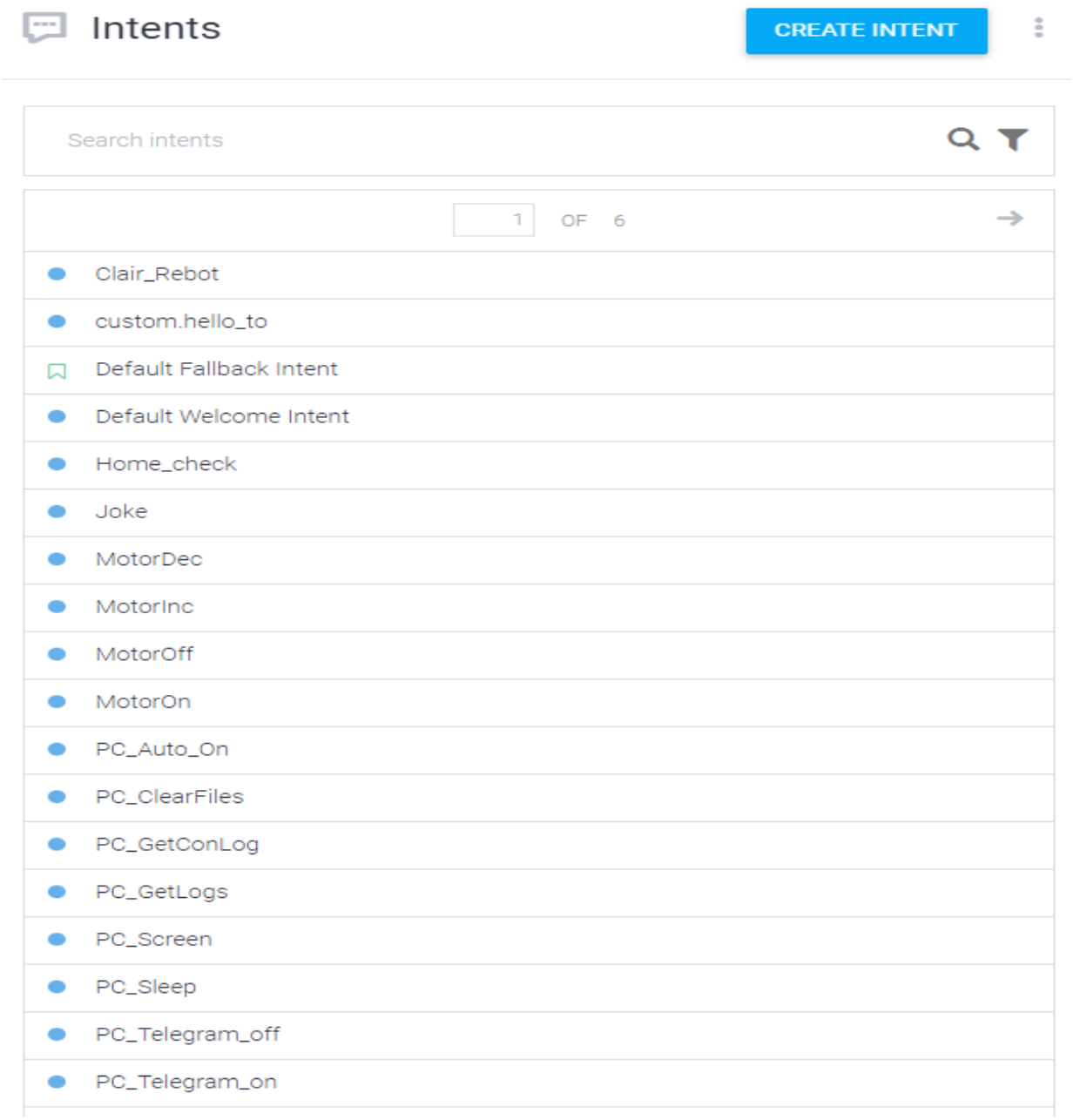

Рисунок 16 – Загальний список Intents

Даний Intent зображений на рис. 17 і 18 відповідає в випадку, коли користувач назвав спеціальні фрази.

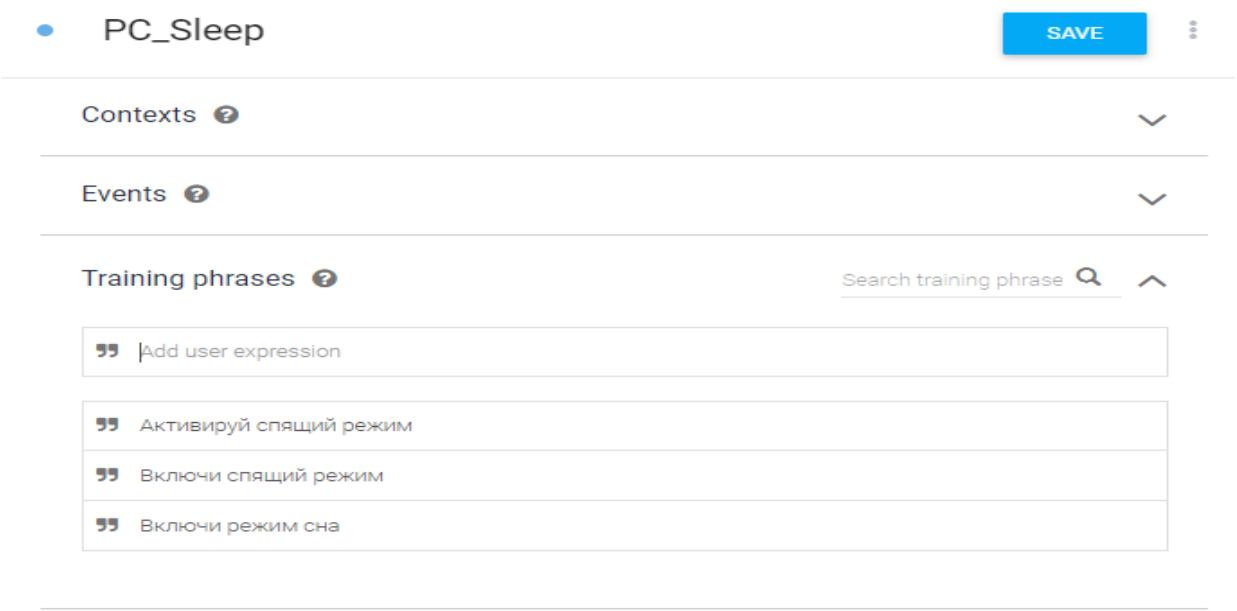

Рисунок 17 – Intent PC_Sleep

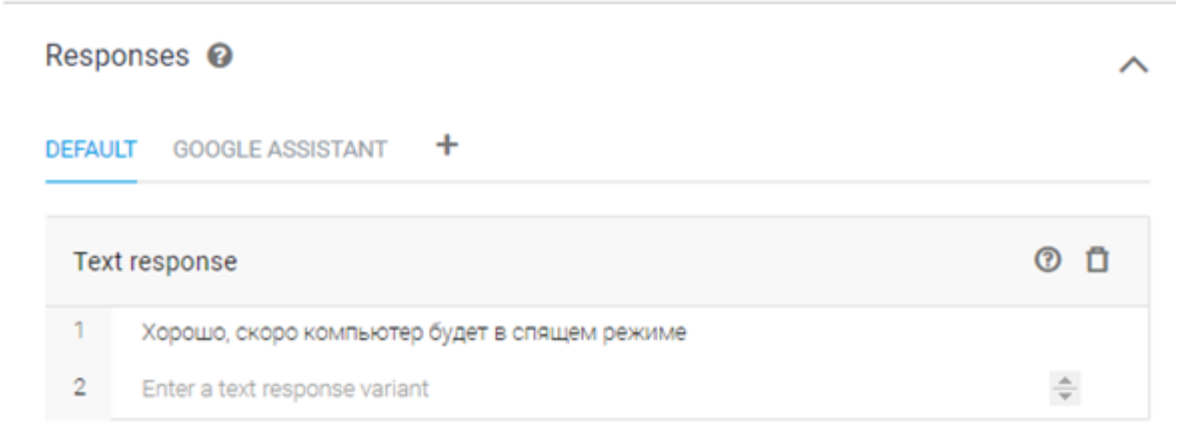

Рисунок 18 – Відповідь PC_SleepIntent

Після того як бот написав в чат відповідь треба йому дати зрозуміти як виконати команду. Зв’язок запит-команда відбувається за допомогою регулярних вираженнь. З попереднього JSON об’єкту беремо назву intent-а.

```
intentName = responseJson['result']['metadata']['intentName']
```

Тепер потрібно для цього інтенту вказати команду для виконання. Для цього створен файл-словник, в якому зберігається назва інтенту та команда, яка повинна виконуватися, при його виклику.

- PC_Telegram_off -teleOff – данний інтент виконує команду, яка вимикає сторонню програму «Telegram»;
- PC_Telegram_on -teleOn – данний інтент виконує команду, яка вмикає сторонню програму «Telegram»;
- PC_Screen -getScreen – данний інтент виконує команду, яка робить знимок екрану;
- PC_GetLogs -getLogs – данний інтент виконує команду, яка викликає записи помилок;
- PC_GetConLog -getConLogs – данний інтент виконує команду, яка викликає записи критичних помилок;
- PC_Auto_Off -pcOff – данний інтент виконує команду, яка вимикає комп’ютер;
- PC_Auto_On -pcOn – данний інтент виконує команду, яка вмикає комп’ютер.

Прикладом вище є записи в цьому файлі. Де спочатку йде назва інтенту, далі дефіс та команда для виконання. Команда є не що інше як назва файлу, який буде викликаний основною програмою, при обробці. Цей файл може бути як такий самий pythonscript, так і bat-файл для системних команд, наприклад вимкнення компьютера.

ВИСНОВКИ

В ході роботи проведено аналіз предметної області, актуальних технологій і програмних рішень, так само була вивчена документація програмного інтерфейсу Bot API, після чого було спроектовано і розроблено програмне забезпечення для сервісу обміну повідомленнями Telegram с використання Bot API, яке, згодом, було протестовано і налагоджено.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЬ

1. Джозеф Вейценбаум стаття з «Вікіпедії» вільної енциклопедії. URL: https://is.gd/ZicOUC. (дата звернення 12.06.19).

2. «Еліза» стаття з «Вікіпедії» вільної енциклопедії. URL: https://is.gd/Slym7h. (дата звернення 12.06.19).

3. Система пошуку «Яндекс». URL: https://yandex.ua/ (дата звернення 12.06.19).

4. Месенджер, що дозволяє обмінюватися повідомленнями і медіафайлами багатьох форматів. URL: https://telegram.org/ (дата звернення 12.06.19).

5. Компанія, яка використовує штучний інтелект і мобільні повідомлення для допомоги студентам у коледжі. URL: https://www.admithub.com/ (дата звернення 12.06.19).

6. Чат-бот компанії «AdmitHub» для допомоги студентам. URL: http://www.olibot.com/ (дата звернення 12.06.19).

7. Дуолинго (англ. Duolingo) - безкоштовна платформа для вивчення мови та краудсорсингових перекладів. URL: https://ru.duolingo.com/ (дата звернення 12.06.19).

8. Сайт компанії «AnzacLiveWizeline».

URL: http://www.anzaclive.com.au/feature.html (дата звернення 12.06.19).

9. Representational State Transfer стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/REST (дата звернення 12.06.19).

10. Application Programming Interface стаття з «Вікіпедії» вільної енциклопедії. URL https://ru.wikipedia.org/wiki/API (дата звернення 12.06.19).

11. SiliconGraphicsInc стаття з «Вікіпедії» вільної енциклопедії.

URL: https://ru.wikipedia.org/wiki/Silicon_Graphics (дата звернення 12.06.19).

12. Microsoft Windows 95 OSR 2,98 стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Windows_95 (дата звернення 12.06.19).

13. WindowsNT 4.0 стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Windows_NT_4.0 (дата звернення 12.06.19).

14. Веб-сторінка визначень типів мови Python. URL: https://svn.python.org/projects/stackless/trunk/Lib/types.py (дата звернення 12.06.19).

15. RFC 2616 – Протокол Передачи Гипертекста – HTTP/1.1 URL: https://rfc2.ru/2068.rfc (дата звернення 12.06.19).

16. HyperText Transfer Protocol стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/HTTP (дата звернення 12.06.19).

17. JavaScript Object Notation стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/JSON (дата звернення 12.06.19).

18. Secure Sockets Layer стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/SSL (дата звернення 12.06.19).

19. Веб-сторінка текстового редактору коду «Atom». URL: https://atom.io/ (дата звернення 12.06.19).

20. Node.js стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Node.js (дата звернення 12.06.19).

21. Git стаття з «Вікіпедії» вільної енциклопедії. URL: https://ru.wikipedia.org/wiki/Git (дата звернення 12.06.19).

22. Електронна версія єдиної системи програмної документації. URL: http://www.pntd.ru/19.701.htm (дата звернення 12.06.19).