

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«___» _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження та швидке прототипування діалогових систем для віртуальних клієнтів
(ботів)

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 122 – “Комп’ютерні науки та інформаційні технології”

Науковий керівник роботи:

(підпис)

к.т.н Щербаков Є.В

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

ст. викл Критська Я.О

(ініціали, прізвище)

Студент:

(підпис)

Гордієнко М.П

(ініціали, прізвище)

Група:

КН-17ДМ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень магістр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 122 – “Комп'ютерні науки та інформаційні технології”
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
I.C. Скарга-Бандурова
« _____ » _____ 20 _____ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Гордієнко Максим Петрович
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та швидке прототипування діалогових систем для віртуальних клієнтів (ботів)

керівник проекту (роботи) Щербаков Є.В, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від _____ " " _____ 2017 р. № _____

2. Строк подання студентом роботи 16.01.2019

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз та постановка задачі; особливості розроблення діалогових систем; огляд існуючих засобів прототипування та розробка діалгових систем для віртуальних клієнтів (ботів) , перевірка реалізованого функціоналу

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

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	<i>Критська Я.О.</i>		

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання, збір матеріалів	5.09.18 - 10.09.18	
2	Огляд літератури й обґрунтування необхідності розробки	11.09.18 – 19.09.18	
3	Отримання завдання по преддипломній практиці, написання звіту	20.09.18 – 20.10.18	
4	Перевірка реалізованого функціоналу	21.10.18 – 20.11.18	
5	Розробка діалогової системи для віртуального бота	21.11.18 – 30.11.18	
6	Розробка заходів з охорони праці	1.12.18 – 10.12.18	
7	Оформлення пояснювальної записки	11.12.18 – 17.12.18	
8	Підготовка та подання магістерської роботи до захисту	18.12.18 – 16.01.19	

Студент _____

(підпис)

Гордієнко М.П _____

(прізвище та ініціали)

Науковий керівник _____

(підпис)

Щербаков Є.В _____

(прізвище та ініціали)

АНОТАЦІЯ

Гордієнко М.П. Дослідження та швидке прототипування діалогових систем для віртуальних клієнтів (ботів)

Розглянуті особливості використання віртуальних клієнтів для потреб вищого навчального закладу. Викладено загальну методику прототипування та розробки діалогових систем. Розроблено спеціальний додаток, що дозволяє абітурієнтам, студентам та викладачам швидко отримувати інформацію через віртуального чат-бота.

Ключові слова: чат-бот, діалогова система, нейронні мережі, тестування, код, програмне забезпечення, віртуальний клієнт, нейро-лінгвістичне програмування.

АННОТАЦИЯ

Гордиенко М.П. Исследование и быстрое прототипирование диалоговых систем для виртуальных клиентов (ботов)

Рассмотрены особенности использования виртуальных клиентов для нужд высшего учебного заведения. Изложена общая методика прототипирования и разработки диалоговых систем. Разработано специальное приложение, позволяющее абитуриентам, студентам и преподавателям быстро получать информацию через виртуального чат-бота.

Ключевые слова: чат-бот, диалоговая система, нейронные сети, тестирование, код, программное обеспечение, виртуальный клиент, нейро - лингвистическое программирование.

ABSTRACT

Gordienko M.P. Research and rapid prototyping of dialogue systems for virtual clients (bots)

The features of the use of virtual clients for the needs of higher education institutions are considered. The general methodology of prototyping and development of interactive systems is described. Developed a special application that allows applicants, students and teachers to get information quickly through a virtual chat bot.

Keywords: chat bot, conversational system, neural networks, testing, code, software, virtual client, neuro - linguistic programming.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1	11
АНАЛІЗ МЕТОДІВ І ЗАСОБІВ РОЗРОБКИ ДІАЛОГОВИХ СИСТЕМ ДЛЯ ВІРТУАЛЬНИХ КЛІЄНТІВ.....	11
1.1 Діалогові системи та віртуальні клієнти (чат-боти).....	11
1.2 Аналіз сучасних засобів створення віртуальних систем. Python	13
1.2.1 Історія.....	13
1.2.2 Переваги	14
1.2.3 Типи й структури даних	15
1.2.4 Недоліки	16
1.2.5 Застосування	17
1.3 Аналіз сучасних засобів створення віртуальних систем. Java	17
1.3.1 Історія.....	18
1.3.2 Переваги	19
1.3.3 Недоліки	19
1.4 Архітектура на основі фреймів	20
1.5 Нейронні мережі	21
1.5.1 Середовище застосування нейронних мереж	21
1.5.2 Типи систем з інтелектуальним інтерфейсом.....	22
1.5.3 Гіпертекстові системи	22
1.5.4 Системи контекстної допомоги.....	23
1.5.5 Системи когнітивної графіки	23
1.6 Машинне навчання	23
1.6.1 Коротка історія та великі надії	23
1.7 Штучні нейронні мережі.....	24
1.8 Аналіз платформ для створення діалогових систем.....	25

1.8.2	Microsoft Bot Framework	26
1.8.3	Wit.ai	26
1.8.4	Semantic Machines	27
1.8.5	Chatfuel.....	27
1.8.6	DialogFlow	27
1.9	Висновки до 1 розділу	28
РОЗДІЛ 2		29
ПРОЕКТУВАННЯ ДІАЛОГОВИХ СИСТЕМ ДЛЯ ВІРТУАЛЬНИХ КЛІЄНТІВ		29
2.1	Вибір діалогової системи	29
2.1.1	LUIS - Language Understanding Intelligent Service (рис. 2.1).....	29
2.1.2	IBM Watson (рис. 2.2)	30
2.1.3	Dialog Flow (рис. 2.3).....	31
2.1.4	WIT.AI (рис. 2.4).....	32
2.1.5	Amazon Lex (рис. 2.5)	32
2.2	Архітектура чат-боту.....	34
2.3	Аналіз вимог.....	35
2.4	Вибір сервісу для обміну повідомленнями	36
2.4.1	Основні можливості Telegram	37
2.5	Діаграма варіантів використання освітнього чат-боту	38
2.6	Висновки до 2 розділу	39
РОЗДІЛ 3		40
РЕАЛІЗАЦІЯ СПРОЕКТОВАНОЇ СИСТЕМИ.....		40
3.1	Засоби реалізації	40
3.1.1	Мова програмування JavaScript.....	40
3.1.2	Переваги JavaScript	41
3.1.3	Недоліки JavaScript.....	41
3.2	Середовище розробки Visual Studio Code	41
3.2.1	Легкість VSC	42
3.2.2	Функції.....	43
3.2.3	Міцна та розширювана архітектура.....	43
3.3	Платформа Dialogflow	43
3.4	Архітектура роботи додатку	44
3.5	Схема роботи додатку та розробка	44
3.5.1	Схема роботи додатку	44
3.6	Розробка чат-боту із використанням діалогових систем	46

3.6.1	Створення чат-боту в месенджері.....	47
3.6.2	Розробка діалогових систем	48
3.6.3	Шаблони Intents	49
3.6.4	Об'єкти Entities	54
3.7	Висновки до 3 розділу	57
РОЗДІЛ 4		58
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ		58
4.1	Загальні питання з охорони праці	58
4.1.1	Правові та організаційні основи охорони праці	59
4.1.2	Організаційно-технічні заходи з безпеки праці.....	59
4.2	Аналіз стану умов праці.....	60
4.2.1	Вимоги до приміщень	60
4.2.2	Вимоги до організації місця праці	61
4.2.3	Навантаження та напруженість процесу праці	61
4.3	Виробнича санітарія	62
4.3.1	Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу.....	62
4.3.2	Пожежна безпека	64
4.3.3	Електробезпека	64
4.4	Гігієнічні вимоги до параметрів виробничого середовища.....	65
4.4.1	Мікроклімат.....	65
4.4.2	Освітлення.....	65
4.5	Вентилювання	67
4.6	Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій.....	67
4.7	Охорона навколишнього природного середовища.....	69
4.7.1	Загальні дані з охорони навколишнього природного середовища	69
4.7.2	Вимоги до збору, пакування та розміщення відходів ІТ галузі	69
4.7.3	Визначення впливу та заходів щодо поводження з відходами ІТ галузі	70
4.8	Висновки до 4 розділу	72
4.8.1	Перелік посилань до 4 розділу	73
ВИСНОВКИ		74
СПИСОК ПОСИЛАНЬ.....		75
Електронні ресурси:		75
ДОДАТОК А		76
Файл Hello.json.....		76

Файл geo.json.....	81
Файл Contacts.json.....	83
Файл address.json.....	85
ДОДАТОК Б.....	87

ВСТУП

Актуальність теми. Вже сьогодні всі створюють ботів. Поняття чат-ботів відомо нам з давніх часів, коли ще була створена ELIZA у 1990 роках. В наш час чат-боти, які використовують обробку природних мов та штучний інтелект спроможні замінити офісних співробітників або консультантів в усіх сферах життя. Вже зараз у всіх крупних компаніях існують віртуальні діалогові системи, тому що це зручно.

Сьогодні чат-боти є частиною віртуальних помічників, таких як Google Assistant, і доступні через програми багатьох організацій, веб-сайти та платформи обміну миттєвими повідомленнями. Інші додатки включають ботів, які використовуються в розважальних цілях, для досліджень, а також соціальних ботів, які рекламують певний продукт, кандидата чи проблему.

Ті, боти які вміють навчатися, спираються на штучний інтелект, щоб спілкуватися з користувачами. Замість заздалегідь підготовлених відповідей, робот відповідає адекватними пропозиціями по темі. До того ж, всі слова, проговорені боту, ботом записуються для подальшої обробки, що дуже зручно. Вже зараз людство створює віртуальних роботів, які за допомогою обробки розмовних мов та штучного інтелекту, спроможні розповісти, як у них справи, розповісти жарт, підтримати розмову.

Робота з обробкою природної мови і машинним навчанням - непросте завдання, особливо для початківців. Останнім часом, у зв'язку з широким розповсюдженням діалогових систем зростає кількість платформ для їх створення.

Чат-боти, зокрема для ВНЗ, буквально зробили революцію в світі вищої освіти, повністю змінюючи спосіб взаємодії студентів зі своїми університетами. Завдяки чат-ботам ефективність освітнього процесу значно зросла. Кожному ВНЗ необхідні якісь інструкції для початківців (абітурієнтів). На початку семестру у кожного абітурієнта та студента виникає питання *«Як доїхати до ВНЗ?»*, *«О котрій годині відкривається*

бібліотека?», «Де будуть заняття по веб-технологіям?». Зазвичай абітурієнти звертаються з такими питаннями до викладачів або, набравшись сміливості, до старших студентів. Це означає, що на початку кожного навчального року у співробітників університетів з'являється безліч додаткових обов'язків, пов'язаних з наданням всієї необхідної інформації і відповідями на однакові питання нових студентів. За останні роки лідируючі ВНЗ світу, такі як *Stanford University, University of Texas at Austin, LMU Munich, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"* та інші, створили свої власні системи чат-ботів для своїх веб-сайтів. Такий чат-бот може служити в якості цілодобової довідкової системи, вартість якої набагато менше, ніж утримання великої кількості додаткових співробітників. Крім того, вони набагато більш доступні, ніж звичайні співробітники, так як студентам не треба шукати їх або стояти в чергах. Все, що їм потрібно, це відкрити веб-сайт університету і набрати своє питання у вікні онлайн-чату. Тому було б доречним створити універсального освітнього чат-бота для ВНЗ СНУ імені Володимира Даля із використанням діалогових систем, який би зміг допомогати абітурієнтам, студентам та викладачам отримувати необхідну інформацію.

Також потрібно відзначити що, незважаючи на досить короткий термін з моменту початку проведення досліджень, значна частина описаних методів оптимізації втратила свою актуальність у зв'язку зі зміною роботи алгоритмів пошукових роботів. Як вже раніше зазначалося, пошукові системи постійно вносять зміни в алгоритм індексування сайтів відповідно до сучасних тенденцій. Таким чином, рекомендації, опубліковані більш ніж рік тому, втрачають свою актуальність, або навпаки вимагають більш докладної оцінки.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконувалася протягом 2017-2018 рр. згідно з планами науково-дослідних робіт кафедри комп'ютерних наук та інженерії в межах НДР «Дослідження у галузі вищої освіти і практики інституційної співпраці» (№ ДР 0113U002236).

Мета і завдання дослідження. Метою магістерської роботи є проектування та реалізація діалогової системи для взаємодії студентів, абітурієнтів та викладачів ВНЗ за допомогою віртуального помічника (чат-бота).

Для досягнення мети в роботі сформульовані і вирішені наступні завдання:

- проаналізовано методи розробки діалогових систем та існуючі програмні засоби, які їх описують;
- визначено основні принципи роботи чат-ботів;
- проведено критичний аналіз статей та публікацій з тематики дослідження;

- проаналізовано сучасні вимоги та зміни в роботі діалогових систем із віртуальними клієнтами;
- проектування та розробка діалогової системи;
- розробка інформаційно-освітнього чат-бота.

Об'єкт проектування – розробка віртуального освітнього чат-бота.

Предмет дослідження - розробка за допомогою спроектованої діалогової системи віртуального освітнього чат-бота.

Методи дослідження. Для вирішення поставлених завдань було застосовано метод розробки нової моделі роботи освітнього чат-боту. Були використані методи веб-хуку, оповіщення про повідомлення, підключення діалогової системи до свого веб-сервісу, який звертається до сховищ даних, реалізує бізнес-логіку, виконує виклики зовнішніх API.

Наукова новизна отриманих результатів. Останнім часом у зв'язку з широким розповсюдженням діалогових систем зростає кількість платформ для їх створення. Наведено порівняння найбільш поширених фреймворків діалогових систем. Вперше було проаналізовано зміни в рейтингах віртуальних чат-ботів, месенджерів які найчастіше використовує молоде покоління. Дістало подальший розвиток дослідження методів розробки нової моделі розробки діалогових систем. Вперше було створено додаток для СНУ ім. В.Даля у вигляді чат-боту у месенджері Telegram.

Практичне значення отриманих результатів. Було створено універсального освітнього чат-бота для ВНЗ СНУ імені Володимира Даля із використанням діалогових систем.

Публікації. По даній роботі було зроблено публікацію:

Конференція "Майбутній науковець" м. Сєверодонецьк Тема доповіді: « Хто такі чат-боти та чого нам чекати в майбутньому? » [1]

Структура і обсяг роботи. Магістерська робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та двох додатків. Робота викладена на 90 сторінках машинописного тексту, містить 77 сторінок основного тексту, 36 рисунків, 2 таблиці, 2 додатки на 16 сторінках. Бібліографічний список включає 25 найменувань.

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ І ЗАСОБІВ РОЗРОБКИ ДІАЛОГОВИХ СИСТЕМ ДЛЯ ВІРТУАЛЬНИХ КЛІЄНТІВ

1.1 Діалогові системи та віртуальні клієнти (чат-боти)

Діалогова система або діалоговий агент - це комп'ютерна система, призначена для спілкування з людиною, з когерентною (послідовною) структурою. Ці програми спілкуються з користувачами на розмовній мові, а також, як правило, діляться на два класи.

1. Діалогові системи, орієнтовані на конкретні завдання, - призначені для конкретного завдання і обробляють короткі бесіди, щоб отримати інформацію від користувача для виконання завдання. Вони включають в себе інтерактивні мобільні цифрові помічники (Siri, Cortana, Alexa, Google Assistant та інші). Такі діалогові системи служать: для виконання інтерактивного пошуку; управління побутовою технікою, системами розумного будинку; допомогою в здійсненні дзвінків або відправлення текстових повідомлень. Крім того, подібні системи служать для інтерактивної допомоги клієнтам на різних веб-сайтах.
2. Чат-боти - це системи, що імітують мовну поведінку людини при тривалому спілкуванні з одним або декількома співрозмовниками, і на відміну від діалогових систем не вирішують якусь конкретну задачу. Ці системи часто мають розважальну функцію.

У свою чергу чат-боти діляться на два класи: системи, засновані на правилах (rulebased systems) і системи на основі корпусів (corpus-based systems).

— Система, заснована на правилах, є відносно простою моделлю, яка може бути адаптована до будь-якої кількості проблем. Як і у випадку з будь-яким штучним інтелектом, система, заснована на правилах, має свої переваги, а також обмеження, які необхідно враховувати перед тим, як вирішити, чи це правильна техніка для даної проблеми. В цілому, системи, засновані на правилах, дійсно можливі лише для проблем, для яких будь-які знання в проблемній області можуть бути написані у вигляді правил «якщо-то», і для яких ця проблемна область невелика. Якщо існує занадто багато правил, система буде важко підтримувати їх і може постраждати від недостатньої продуктивності.

— Системи на основі корпусів здатні імітувати спілкування з живою людиною. Такі системи за допомогою методів пошуку інформації або нейромережових технологій здатні ефективно відображати призначені для користувача висловлювання як відповідь системи. Чат-боти допомагають замовити квиток на потяг, забронювати столик в ресторані або оплатити банківські рахунки в будь-яку годину дня і ночі. Вони живуть в месенджерах та дозволяють бізнесу автоматизувати багато процесів взаємодії з клієнтом.

Віртуальний співрозмовник, програма-співрозмовник, онлайн-співрозмовник або просто чат-бот - вдала імітація спілкування з живою людиною. Вона допомагає інтернет-користувачеві зорієнтуватися і надає відповіді на актуальні питання. Для зв'язку використовуються голосові або текстові повідомлення. Чат-боти створюються на базі таких додатків, як Telegram, FB Messenger, Skype, Viber, WhatsApp, Slack та інші.

Термін віртуальний чат-бот вперше вжив Майкл Маулдін (творець першого Вербота, Julia) у 1994 році, щоб описати ці розмовні програми. Сьогодні чат-боти є частиною віртуальних помічників, таких як Google Assistant, і доступні через програми багатьох організацій, веб-сайти та платформи обміну миттєвими повідомленнями. Інші додатки включають ботів, які використовуються в розважальних цілях, для досліджень та соціальних ботів, які рекламують певний продукт, кандидата чи проблему.

Класичними першими історичними чат-ботами вважаються — ELIZA (1966) і PARRY (1972). Більш нові моделі були: A.L.I.C.E., Jabberwacky і D.U.D.E. Незважаючи на те, що ELIZA і PARRY були використані виключно для імітації набраної розмови, у багатьох чатах з обговоренням зараз входять такі функціональні функції, як ігри та можливості пошуку в Інтернеті. У 1984 році була опублікована книга, що називається *The Policeman's Beard is Half Constructed*, яка нібито була написана чат-ботом Racter (хоча програма, яка була випущена, не могла це зробити).

Одним з найважливіших напрямків дослідження AI є обробка природної мови. Зазвичай більш слабкі штучні інтелекти використовують спеціалізоване програмне забезпечення або мови програмування, створені спеціально для необхідної вузької функції. Наприклад, A.L.I.C.E. використовує мову розмітки, що називається AIML, яка є специфічною для його функції як розмовного агента, і з тих пір вона широко використовується іншими розробниками, так званими Alicebots. Тим не менш, A.L.I.C.E. як і раніше, було побудовано виключно на основі методів узгодження зразків без будь-яких можливостей для обґрунтування, той самий метод ELIZA використовувався ще в 1966 році. Це не слабкий штучний інтелект, який не потребує розумності та логічних розумових здібностей.

1.2 Аналіз сучасних засобів створення віртуальних систем. Python

Python - інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

1.2.1 Історія

Розробка мови Python була розпочата в кінці 1980-х років співробітником голландського інституту CWI Гвідо ван Россумом. Для розподіленої ОС Amoeba потрібна була розширювана скриптова мова, і Гвідо почав писати Python на дозвіллі, запозичивши деякі напрацювання для мови ABC (Гвідо брав участь у розробці цієї мови, орієнтованої на навчання програмування). У лютому 1991 року Гвідо опублікував вихідний текст в групі новин alt.sources. Мова почала

вільно поширюватися через Інтернет і сподобалася іншим програмістам. З 1991 року Python є цілком об'єктно-орієнтованим. Python також запозичив багато рис таких мов, як C, C++, Modula-3 і Icon, й окремі риси функціонального програмування з Ліспу.

Назва мови виникла зовсім не від виду плазунів. Автор назвав мову на честь популярного британського комедійного серіалу 70-х років «Повітряний цирк Монті Пайтона». Втім, все одно назву мови частіше асоціюють саме зі змією, ніж з фільмом — піктограми файлів в KDE або в Windows, і навіть емблема на сайті python.org зображують зміїну голову.

Наявність дружньої спільноти користувачів, поряд з дизайнерською інтуїцією Гвідо, вважається одним з головних факторів успіху Python. Розвиток мови відбувається згідно з чітко регламентованими процесами створення, обговорення, відбору та реалізації документів PEP (Python Enhancement Proposal) — пропозицій щодо розвитку Python.

Python портований і працює майже на всіх відомих платформах — від КПК до мейнфреймів. Існують порти під Microsoft Windows, всі варіанти UNIX (включаючи FreeBSD та GNU/Linux), Plan 9, Mac OS та Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 та навіть OS/390, Symbian та Android.

У міру старіння платформи її підтримка в основній гілці мови припиняється. Наприклад, з версії 2.6 припинена підтримка Windows 95, Windows 98 та Windows ME. Однак на цих платформах можна використовувати попередні версії Python — спільнота активно підтримує версії Python починаючи від 2.3 (для них виходять виправлення).

При цьому, на відміну від багатьох портованих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Навіть більше, існує спеціальна версія Python для віртуальної машини Java — Jython, що дозволяє інтерпретатору виконуватися на будь-якій системі, яка підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python й навіть бути написаними на ньому. Також кілька проектів забезпечують інтеграцію з платформою Microsoft.NET, основні з яких — IronPython та Python.Net.

1.2.2 Переваги

Серед основних її переваг можна назвати такі:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм (що властиве більшості інтерпретованих мов);
- стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);

- можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написано на мові Python;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- відкритий код (можливість редагувати його іншими користувачами).

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ. Інтерпретатор мови Python і багата Стандартна бібліотека (як вихідні тексти, так і бінарні дистрибутиви для всіх основних операційних систем) можуть бути отримані з сайту Python www.python.org, і можуть вільно розповсюджуватися. Цей самий сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

1.2.3 Типи й структури даних

Python підтримує динамічну типізацію, тобто, тип змінної визначається лише під час виконання. З базових типів слід зазначити підтримку цілих чисел довільної довжини і комплексних чисел. **Python** має багату бібліотеку для роботи з рядками, зокрема, кодованими в юнікодi. З колекцій Python підтримує кортежі (tuples), списки (масиви), словники (асоціативні масиви) і від версії 2.4, множини. Система класів підтримує множинне успадкування і метапрограмування. Будь-який тип, включаючи базові, входить до системи класів, й за необхідності можливе успадкування навіть від базових типів.

Багата стандартна бібліотека є однією з привабливих сторін Python. Тут є засоби для роботи з багатьма мережевими протоколами та форматами Інтернету, наприклад, модулі для написання HTTP-серверів та клієнтів, для розбору та створення поштових повідомлень, для роботи з XML, тощо. Набір модулів для роботи з операційною системою дозволяє писати крос-платформні застосунки. Існують модулі для роботи з регулярними виразами, текстовими кодуваннями, мультимедійними форматами, криптографічними протоколами, архівами, серіалізацією даних, юніт-тестуванням та ін.

1.2.4 Недоліки

1. Низька швидкодія

Python, як і багато інших інтерпретованих мов, які не застосовують, наприклад, JIT-компілятори, мають загальний недолік — порівняно невисоку швидкість виконання програм. Однак, у випадку з **Python** цей недолік компенсується зменшенням часу розробки програми. У середньому, програма, написана на **Python**, в 2-4 рази компактніша, ніж її аналог на C++ або Java. Збереження байт-коду (файли .pyc і .pyo) дозволяє інтерпретатору не витратити зайвий час на перекомпіляцію коду модулів при кожному запуску, на відміну, наприклад, від мови Perl. Крім того, існує спеціальна JIT-бібліотека pycos (проте призводить до збільшення споживання оперативної пам'яті). Ефективність pycos в значній мірі залежить від архітектури програми. Існують проекти реалізацій мови **Python**, що вводять високопродуктивні віртуальні машини (VM) як компілятора заднього плану. Прикладами таких реалізацій може служити PyPy, що базується на LLVM; більш ранньою ініціативою є проект Pango. Очікується, що використання VM типу LLVM призведе до тих самих результатів, що й використання аналогічних підходів для реалізацій мови Java, де низька обчислювальна продуктивність в основному подолана.

2. Відсутність статичної типізації

Відсутність статичної типізації є не стільки вадю інтерпретатора, скільки вибором розробника мови. Справа в тому, що в **Python** прийнята так звана «качина типізація». У силу цього типи переданих значень недоступні на етапі компіляції, та помилки на зразок `AttributeError` можуть виникати під час виконання. Відсутність статичної типізації також є однією з основних причин низької швидкодії. Існують модулі, які дозволяють контролювати типи параметрів функцій на етапі виконання, наприклад, `typecheck` або `method signature checking decorators`. Додавання необов'язкової статичної типізації параметрів функції заплановано для Python3000. При цьому, однак, безпосередньо інтерпретатор не буде перевіряти типи, а тільки додавати відповідну інформацію до метаданих функції для її (інформації) подальшого використання модулями розширень. Відсутність статичної типізації і деякі інші причини не дозволяють реалізувати в Python механізм перевантаження функцій на етапі компіляції. Можливості Python дозволяють реалізувати динамічну перевантаження на

етапі виконання, що, звичайно, уповільнює виклик, так як вирішення яку саме функцію викликати проводиться при кожному зверненні і є, в загальному випадку, досить складною процедурою. Відсутність перевантаження в Python намагаються компенсувати використанням віртуальних функцій.

3. Неможливість модифікації вбудованих класів

У порівнянні з Ruby та деякими іншими мовами, в Python відсутня можливість модифікувати вбудовані класи, такі, як `int`, `str`, `float`, `list` та інші, що, однак, дозволяє Python споживати менше оперативної пам'яті і швидше працювати. Ще однією причиною введення такого обмеження є необхідність узгодження з модулями розширення. Багато модулів (з метою оптимізації швидкодії) перетворюють Python-об'єкти елементарних типів до відповідних C-типів замість маніпуляцій з ними за допомогою C-API.

1.2.5 Застосування

Python — стабільна та поширена мова. Вона використовується в багатьох проектах та в різних якість: як основна мова програмування або для створення розширень та інтеграції додатків. На Python реалізована велика кількість проектів, також вона активно використовується для створення прототипів майбутніх програм та чат-ботів. Python використовується в багатьох великих компаніях.

1.3 Аналіз сучасних засобів створення віртуальних систем. Java

Java - об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи **Java**. З 2009 року мовою займається компанія «Oracle», яка того ж року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Мова в значній мірі запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні,

наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

1.3.1 Історія

Мова програмування **Java** зародилася в 1991 р. в лабораторіях компанії Sun Microsystems. Розробку проекту започаткував Джеймс Гослінг, сам проект мав назву «Green» (Зелений). Створення першої робочої версії, яка мала назву «Oak» (дуб), зайняло 18 місяців. Оскільки виявилось, що ім'я Oak уже використовувалось іншою фірмою, то в результаті тривалих суперечок навколо назви нової мови з-поміж ряду запропонованих було вибрано назву Java, у 1995 р. мову було офіційно перейменовано. Головним мотивом створення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, що вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

Досить скоро майже всі найпопулярніші тогочасні веб-оглядачі отримали можливість запускати «безпечні» для системи Java-аплети всередині веб-сторінок. У грудні 1998 р. Sun Microsystems випустила Java 2 (спершу під назвою J2SE 1.2), де було реалізовано декілька конфігурацій для різних типів платформ. Наприклад, J2EE призначалася для створення корпоративних застосунків, а значно урізана J2ME для приладів з обмеженими ресурсами, таких як мобільні телефони. У 2006 році в маркетингових цілях версії J2 було перейменовано у Java EE, Java ME та Java SE відповідно. 13 листопада 2006 року Sun випустили більшу частину Java як вільне та відкрите програмне забезпечення згідно з умовами GNU General Public License (GPL). 8 травня 2007 корпорація закінчила процес, в результаті якого всі початкові коди Java були випущені під GPL, за винятком невеликої частини коду, на який Sun не мала авторського права. Під «незалежністю від архітектури» мається на увазі те, що програма, написана на мові Java, працюватиме на будь-якій підтримуваній апаратній чи системній платформі без змін у початковому коді та перекомпіляції. Цього можна досягти, компілюючи початковий Java код у байт-код, який є спрощеними машинними командами. Потім програму можна виконати на будь-якій платформі, що має встановлену віртуальну машину Java, яка інтерпретує байткод у код, пристосований до специфіки конкретної операційної системи і процесора. Зараз віртуальні машини **Java** існують для більшості процесорів і операційних систем. Стандартні бібліотеки забезпечують загальний спосіб доступу до таких платформозалежних особливостей, як обробка графіки, багатопотоковість та роботу з мережами. У деяких версіях задля збільшення

продуктивності JVM байт-код можна компілювати у машинний код до або під час виконання програми.

1.3.2 Переваги

Основна перевага використання байт-коду — це портативність. Тим не менш, додаткові витрати на інтерпретацію означають, що інтерпретовані програми будуть майже завжди працювати повільніше, ніж скомпільовані у машинний код, і саме тому Java одержала репутацію «повільної» мови. Проте, цей розрив суттєво скоротився після введення декількох методів оптимізації у сучасних реалізаціях JVM. На противагу C++, **Java** об'єктно-орієнтованіша. Всі дані і дії групуються в класи об'єктів. Виключенням з повної об'єктності (як скажімо в Smalltalk) є примітивні типи (int, float тощо). Це було свідомим рішенням проєктувальників мови задля збільшення швидкості. Через це Java не вважається повністю об'єктно-орієнтовною мовою. У Java всі об'єкти є похідними від головного об'єкта (він називається просто Object), з якого вони успадковують базову поведінку і властивості.

1.3.3 Недоліки

1. Платне комерційне використання

Нещодавно Oracle оголосила, що з 2019 року компанія почне стягувати плату за використання Java Standard Edition 8 в «комерційних цілях». За все нові оновлення та виправлення помилок доведеться заплатити. Плата залежить від кількості користувачів або комп'ютерів. Поточна версія Java безкоштовна для простого використання. Таким чином, кожна використовуюча Java компанія повинна оцінити, наскільки ефективно вона використовує цю мову. Компанія повинна зрозуміти, що вигідніше: шукати альтернативне рішення або продовжувати користуватися Java.

2. Низька продуктивність

У будь-якої мови високого рівня досить низька продуктивність через компіляції та абстракції за допомогою віртуальної машини. Однак це не єдина причина низької швидкості Java. Наприклад, додаток очищення пам'яті: це корисна функція, яка, на жаль, призводить до значних проблем з продуктивністю, якщо вимагає більше 20 відсотків часу процесора. Погана настройка кешування може викликати надмірне використання пам'яті. Існує також взаємне блокування потоків: так відбувається, коли кілька потоків намагаються отримати доступ до

одного і того ж ресурсу. В цьому випадку відбувається кошмар кожного Java-розробника - помилка через брак пам'яті. Проте вміле планування може вирішити всі ці проблеми.

3. Відсутність нативного дизайну

Для створення графічного інтерфейсу користувача (GUI) розробники використовують різні інструменти, орієнтовані для конкретної мови. Для Android-додатків є Android Studio, яка допомагає створювати додатки з нативним дизайном. Однак, коли справа доходить до призначеного для користувача інтерфейсу на ПК, Java-інструменту для створення нативного дизайну немає. Є кілька інструментів для розробки GUI для Java: найпопулярніші з них - Swing, SWT, JavaFX, JSF. Бібліотека Swing - це старий, але надійний крос-платформний інструмент, інтегрований в різні Java-IDE, в тому числі Eclipse і NetBeans. Однак, якщо ви не використовуєте шаблони, ви помітите невідповідності інтерфейсу. SWT використовує власні компоненти, але не підходить для складного інтерфейсу. JavaFX - лаконічний і сучасний, але занадто новий. В цілому, перед створенням GUI на Java потрібно докладніше вивчити інструменти.

4. Багатослівний і складний код

Багатослівність коду може здатися перевагою, яке допоможе при вивченні мови. Однак, довгі, надмірно складні речення ускладнюють читання і перегляд коду. Як і природні мови, багатьма мовами програмування високого рівня містять зайву інформацію. Java - це легша версія неприступного C ++, яка змушує програмістів прописувати свої дії словами з англійської мови. Це робить мову більш зрозумілим для неспеціалістів, але менш компактним.

5. Застосування

Більшість організацій так чи інакше використовують Java. Широкий спектр варіантів використання Java робить її практично непомітною у використанні: тому часто виникає питання «де використовується Java?».

1.4 Архітектура на основі фреймів

Більшість діалогових систем, орієнтованих на конкретні завдання, засновані на архітектурі на основі фреймів (frame-based architecture).

Фрейм (frame) - набір слотів, і кожен слот може приймати одне конкретне значення з фіксованого набору. Сучасні діалогові системи, орієнтовані на завдання, засновані на онтології предметної області - структурі знань, представляє види намірів, які система може витягувати з призначених для користувача пропозицій. Онтологія визначає один або кілька фреймів, кожен з яких представляє собою набір слотів, і визначає значення, які може приймати кожен слот. Набір слотів вказує, що система повинна знати, і наповнення кожного слота обмежена значеннями певного семантичного типу.

Наприклад, в області туризму слот може мати тип city (місто), тип date (дата) або time (час). Типи в сучасних фреймових діалогових агентів, можуть мати ієрархічну структуру; наприклад, тип дати сам є фреймзі слотами таких типів, як integer або назви днів тижня. Архітектура управління цими діалоговими системами розроблена навколо фреймів. Мета полягає в тому, щоб заповнити слоти в кадрі, де допомогою наповнювачів, які мав на увазі користувач, а потім виконати відповідну дію для користувача. Один з компонентів діалогових систем на основі фреймів - компонент розуміння природної мови. **Мета** даного компонента - витягти три речі з висловлювання користувача.

1.5 Нейронні мережі

Область машинного навчання і штучного інтелекту в наші дні набула широкого поширення. **Штучний інтелект** (ШІ, англ. Artificial intelligence, AI) – це великий міждисциплінарний напрямок, завданням якого є моделювання і формалізація інтелектуальних дій людини.

1.5.1 Середовище застосування нейронних мереж

Виконання обчислювальних завдань і завдань аналізу даних є великим додатком комп'ютерних наук.

Такі завдання умовно можна класифікувати за рівнями складності:

- Завдання, які вирішуються алгоритмічно.
- Завдання, рішення яких вимагає неглибокого аналізу оброблюваних
- Завдання, для вирішення яких необхідний глибокий аналіз складних

Для задач, що входять в третю категорію, застосування нейромереж найефективніше. Нейронні мережі мають здатність глибокого аналізу даних, що обробляються, виявлення складних нелінійних залежностей, виявлення нечітких взаємозв'язків. Так як розробляється

система передбачає розпізнавання запитів природною мовою і розпізнавання голосу, то для вирішення даного завдання найкращим чином підходить використання нейронних мереж.

1.5.2 Типи систем з інтелектуальним інтерфейсом

Застосування штучного інтелекту для посилення комунікативних здібностей інформаційних систем призвело до появи систем з інтелектуальним інтерфейсом, серед яких можна виділити наступні типи:

1. Інтелектуальні бази даних. Дозволяють на відміну від традиційних БД забезпечувати вибірку необхідної інформації, що не присутньої в явному вигляді, а виводиться з сукупності збережених даних.
2. Розмовний інтерфейс. Застосовується для доступу до інтелектуальним баз даних, контекстного пошуку документальної текстової інформації, голосового введення команд в системах управління, машинного перекладу з іноземних мов.

Для реалізації ПМ-інтерфейсу необхідно вирішити проблеми **морфологічного, синтаксичного і семантичного аналізу**, а також завдання синтезу висловлювань на розмовній мові.

При морфологічному аналізі здійснюються розпізнавання і перевірка правильності написання слів в словнику.

Синтаксичний контроль передбачає розкладання вхідних повідомлень на окремі компоненти, перевірку відповідності граматичним правилам внутрішнього представлення знань і виявлення відсутніх частин.

Семантичний аналіз забезпечує встановлення смислової правильності синтаксичних конструкцій. На відміну від аналізу синтез висловлювань полягає в перетворенні цифрового подання інформації в уявлення природною мовою.

1.5.3 Гіпертекстові системи

Використовуються для реалізації пошуку по ключовим словам у базах даних з текстовою інформацією. Для більш повного відображення різних смислових відносин термінів потрібна складна семантична організація ключових слів. Вирішення цих завдань здійснюється за допомогою інтелектуальних гіпертекстових систем, в яких механізм пошуку спочатку працює з базою знань ключових слів, а потім - з самим текстом. Аналогічним чином проводиться пошук мультимедійної інформації, що включає крім тексту графічну інформацію, аудіо- та відеообрази.

1.5.4 Системи контекстної допомоги

Відносяться до класу систем поширення знань. Такі системи є, як правило, додатками до документації. Системи контекстної допомоги - окремий випадок гіпертекстових і ПМ-систем. У них користувач описує проблему, а система на основі додаткового діалогу конкретизує її і виконує пошук відносяться до ситуації рекомендацій. У звичайних гіпертекстових системах, навпаки, комп'ютерні програми нав'язують користувачеві схему пошуку необхідної інформації.

1.5.5 Системи когнітивної графіки

Орієнтовані на спілкування з користувачем ІВС за допомогою графічних образів, які генеруються відповідно до змін параметрів модельованих або процесів, що спостерігаються. когнітивна графіка дозволяє в наочному і виразному вигляді уявити безліч параметрів, що характеризують досліджуване явище, звільняє користувача від аналізу тривіальних ситуацій, сприяє швидкому освоєнню програмних засобів і підвищенню конкурентоспроможності розроблюваних ІВС. Застосування когнітивної графіки особливо актуально в системах моніторингу та оперативного управління, в навчальних і тренажерних системах, в оперативних системах прийняття рішень, які працюють в режимі реального часу.

Системи штучного інтелекту повинні виконувати такі дії, які зазвичай вважаються властивими для людини: упорядкування наявних знань, формування логічного і непевного виведення, побудова міркувань, класифікація інформації, прийняття рішень і т.д.

1.6 Машинне навчання

Машинне навчання (англ. Machine learning) є область комп'ютерних наук, що включає методи, які на основі аналізу оброблюваних даних підвищують показник якості свого функціонування за рахунок відповідного налаштування своїх внутрішніх параметрів.

1.6.1 Коротка історія та великі надії

Першу програму на основі алгоритмів, здатних самонавчатися, розробив Артур Самуель (Arthur Samuel) в 1952 році, призначена вона була для гри в шашки. Самуель дав і перше визначення терміну «машинне навчання»: це «область досліджень розробки машин, які не є

задалегідь запрограмованими». Вже в 1957 році була запропонована перша модель нейронної мережі, що реалізує алгоритми машинного навчання, схожі на сучасні. В даний час ведеться розробка самих різних систем машинного навчання, призначених для використання в таких технологіях майбутнього, як Інтернет Речей, Промисловий Інтернет Речей, в концепції «розумний» місто, при створенні безпілотного транспорту і в багатьох інших.

Про те, що на машинне навчання зараз покладають великі надії, свідчать такі факти.

- В компанії Google вважають, що скоро її продукти «перестануть бути результатом традиційного програмування - в їх основу буде покладено машинне навчання».
- Компанії Google, Facebook, Apple, Amazon, Microsoft і китайська фірма Baidu вступили в боротьбу за талановитих фахівців у сфері штучного інтелекту.
- Марк Цукерберг, генеральний директор Facebook, особисто - по телефону і по відеочату - бере участь в спробах його компанії переманити найкращих випускників.
- Відвідуваність на найважливіших академічних конференціях в цій сфері збільшилася майже в чотири рази.
- Так нові продукти, як Siri від Apple, M від Facebook, Echo від Amazon були створені за допомогою машинного навчання.

1.7 Штучні нейронні мережі

Штучні нейронні мережі (ШНМ; англ. Artificial neural networks, ANN) - область машинного навчання, що займається вирішенням завдань на основі математичних моделей організації і функціонування біологічних нейронних систем.

Для машинного навчання використовують різні технології та алгоритми. Зокрема, можуть застосовуватися дискримінантний аналіз, байєсовські класифікатори та багато інших математичних методів. Але в кінці XX століття все більше уваги почали приділяти штучним нейронним мережам (ANN). Черговий вибух інтересу до них почався в 1986 році, після істотного розвитку т.зв. «Методу зворотного поширення помилки», який з успіхом застосували при навчанні нейронної мережі. ANN є системою з'єднаних і взаємодіючих між собою штучних нейронів, виконаних на основі порівняно простих процесорів. Кожен процесор ANN періодично отримує сигнали від одних процесорів (або від сенсорів, або від інших джерел сигналів) і періодично посилає сигнали іншим процесорам. Всі разом ці прості процесори, з'єднані в мережу, здатні вирішувати досить складні завдання. Найчастіше нейрони розташовуються в мережі за рівнями (їх ще називають шарами). Нейрони першого рівня - це, як правило, вхідні. Вони отримують дані ззовні (наприклад, від сенсорів системи розпізнавання осіб) і після їх

обробки передають імпульси через синапси нейронів на наступному рівні. Нейрони на другому рівні (його називають прихованим, оскільки він безпосередньо не пов'язаний ні з входом, ні з виходом ANN) обробляють отримані імпульси і передають їх нейронам на вихідному рівні.

Оскільки мова йде про імітацію нейронів, то кожен процесор вхідного рівня пов'язаний з декількома процесорами прихованого рівня, кожен з яких, в свою чергу, пов'язаний з декількома процесорами рівня вихідного. Така архітектура найпростішої ANN, яка здатна до навчання і може знаходити прості взаємозв'язку в даних.

Глибоке (глибинне) навчання може бути застосоване лише по відношенню до більш складних ANN, що містить кілька прихованих рівнів. При цьому рівні нейронів можуть чергуватися з шарами, які виконують складні логічні перетворення. Кожен наступний рівень мережі шукає взаємозв'язки в попередньому. Така ANN здатна знаходити не тільки прості взаємозв'язки, а й взаємозв'язки між взаємозв'язками. Саме завдяки переходу на нейромережу з глибинним навчанням компанії Google вдалося різко підвищити якість роботи свого популярного продукту «Перекладач». Зокрема, якість перекладу між англійською та французькою мовами підвищився відразу на 7 балів, тобто більш ніж на 20%. Попередня система, яка виконувала фразовий статистичний машинний переклад, домоглася подібного поліпшення за весь час свого існування (з 2006 року)

На рис. 1.1 зображено зв'язок таких понять, як **штучний інтелект**, **машинне навчання**, **штучні нейронні мережі** в загальному вигляді.

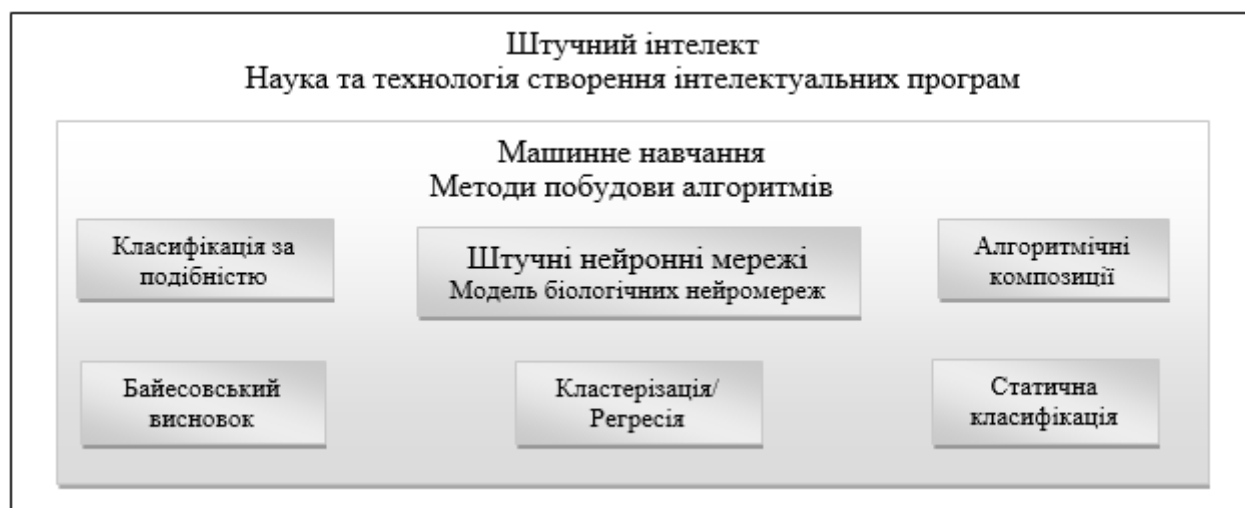


Рисунок 1.1 Зв'язок понять ІІ, МО та ІНС

1.8 Аналіз платформ для створення діалогових систем

У зв'язку з широким розповсюдженням діалогових систем останнім часом зростає кількість платформ для їх створення. Наведемо порівняння найбільш поширених фреймворків діалогових систем.

1.8.1 Rasa

Rasa є провідним інструментальним засобом для машинного навчання з відкритим вихідним кодом, який дозволяє розробникам розширювати боти, виходячи з простих питань. На даний час цієї платформою користуються такі компанії як B.M.W, Raiffeisen, АХА, Parallon. Rasa NLU - це інструмент для обробки природних мов з відкритим вихідним кодом для класифікації цілей та вилучення об'єктів у чат-боти.

1.8.2 Microsoft Bot Framework

Microsoft Bot Framework є комплексне фреймворк для створення і розгортання високоякісних спамерських пошукових роботів. Він має власний SDK (software development kit - набір засобів розробки, який дозволяє створювати додатки для певного пакета програм) Bot Builder. Вся система складається з трьох частин: Bot Connector, Portal Developer і Bot Directory. Структура надає можливість для розміщення бота в додатку або на веб-сайті. Microsoft Bot Framework здатний розпізнавати наміри того, що сказав користувач. Крім того, можна підключити модуль LUIS для розуміння природної мови, модуль Cortana для обробки голосу і Bing API для виконання пошуку. Однак, даний фреймворк має ряд обмежень: складність інтеграції бота в деякі месенджери, широко використовувані в Росії, обмежений безкоштовний період використання в 30 днів.

1.8.3 Wit.ai

Wit.ai - безкоштовний сервіс від facebook для обробки природного мови і для створення ботів зі штучним інтелектом, який приймає на вхід текстову або голосову фразу природною мовою і формує відповідь за допомогою нейромережі. Платформа дозволяє використовувати суті, наміри, контексти і дії і включає обробку природної мови. Даний сервіс має обмеження: можливі проблеми з інтеграцією до деяких месенджерів.

1.8.4 Semantic Machines

Компанія Semantic Machines розробила пропріетарного розмовного інтелектуального агента. Дана платформа має наступні можливості: механізм бесіди, синтез мови, глибоке навчання, розпізнавання мови, витяг семантичного наміри, технологію генерації мови та інші. Складність використання даної платформи в тому, що Semantic Machines публічно не розголошує умови використання. На офіційному сайті можна помітити, що цю платформу використовують зазвичай у великих компаніях.

1.8.5 Chatfuel

Chatfuel - це один з популярних бот-конструкторів, який не вимагає навичок програмування. Основний інструмент побудови бота - блок. Він складається з однієї або декількох карток повідомлень, які відправляються разом з користувачем бота. На наступному кроці відбувається зв'язування блоків один з одним за допомогою кнопок в текстових або карткових галереях. У Chatfuel було впроваджено безліч плагінів: пошук Google, пошук Bing, API JSON, призначений для користувача введення та інші.

1.8.6 DialogFlow

Інша безкоштовна діалогова платформа для розробки діалогових систем для додатків, веб-сайтів, платформ обміну повідомленнями і пристроїв – DialogFlow. За допомогою методів машинного навчання DialogFlow розпізнає намір того, що сказав користувач і дозволяє діалоговій системі надавати точні відповіді. Розпізнавання природної мови працює на основі Google's Machine Learning і крім наміри витягує з повідомлення користувача необхідні параметри. DialogFlow перетворює текст запиту в керовані дані і повертає вихідні дані як об'єкт відповіді JSON. Сервіс активно розвивається, має зрозуміле API, підтримує російську мову.

1.9 Висновки до 1 розділу

У цьому розділі були розглянуті такі поняття предметної області, як діалогова система і чат-бот, штучний інтелект, машинне навчання та нейронні мережі. Таким чином, у завдання даної дослідницької роботи входить прототипування та дослідження діалогових систем за допомогою засобів нейромережевого програмування та машинного навчання.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ДІАЛОГОВИХ СИСТЕМ ДЛЯ ВІРТУАЛЬНИХ КЛІЄНТІВ

2.1 Вибір діалогової системи

Задовго до того, як були введені терміни розмовний інтерфейс або чат-бот, Тьюрінг (1950) описав їх як кінцевий тест для штучного інтелекту. Незважаючи на свою давню історію, останнім часом існує ажіотаж з приводу чат-ботів в різних областях (див., наприклад, Ferrara et al., 2016 і промисловість Gartner, 2016).

Хоча існує багато відповідних причин для такого розвитку, вважається, що три ключові зміни це:

- Зростання універсальних платформ для спілкування (наприклад Telegram, Facebook Messenger, Slack та ін.).
- Досягнення в машинному навчанні (ML).
- Розуміння природних мов (NLU), як сервіс.

На момент написання дипломного проекту серед найбільш популярних послуг NLU (або діалогових систем) є:

- LUIS;
- Watson Conversation;
- API.ai (DialogFlow);
- wit.ai;
- Amazon Lex;

Розберемо кожну діалогову систему далі окремо.

2.1.1 LUIS - Language Understanding Intelligent Service (рис. 2.1)



Рисунок 2.1 – Логотип LUIS

Майже 41% підприємств у дослідженні Mindbrowser показав, що вони віддають перевагу Microsoft Bot Framework. Він має свій власний SD Builder Bot, який включає в себе .NET SDK і Node.js SDK. Вся система складається з трьох частин: Bot Connector, Portal для розробників і Bot Directory. Структура забезпечує API прямого доступу REST, яка може бути використана для розміщення бота в додатку або веб-сайті. Він є відкритим кодом і доступний для всіх на Github, і підтримує автоматичний переклад на більш ніж 30 мов. Microsoft Bot Framework розуміє наміри користувачів. Можна включити **LUIS** для розуміння природної мови, Cortana для голосу та API Bing для пошуку. Microsoft Language Understanding Intelligent Service (LUIS) використовує цілі та об'єкти. Усі його програми зосереджені на конкретній темі або стосуються змісту. Технологія активного навчання є однією з можливостей LUIS. Можна використовувати попередньо побудовані моделі від Bing і Cortana. Розгортання моделей до кінцевої точки HTTP - це операція одним клацанням миші; вона повертає легкий у використанні JavaScript Object Notation (JSON). LUIS пропонує набір програмних REST API, які можуть бути використані розробниками для автоматизації процесу створення додатків. Крім того, доступні кілька SDK: C # SDK, Python SDK, Node JS SDK і Android SDK. Підтримувані мови - англійська, французька, італійська, німецька, іспанська, бразильська португальська, японська, корейська та китайська. LUIS пропонує безкоштовні і стандартні плани.

2.1.2 IBM Watson (рис. 2.2)



Рисунок 2.2 – Логотип IBM Watson

За даними дослідження Mindbrowser спільно з журналом Chatbots Journal IBM Watson є першим вибором як платформа для створення ботів для 61% підприємств. Однією з найважливіших частин Уотсона є служба розмов. Він побудований на нейронній мережі (один мільярд слів Вікіпедії), розуміє наміри, інтерпретує об'єкти та діалоги, підтримує англійську та японську мови, а також надає засоби розробки, такі як Node SDK (Software Development Kit), SDK для Java, SDK для Python, SDK для iOS та Unity SDK. IBM пропонує безкоштовні,

стандартні та преміальні плани. Королівський банк Шотландії запустив Luvo, **чат-бот** навчав відповідати на питання клієнтів у реальному часі. Він направляє надто складні запити до консультантів, які використовують IBM Watson Conversation. Типові питання, на які вона відповідає персоналом через веб-чат, включають: Я втратив картку - які кроки потрібно зробити зараз? Я заблокував свій PIN-код - як його розблокувати? Як замовити карт-рідер для свого бізнесу?

2.1.3 Dialog Flow (рис. 2.3)



Рисунок 2.3 – Логотип Dialogflow

Dialogflow - платформа для створення діалогового інтерфейсу. З її допомогою ми виробляємо обробку природної мови в мобільних і веб-додатках, пристроях IoT, чат-ботах. За допомогою Dialogflow додаток або бот надає методи введення і виведення і реагує на які спонукають до дій дані. Також, можна використовувати webhook, щоб підключити Dialogflow до свого веб-сервісу, який, внаслідок, буде звертатися до сховищ даних, реалізувати бізнес-логіку, виконувати виклики зовнішніх API. Беручи запит у вигляді тексту на природній мові або нікого події, Dialogflow погоджує запит з найбільш підходящим шаблоном. При цьому він ґрунтується на інформації, що міститься в шаблоні (приклади, сутності, контекст, параметри) і машинному навчанні. Dialogflow формує відповідний запит і повертає дані у вигляді об'єкта відповіді JSON.

Особливості платформи:

- Дозволяє створити і впровадити чат-бота всього за кілька днів, і вдосконалювати його потім в процесі спілкування з користувачами.
- Інструменти розуміння природної мови допомагають створювати унікальні сценарії діалогів, з відповідними діями, і аналізувати взаємодію з користувачами.
- Зумовлені пакети знання, зібрані за кілька років роботи, включають енциклопедичні дані, новини, замовлення квитків, прогнози погоди, і інше.
- Машинне навчання на прикладах, наданих розробниками, і на спілкуванні з користувачами дозволяє постійно вдосконалювати додаток.
- Інтегрується з Alexa, Cortana і основними платформами для обміну повідомленнями (Skype, Slack, Facebook Twilio SMS, Cisco Spark і іншими).

- Непомітно перемикається між темами розмовами, але, при цьому, запам'ятовує, на чому ви зупинилися в кожному випадку.
- Надає SDK для підтримки кілька платформ і мов програмування, включаючи iOS, Apple Watch і Mac OS X, Android, HTML, Cordova, JavaScript, Python, Ruby, C #, Xamarin і Unity.
- Підтримує 14 мов, включаючи російську та українську.
- Кожен день обробляє мільйони запитів користувачів при 99.99% гарантованого аптайма.

2.1.4 WIT.AI (рис. 2.4)



Рисунок 2.4 – Логотип WIT.AI

Майже 45% респондентів у дослідженні Mindbrowser надали дані, що вони довіряють Wit.ai більше, ніж будь-яка інша платформа для будівництва ботів. До квітня 2017 року її використовувало більше 100 000 розробників. Wit.ai дозволяє використовувати об'єкти, наміри, контексти і дії, а також включає в себе обробку природних мов (НЛП). Є кілька клієнтів: Node.js, Python, Ruby і HTTP API для інших платформ. Він доступний для розробників для використання з iOS, Android, Windows Phone, Raspberry Pi, Python, C і Rust; він також має модуль JavaScript. Wit.ai підтримує близько 50 мов, і це безкоштовно.

2.1.5 Amazon Lex (рис. 2.5)

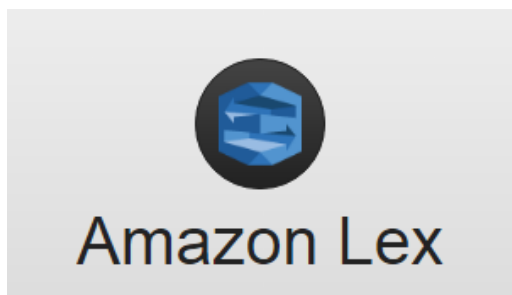


Рисунок 2.5 – Логотип Amazon Lex

Amazon Lex - сервіс для створення голосових і текстових діалогових інтерфейсів в будь-яких додатках. Amazon Lex надає розширені функціональні можливості глибокого навчання, такі як автоматичне розпізнавання мови (ASR), призначене для перетворення мови в текст, і розуміння природних мов (NLU), призначене для визначення сенсу тексту. Це дозволяє створювати привабливі для користувачів додатка з можливостями природного діалогової взаємодії. Використання Amazon Lex відкриває кожному розробнику можливість використання технологій глибокого навчання, на базі яких працює Alexa, що дозволяє з легкістю створювати складні боти для взаємодії на природній мові (чатбот). Сервіс почав розвиватись дуже стрімко і вже зараз вони надають свої сервіси у користування провідним компаніям світу таким як NASA. Також надають «GrowthBot від HubSpot - це універсальний чатбот, який допомагає маркетологам і співробітникам, які працюють в сфері продажів, працювати більш ефективно, надаючи доступ до необхідних даних і сервісів за допомогою діалогового інтерфейсу. GrowthBot може допомогти маркетологам при створенні контенту, вивченні конкурентів і моніторингу аналітичних результатів. Використовуючи Amazon Lex, ми додаємо складні можливості обробки природної мови, які дозволяють GrowthBot надавати нашим користувачам інтуїтивно зрозумілий інтерфейс. Amazon Lex дозволяє нам використовувати всі переваги покращеного штучного інтелекту і машинного навчання без необхідності створювати алгоритми самостійно.

На рис. 2.6 зображено таблицю порівняння основної функціональності, що пропонується різними службами. Всі вони, за винятком Amazon Lex, поділяють одну і ту ж основну концепцію: на основі прикладних даних користувач може навчити класифікатор класифікувати так звані наміри (які представляють намір всього повідомлення і не пов'язані з певною позицією в межах повідомлення) і об'єктів (які можуть складатися з одного або декількох символів).

NLU	Наміри	Об'єкти	Пакетний імпорт
LUIS	+	+	+
Watson	+	+	+
API.ai	+	+	+
wit.ai	+	+	0
Lex	+	0	-

Рисунок 2.6 – Порівняння основних функціональностей діалогових систем

У якості діалогової системи було обрано платформу **Dialogflow**, так як вона безкоштовна для розробки діалогових систем для додатків, веб-сайтів, платформ обміну повідомленнями і пристроїв, а також Інструменти розуміння природної мови допомагають створювати унікальні сценарії діалогів, з відповідними діями, і аналізувати взаємодію з користувачами.

2.2 Архітектура чат-боту

Для того, щоб зрозуміти як працює чат бот, потрібно спочатку поглянути на загальну архітектуру. Універсальна схема роботи чат-боту показана на рис. 2.7.

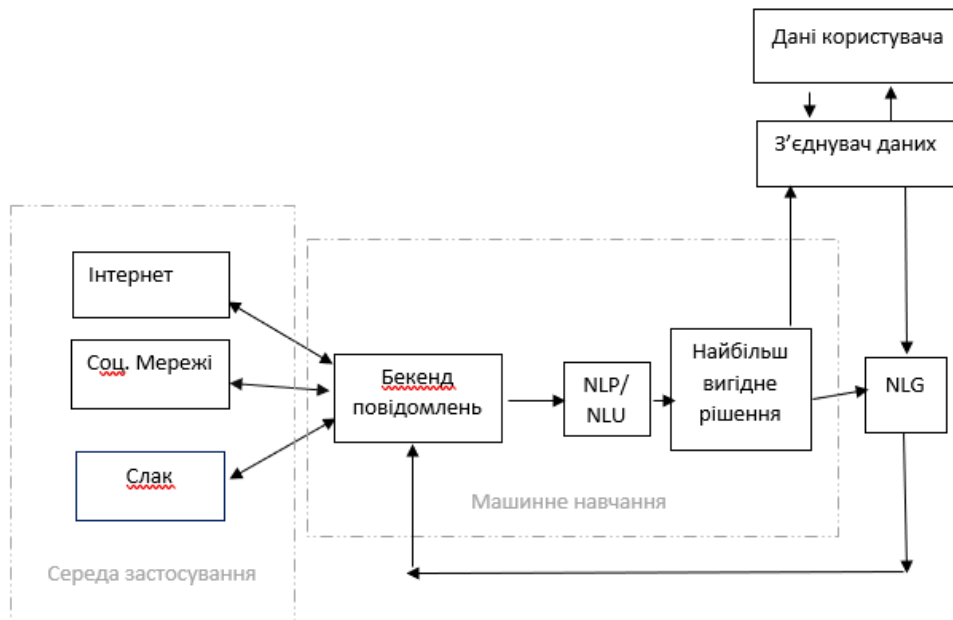


Рисунок 2.7 – Схема роботи чат-боту

Обробка природних мов (NLP) використовується для опису здатності машини приймати те, що сказано, розібрати, зрозуміти його значення, визначити відповідні дії та відповісти на мові, яку користувач зрозуміє.

Розуміння мови (NLU) - це частина обробки природної мови, що стосується набагато вужчого, але не менш важливого аспекту того, як найкраще справлятися з неструктурованими входами та перетворювати їх у структуровану форму, яку машина може зрозуміти та відтворити. Генерація природної мови (NLG) – перетворення структурованих даних в текст.

Мета розуміння мови (NLU) - це є вилучення структурованої, семантичної інформації з неструктурованого вкладу природної мови, наприклад, повідомлення чату. Вони, головним чином, роблять це, додаючи визначені користувачем позначки до повідомлень або частин повідомлень.

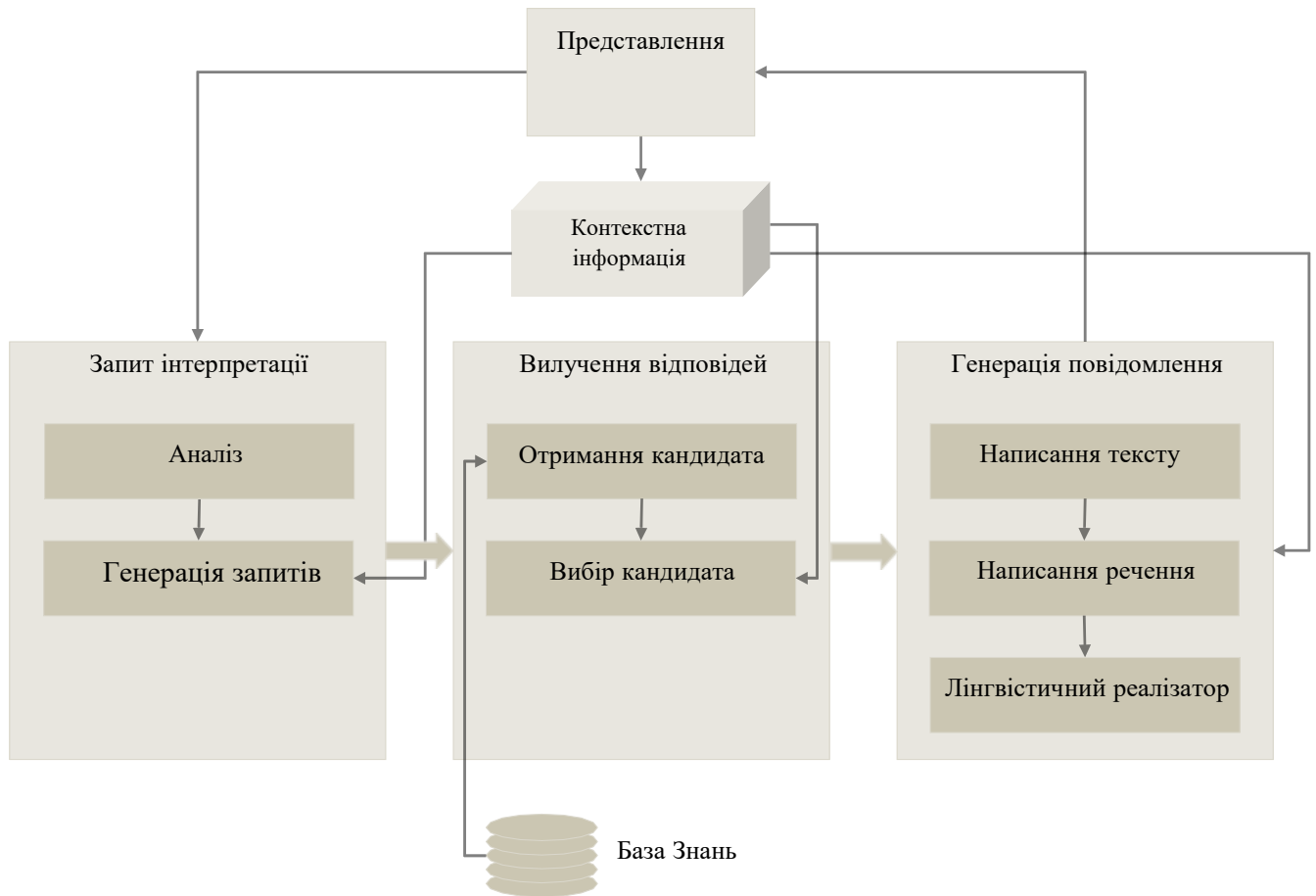


Рисунок 2.8 – Архітектура чат-бота

Архітектура чат-боту, яка зображена на рис.2.8, складається з трьох основних частин: інтерпретація запитів, вилучення відповідей і генерації повідомлень. Генерація повідомлень відповідає генерації природних мов (NLG), описаному в Reiter and Dale (2000).

2.3 Аналіз вимог

Перш ніж приступати до проектування діалогової системи, сформулюємо вимоги до розроблюваної системи на основі даних, отриманих в першому розділі, ряді внутрішніх та зовнішніх особливостей протипування та розробки чат-ботів, зокрема чат-ботів для ВНЗ, які буквально зробили революцію в світі вищої освіти, повністю змінюючи спосіб взаємодії студентів зі своїми університетами. Завдяки чат-ботам ефективність освітнього процесу значно зросла. Кожному ВНЗ необхідні якісь інструкції для початківців (абітурієнтів). Наша система повинна розроблятися для **абітурієнтів, студентів та викладачів**, котрим, використовуючи

додаток на смартфоні, потрібно дізнатись необхідну інформацію про ВНЗ, наприклад, адресу, контакти, електроний розклад занять, веб-сайт, соціальні сторінки, тощо.

Розроблювана діалогова система повинна обробляти запити користувачів та відповідати наступним вимогам:

1. Бути легко доступною і простою у використанні для більшості студентів.
2. Обробляти повідомлення користувачів в режимі реального часу.
3. Мати можливість змінювати дані, які використовуються чат-ботом.
4. Безперервно працювати, тому що користувачам системи може знадобитися інформація в будь-який час.

Розроблюваний чат-бот повинен працювати в наступній послідовності:

1. Користувач надсилає команду в чат.
2. Месенджер обробляє запит та передає його на сервер чат-боту.
3. Сервер надає сформовану відповідь в чат через front-end діалогової системи.
4. Отримана відповідь з'являється на екрані у користувача.

2.4 Вибір сервісу для обміну повідомленнями

На даний момент існує декілька доступних платформ для обміну миттєвими повідомленнями: WhatsApp, ВК, Facebook, Telegram та інші.

Telegram - багатоплатформовий месенджер, який дозволяє обмінюватися повідомленнями і медіафайлами багатьох форматів. На даний час Telegram входить у п'ятірку найпопулярніших месенджерів у світі, серед переваг якого є: використовується відкритий API, ним можливо користуватись на більшості ОС. Крім цього, на даний момент основне спілкування більшості студентів з навчальних питань відбувається саме в месенджері Telegram. Telegram для iPhone був випущено 14 серпня 2013 року. Альфа-версія Telegram для Android була офіційно випущена 20 жовтня 2013 року. Логотип зображено на рис.2.9.



Рисунок 2.9 – Логотип Telegram

Telegram створений для всіх, хто вважає за краще швидкий і надійний обмін повідомленнями. Підприємці та невеликі компанії оцінять групи, додатки для комп'ютера і роботу з файлами. У групових чатах доступна пересилання повідомлень, згадки і хештеги - тому навіть у великих конференціях (підтримується до 100 тисяч учасників) спілкування ефективно і впорядковано. На відміну від WhatsApp, Telegram працює в хмарі, і, крім того, надійно зашифровує весь трафік (в тому числі і отримані елементи). Завдяки цьому, можна користуватися месенджером на будь-якому пристрої і обмінюватися необмеженою кількістю фотографій, відео та інших файлів (doc, zip, mp3, і т. Д.). Також, завдяки розподіленій інфраструктурі та потужним серверів, Telegram набагато безпечніше і швидше, ніж будь-яка подібна програма (WhatsApp, Viber, та інші). Іншими словами, ви можете користуватися Telegram не тільки на вашому мобільному пристрої, але і на домашньому комп'ютері, і навіть онлайн, чого не може зробити Whatsapp.

2.4.1 Основні можливості Telegram

Основні можливості месенджера Telegram:

- Є можливість писати всім, хто є в ваших контактах і у кого встановлений Telegram. Також ви можете зв'язатися з будь-якою людиною, у якого є ім'я користувача (username). Для свого облікового запису ви теж можете придумати ім'я користувача. Зверніть увагу, що в такому випадку інші користувачі не бачать ваш номер телефону.
- Щоб запросити друзів в Telegram, досить відправити посилання на скачування програми будь-якого вашого контакту. Як тільки ваш друг встановить додаток, він з'явиться у вашому списку контактів.
- Статус онлайн або офлайн. Зовсім приховати статус онлайн можна. Однак, ви можете приховати інформацію про те, коли в останній раз ви були в мережі. Це робиться в налаштуваннях Приватності і безпеки. Зверніть увагу: ви не побачите останнім часом в мережі тих людей, від яких приховали своє.

— Безпека. Telegram набагато безпечніше інших подібних месенджерів, типу WhatsApp або Line. Телеграм заснований на протоколі MTProto створеному за перевіреними часом алгоритмами, і забезпечує високу швидкість і надійність.

2.5 Діаграма варіантів використання освітнього чат-боту

Діаграми варіантів використання (рис. 2.10) відображають взаємини і залежності між прецедентами і діючими особами, які беруть участь в процесі.

Користувач - дійова особа, що взаємодіє з системою через варіант використання. Позначається умовним знаком людини.

Прецедент (варіант використання) - еліпс з написом, що показує їх системою дії, що призводять до спостережуваних користувачів результатами. Діаграма варіантів використання дозволяє наочно показати функціональні вимоги до проектованої системи.

Виходячі з цієї спроектованої діаграми яка зображена на рис.2.10 ми можемо побачити що користувачу достатньо буде зайти в месенджер знайти чат-бота через пошук, або наприклад перейти за посиланням через сайт ВНЗ, та почати розмову з віртуальним помічником.

Варіанти запитів можуть бути такі :

- Дізнатись адресу ВНЗ -> Чат-бот надасть адресу ВНЗ із посилання на мапу в Google Maps.
- Отримати інформацію про розклад занять,написавши вашу назву групи, та день -> Чат-бот надасть перелік занять на надану дату.
- Отримати посилання на сайт ВНЗ -> Чат-бот надасть посилання, та запропонує підписатися на соціальні мережі.
- Отримати контакти викладачів ВНЗ -> Чат-бот надасть контактну інформацію про викладачів вашої кафедри де ви навчаєтесь, а також посилання на сайт.

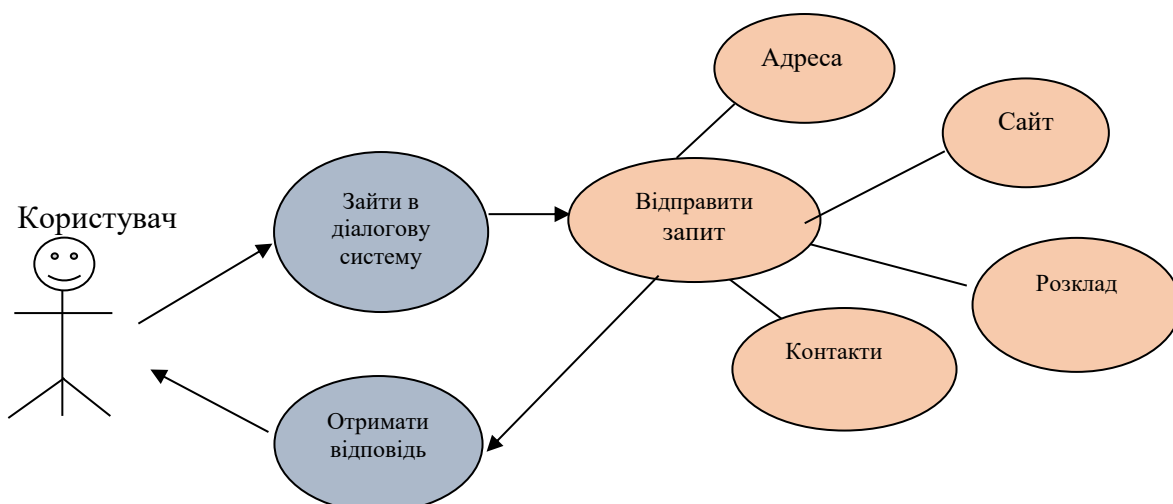


Рисунок 2.10 – Діаграма варіантів використання

2.6 Висновки до 2 розділу

У цьому розділі був представлений процес проектування системи, при якій були представлені вимоги, наочно показана очікувана робота системи з боку користувача за допомогою діаграми варіантів використання. Було проаналізовано зміни в рейтингах віртуальних чат-ботів, месенджерів, які найчастіше використовує сучасна молодь. У відповідності з поставленими вимогами були обрані найбільш підходящі для реалізації системи сервіси і платформи. Була представлена спроектована схема варіантів використання додатку, яка буде далі реалізована на практиці.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ СПРОЕКТОВАНОЇ СИСТЕМИ

3.1 Засоби реалізації

3.1.1 Мова програмування JavaScript



Рисунок 3.1 – Логотип JavaScript

JavaScript (рис. 3.1) - мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний і функціональний стилі програмування. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах, як мова сценаріїв для додання інтерактивності веб-

сторінок. У нашому випадку мова програмування JavaScript із використанням фреймворку NodeJS підходить для написання чат-боту через платформу Dialogflow.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу. На JavaScript вплинули багато мов, при розробці була мета зробити мову схожим на Java, але при цьому легким для використання непрограмістів.

3.1.2 Переваги JavaScript

- Жоден сучасний браузер не обходиться без підтримки JavaScript.
- З використанням написаних на JavaScript плагінів і скриптів впорається навіть не фахівець.
- Корисні функціональні налаштування.
- Постійно удосконалюється мова - зараз розробляється бета-варіація проекту, JavaScript2.
- Взаємодія з додатком може здійснюється навіть через текстові редактори - Microsoft Office і Open Office.
- Перспектива використання мови в процесі навчання програмуванню і інформації.

3.1.3 Недоліки JavaScript

- Знижений рівень безпеки через повсюдний і вільний доступ до вихідного коду популярних скриптів.
- Безліч дрібних дратівливих помилок на кожному етапі роботи. Велика частина з них легко виправляється, але їх наявність дозволяє вважати цю мову менш професійною, порівняно з іншими.
- Повсюдне поширення. Своєрідним недоліком можна вважати той факт, що частина активно використовуваних програм (особливо додатків) перестануть існувати при відсутності мови, оскільки цілком базуються на ній.

3.2 Середовище розробки Visual Studio Code

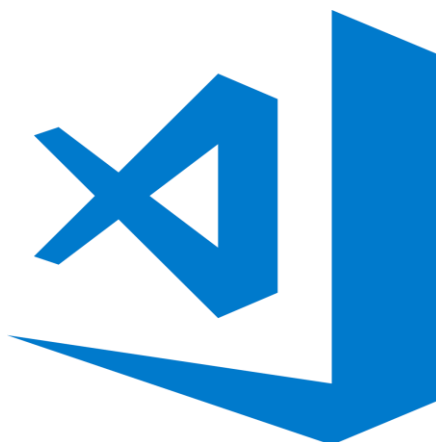


Рисунок 3.2 – Логотип VS code

Visual Studio Code (рис.3.2) - популярний безкоштовний редактор коду, створений компанією Microsoft для програмістів. VS Code ніяк не пов'язаний з Visual Studio. VS Code працює швидше Атома, активно розвивається і легко розширюється плагінами. Visual Studio Code має підтримку плагінів, доступних через Visual Studio Marketplace. Вони можуть включати в себе доповнення до редактора, підтримку додаткових мов програмування, статичні аналізатори коду.

Детальніше про **переваги VSC** наведено нижче.

3.2.1 Легкість VSC

У своєму серці Visual Studio Code має блискавичний редактор вихідних кодів, ідеально підходить для щоденного використання. Завдяки підтримці сотень мов, VS Code допомагає вам підсвічуванням синтаксису, відповідністю дужок, автоматичними відступами, вибором вікон та фрагментів і багатьом іншим. Інтуїтивно зрозумілі комбінації клавіш, зручні налаштування та доповнення клавіатурних скорочень дозволяють легко переглядати код. Для серйозного кодування ви часто користуєтеся інструментами з більшим розумінням коду, ніж просто блоками тексту. Код Visual Studio містить вбудовану підтримку завершення коду Intelli Sense, розуміння багатозначного семантичного коду та навігацію, а також рефакторинг коду. І коли кодування стає жорстким, важко отримати налагодження. Налагодження часто є однією з функцій, яку розробники найчастіше забувають при зростаючому досвіді кодування. Редактор коду Visual Studio містить інтерактивний відладчик, тому ви можете переходити через вихідний код, перевіряти змінні, переглядати стеки викликів і виконувати команди з консолі.

Код VS також інтегрується з інструментами побудови та сценаріїв для виконання типових завдань, що робить щоденні робочі процеси більш швидкими. Код VS підтримує Git, тому ви можете працювати з керуванням джерелом, не залишаючи редактор, включаючи перегляд змін у відставці.

3.2.2 Функції

Налаштуйте кожну функцію за своїм смаком і встановіть будь-яку кількість розширень сторонніх виробників. Хоча більшість сценаріїв працюють "з коробки" без конфігурації. VS Code - це проект з відкритим кодом, завдяки чому ви також можете внести свій внесок до зростаючої та живої спільноти GitHub. Код VS включає в себе збагачену вбудовану підтримку для розробки Node.js з JavaScript і TypeScript, що працюють на тих самих базових технологіях, що приводять Visual Studio. VS Code також включає в себе чудові інструменти для веб-технологій, таких як JSX / React, HTML, CSS, SCSS, Less, і JSON.

3.2.3 Міцна та розширювана архітектура

Архітектурно, Visual Code Code поєднує в собі найкраще з веб, рідних і розмовних специфічних технологій. Використовуючи Electron, VS Code поєднує веб-технології, такі як JavaScript і Node.js зі швидкістю і гнучкістю вітчизняних програм. VS Code використовує більш нову, більш швидку версію того ж самого індустріально-сильного HTML-редактора, який керує редактором хмари "Monaco", F12 Tools Internet Explorer та іншими проектами. Крім того, VS Code використовує архітектуру сервісів інструментів, що дозволяє їм інтегруватися з багатьма тими ж технологіями, що забезпечують роботу Visual Studio, включаючи Roslyn для .NET, TypeScript, механізм налагодження Visual Studio тощо. Код Visual Studio включає в себе загальнодоступну модель розширюваності, яка дозволяє розробникам створювати та використовувати розширення, а також налаштовувати свій досвід редагування-збирання-налагодження.

3.3 Платформа Dialogflow

Dialogflow є зручною, інтуїтивно зрозумілою платформою, яка використовує нейролінгвістичне програмування і підходить для створення чат-ботів за допомогою інтеграції з Telegram. Платформа Dialog Flow розроблена для початківців які тільки починають свій шлях по протипуванню діалогових систем для віртуальних клієнтів, вона проста і надає широкі варіанти редагування діалогових вікон. Після створення агента у Dialogflow можна

використовувати інструменти інтеграції Dialogflow, щоб зробити ваш агент доступним на декількох платформах. Інтеграція одним натисканням Dialogflow допоможе вам керувати інтеграцією вашого агента з помічником Google за допомогою дій на Google і ряду популярних платформ для обміну повідомленнями, таких як Slack, Facebook Messenger і Twitter. Dialogflow також дозволяє агентам легко експортуватися або імпортуватися з інших платформ для розуміння природної мови, таких як Amazon Alexa і Microsoft Cortana.

3.4 Архітектура роботи додатку

Як і всі додатки наш віртуальний складається з двох основних частин розробки: це back-end та front-end.

Back-end є частиною бота, яка виконується на сервері, у нашому випадку це Dialog Flow на мові програмування JavaScript та фреймворку NodeJS. Node JS – програмна платформа, заснована на движку V8 (здійснює трансляцію JavaScript в машинний код), що перетворює JavaScript з вузькоспеціалізованого мови в мову загального призначення. Node.js додає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API (написаний на C++), підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики з JavaScript-коду.

Front-end - частиною бота (месенджер **Telegram**), взаємодія користувача з якою проводиться через додаток на мобільному телефоні або ПК. Даний месенджер служить чат-платформою, з якою пов'язується back-end і встановлює URL-з'єднання, яке дозволяє безпечно відправляти і отримувати повідомлення через простий HTTP-запит(S) за допомогою фреймворка NodeJS.

3.5 Схема роботи додатку та розробка бота

Освітньо-інформаційний чат-бот розробляється для студентів та викладачів, яким необхідно дізнатись ту чи іншу інформацію про ВНЗ, використовуючи додаток на телефоні (наприклад адресу, контакти, електроний розклад занять, веб-сайт, соціальні сторінки, тощо).

3.5.1 Схема роботи додатку

На рис. 3.3 зображена схема роботи чат-боту. Проаналізуємо схему детальніше:

1. Користувач за допомогою свого смартфона переходить у месенджер, знаходить чат – бот. Чат – бот у свою чергу надає користувачеві всі

доступні команди. Це може бути отримання адреси ВНЗ, отримання контактів викладачів, отримання посилання на сайт ВНЗ, отримання розкладу занять на сьогодні або на будь-який інший день.

2. Після того як користувач обрав команду, сервер Telegram отримує повідомлення, що користувачеві необхідна саме така інформація, він передає на свій сервер HTTPS-запит, обробляє, перевіряє, чи є вона в базі даних, та передає на сервер платформи Dialog Flow, на якій створена діалогова система.
3. Платформа обробки розмовних мов Dialog Flow приймає запит від сервера, обробляє його за допомогою NLP (розуміння розмовних мов) і далі, якщо така команда прописана на сервері, система передає його у зворотньому напрямку вже на сервер Telegram.
4. Фрейм Користувач отримує необхідну відповідь.



Рисунок 3.3 – Схема роботи чат-боту

Чат-боти, зокрема для освіти, буквально зробили революцію у світі вищої освіти, повністю змінюючи спосіб взаємодії студентів зі своїми університетами. Завдяки чат-ботам ефективність освітнього процесу значно зростає. Кожному ВНЗ необхідні якісь інструкції для

абітурієнтів. На початку семестру у кожного абітурієнта та студента виникає питання, зазвичай абітурієнти звертаються з такими питаннями до викладачів або, набравшись сміливості, до старших студентів. Тому було б доречним створити універсального освітнього чат-бота для ВНЗ СНУ імені Володимира Даля із використанням діалогових систем, який би зміг допомогати абітурієнтам та студентам отримувати необхідну інформацію.

3.6 Розробка чат-боту із використанням діалогових систем

Перш ніж починати розробку бота, необхідно зареєструватися, через спеціального бота *@BotFather* прописати команду `/start` і отримати унікальний `id`, який є одночасно і токеном (рис.3.4 та рис. 3.5).



Рисунок 3.4 - Токен для авторизації

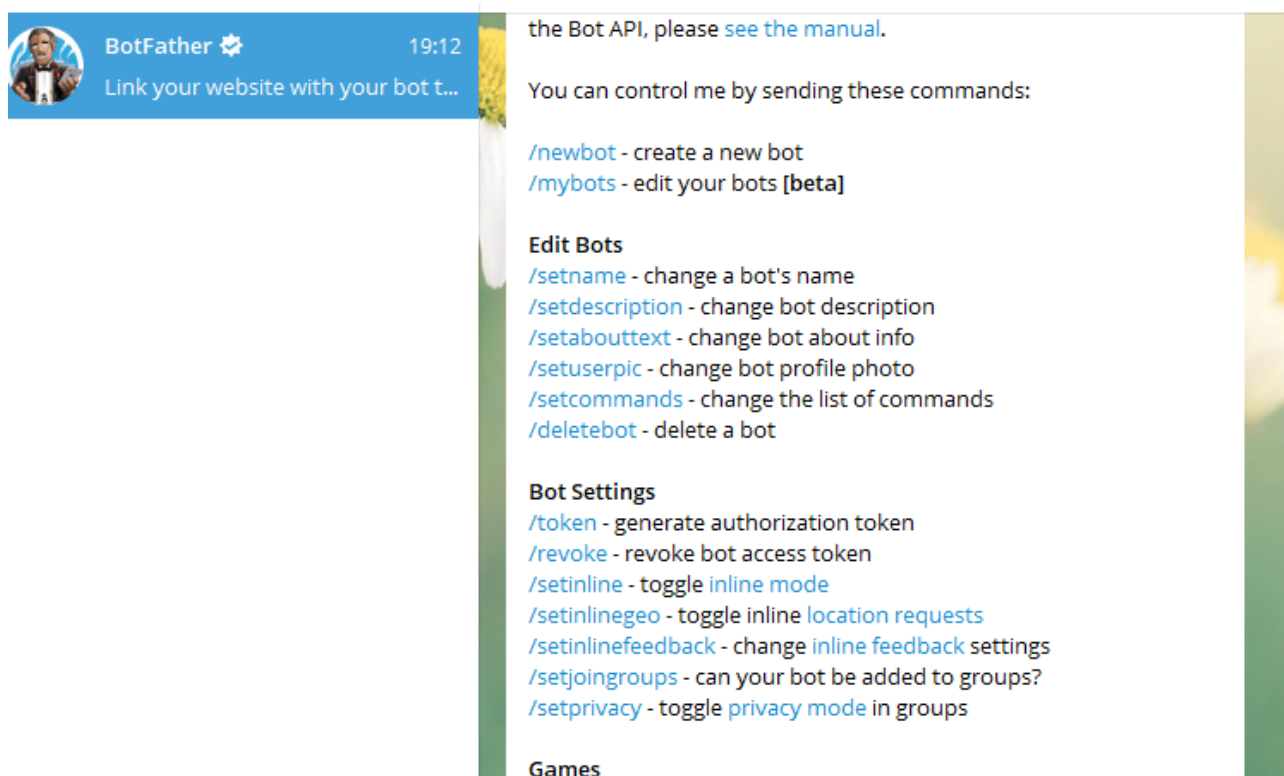


Рисунок 3.5 - Реєстрація нового бота через спеціального бота

3.6.1 Створення чат-бота в месенджері

Задаємо боту команду `/start` і отримуємо список всіх його команд. Першу і головну - `/newbot` - відправляємо йому і бот просить придумати ім'я нашого нового бота. Єдине обмеження на ім'я - в кінці воно повинно закінчуватися на «bot». Створуємо **чат-бот** через месенджер Telegram, даємо йому ім'я **VNUbot**, можемо додати аватар та опис, для яких цілей буде використовуватись бот (рис. 3.6).

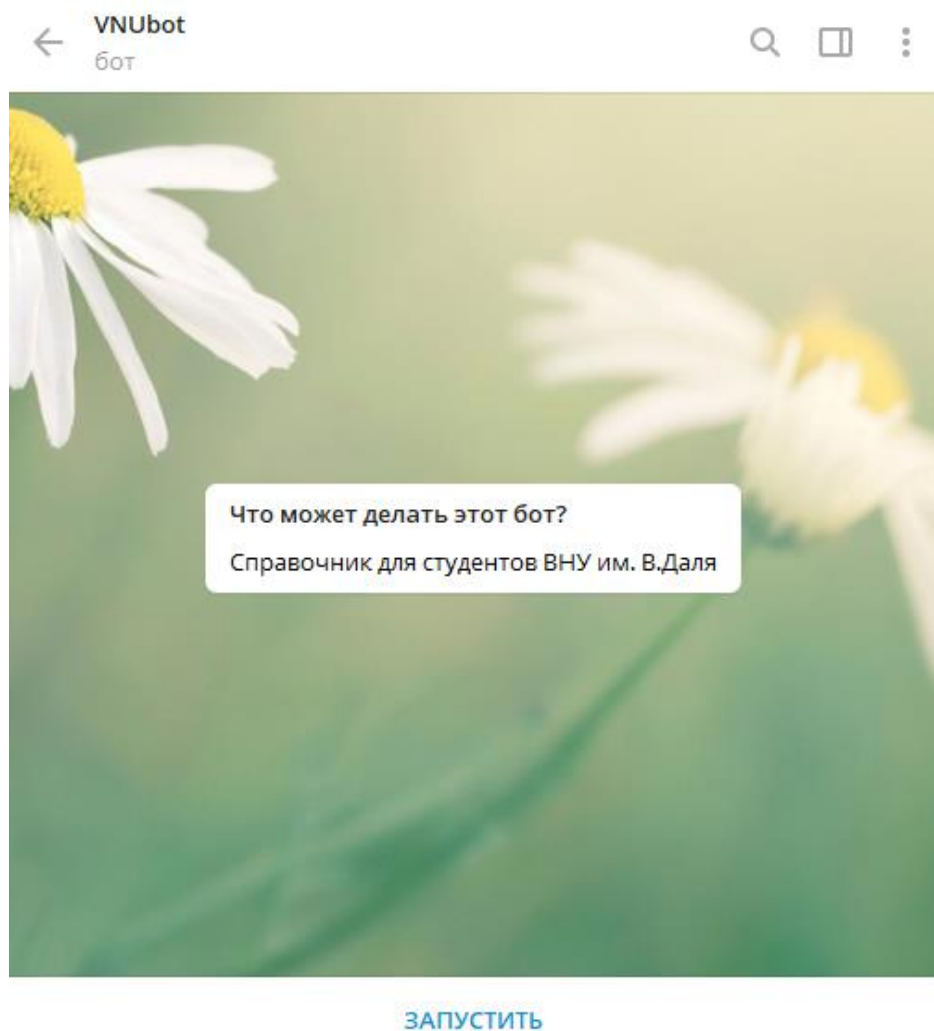


Рисунок 3.6 - Створений чат-бот

3.6.2 Розробка діалогових систем

Після того, як ми створили нашого бота, переходимо у платформу Dialog Flow для створення прототипу діалогових систем (рис. 3.7). Реєструємося через Google та потрапляємо у меню створення діалогових систем.

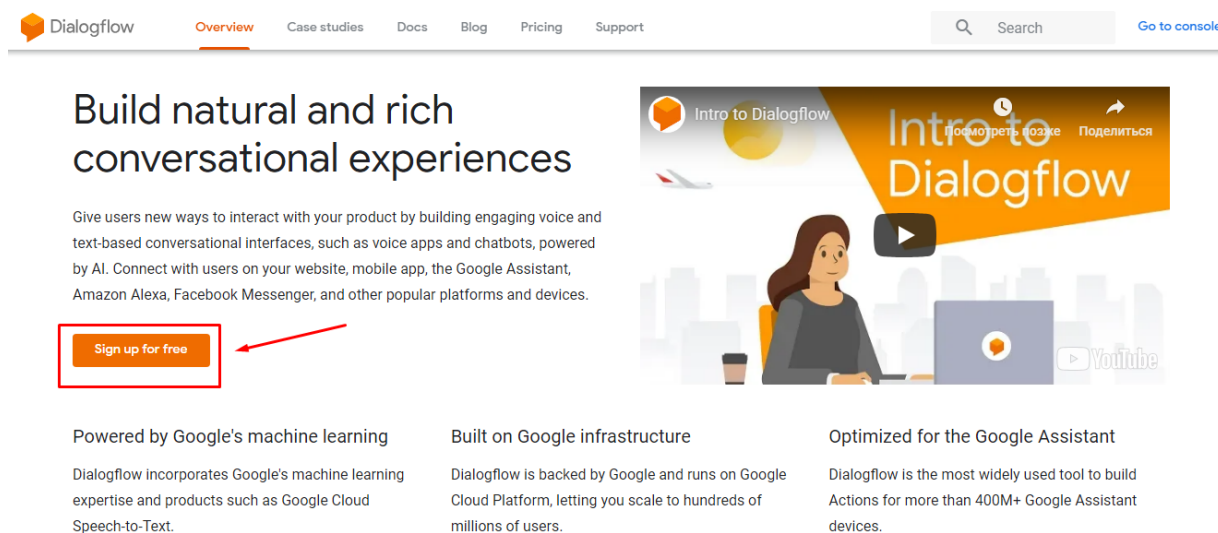


Рисунок 3.7 – Головна сторінка Dialogflow

У сервісі Dialog Flow за допомогою методів машинного навчання текст приводиться до одного з заздалегідь сконфігурованих шаблонів Intents, та зокрема Entities. На рис. 3.8 зображено ліве меню.

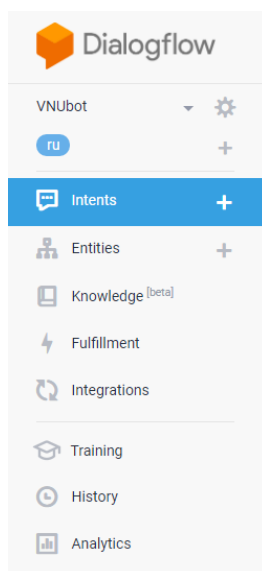


Рисунок 3.8 – Ліве меню Dialogflow

3.6.3 Шаблони Intents

Intents (намір, призначення) - є зіставлення між тим, що говорить користувач, і якої мети повинно бути досягнуто програмним забезпеченням. За допомогою цього меню створюються діалогові вікна, які представлені нижче на рис. 3.9. У ньому за допомогою отриманих в сервісі даних формується команда, яка буде відправлена в діалоговому вікні користувачеві. Були створені шаблони із запитом користувача. Розберемо окремо кожний шаблон діалогових вікон.

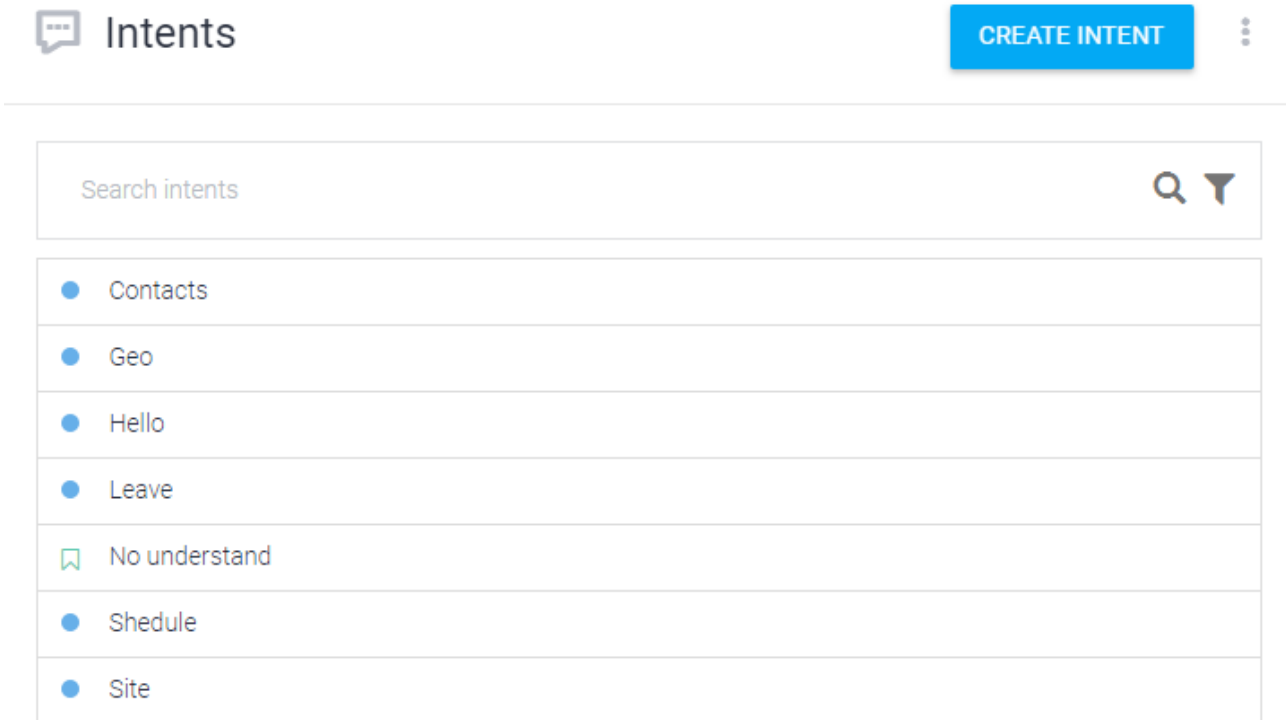


Рисунок 3.9 – Сконфігуровані шаблони Intents

У кожному шаблоні є окремі фрази, які може задати агент, та фрази, на які повинен відповісти бот користувачеві. Вони показані на рис. 3.10.

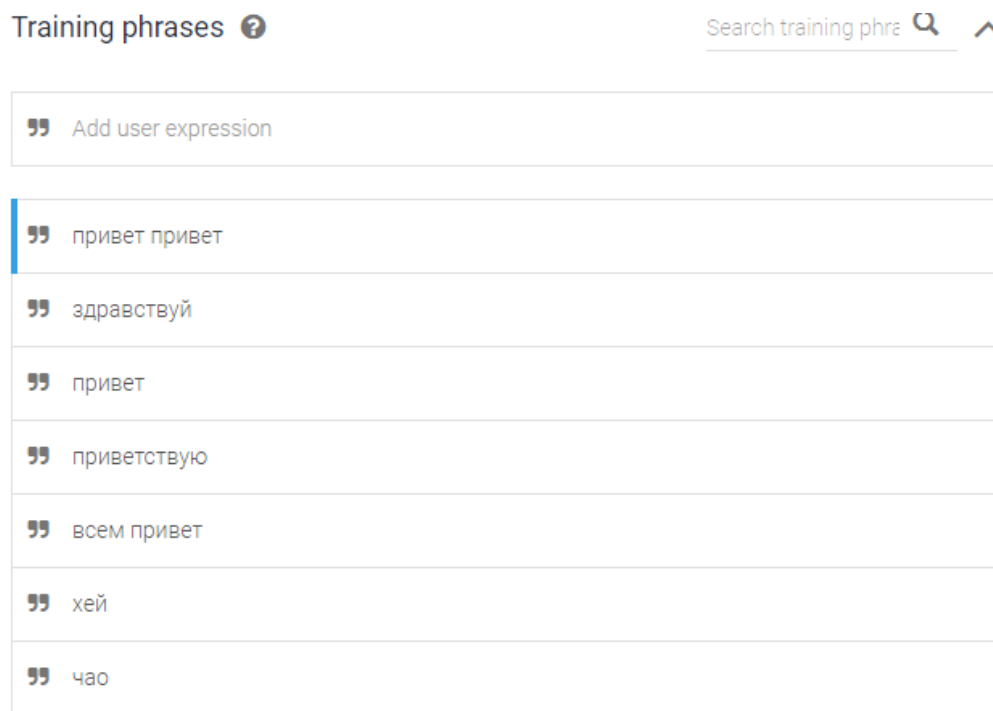


Рисунок 3.10 – Фрази, які буде використовувати агент

Фрази, які буде використовувати чат-бот, конфігуруються кожна окремо. До них ще можна додати 3 або 4, щоб чат-бот був схожим на співрозмовника із використанням NLP.

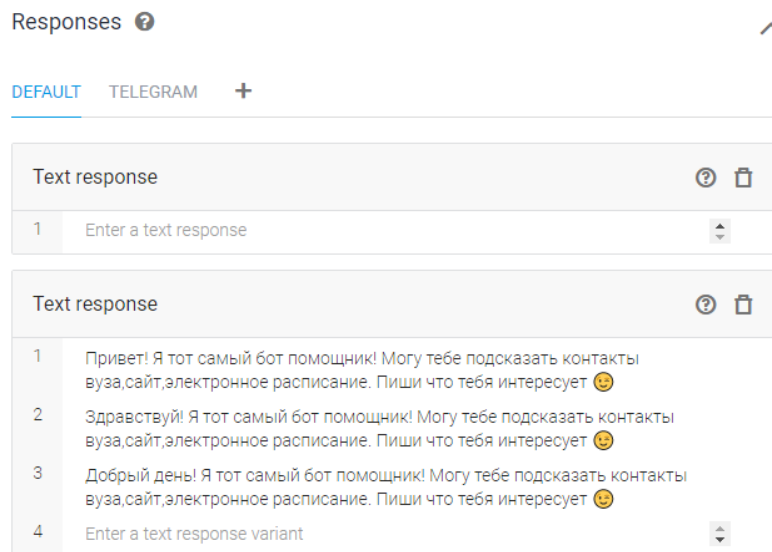


Рисунок 3.11 – Фрази чат-боту

Розберемо приклад шаблону Гео (рис. 3.12), який використовує засіб Google API. Нам він потрібний для того, щоб отримувати інформацію з Google Maps.

Geo SAVE ⋮

Action and parameters ^

Enter action name //

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	Address	@Address	SAddress	<input checked="" type="checkbox"/>	https://go.o.gl/...
<input checked="" type="checkbox"/>	geo-state	@sys.geo-state	Sgeo-state	<input checked="" type="checkbox"/>	https://go.o.gl/...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

[+ New parameter](#)

Рисунок 3.12 – Шаблон Geo із використанням API Google Maps

Коли користувач пише в чат команду, щоб дізнатись адресу, і відправляє її на сервер Dialog Flow, де в шаблоні вказаний зовнішній API, система повертає відповідь користувачеві зі вказаною адресою ВНЗ та посиланням на Google Maps (див. рис. 3.13).

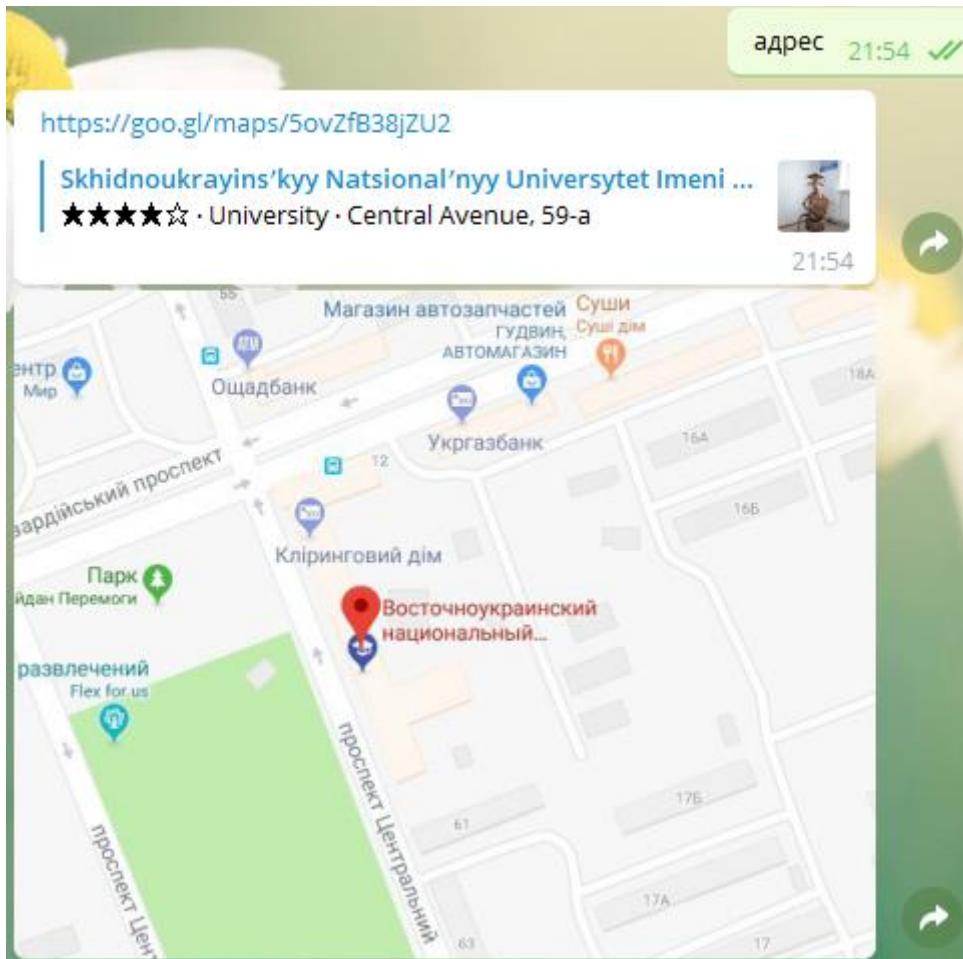
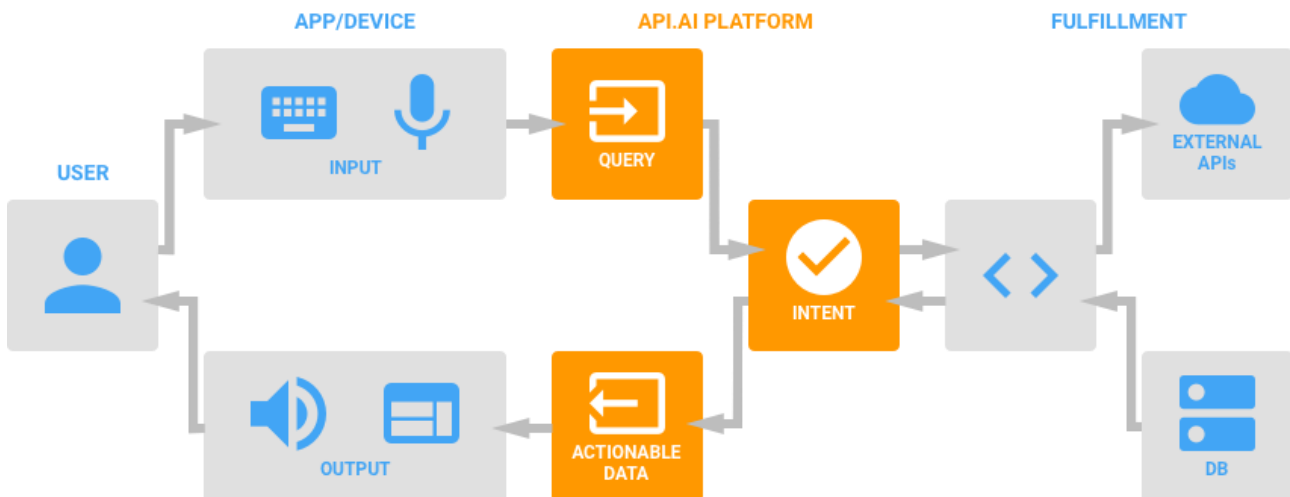


Рисунок 3.13 – Адреса ВНЗ у месенджері

Тим самим, користувач побачить в діалозі перетворення запиту в природній формі в команду. Дана команда буде оброблена відповідним модулем.

Загальна схема роботи сервісу Dialog Flow представлена на рис. 3.14.



Малюнок 3.14 – Схема роботи DialogFlow

3.6.4 Об'єкти Entities

Ще один пункт меню - це об'єкт Entity, що представляє собою інструмент, який використовується для вилучення значень параметрів з вхідних запитів на природній мові. Об'єкти, які використовуються конкретним агентом, будуть залежати від очікуваних значень параметрів, які будуть повернуті агентом. На рис. 3.15 представлені всі створені об'єкти.

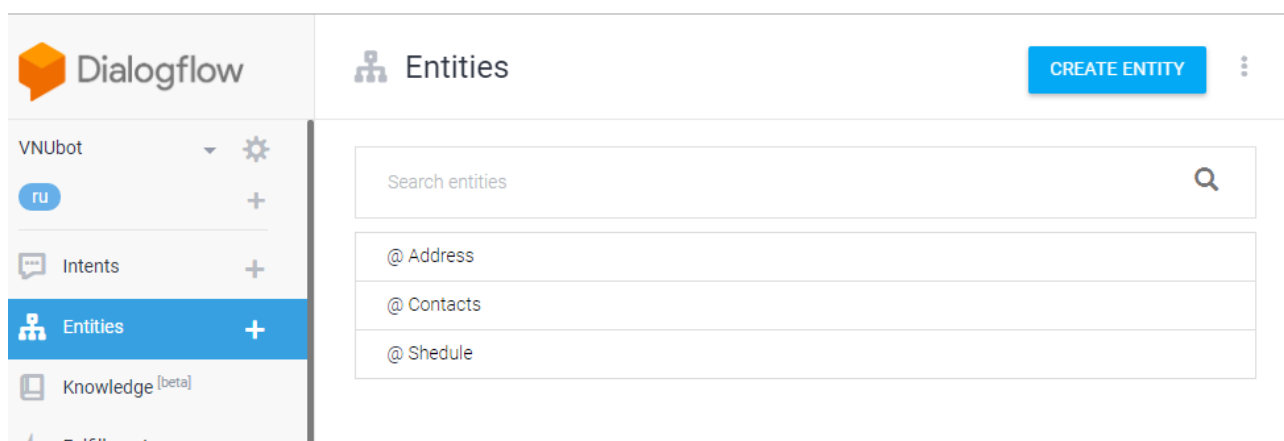


Рисунок 3.15 – Об'єкти entities

Після того, як ми виконали налаштування, додали шаблони та об'єкти, можемо виконати інтеграцію з сервісами, у яких можливе використання чат-ботів.

Можливо виконати інтеграцію з Viber, Slack, Facebook та іншими (рис.3.16); у нашому випадку ми обираємо Telegram, тому що цим месенджером найчастіше користуються студенти та абітурієнти ВНЗ СНУ ім. Володимира Даля.

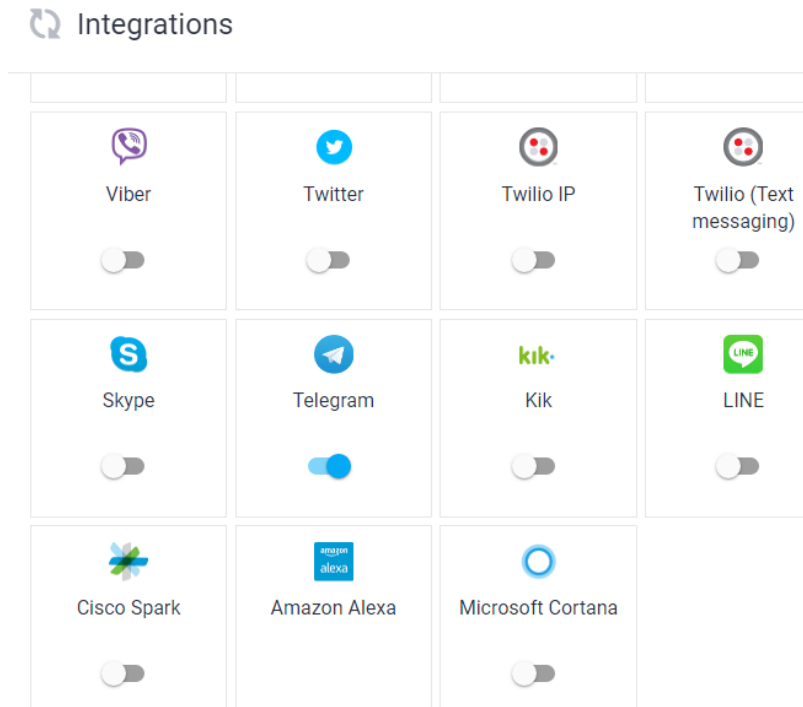


Рисунок 3.16 – Виконання інтеграції з сервісом

Після того, як виконали інтеграцію з сервісом, переходимо у наш месенджер та пишемо питання чат-боту (рис 3.17 та рис. 3.18).

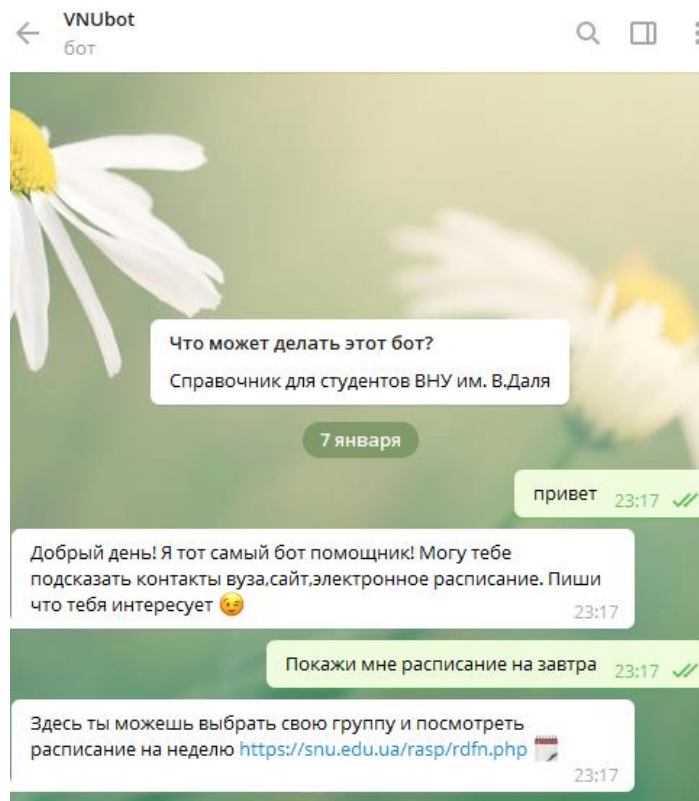


Рисунок 3.17 – Запитання студента для чат-бота про розклад занять на завтра

Якщо питань у користувача не залишилось, він може завершити роботу з ботом, який у відповідь надасть посилання на соціальную сторінку у фейсбуці (рис. 3.18). Також студент може залишити побажання по роботі чат-бота адміністратору, надати ідеї по подальшій розробці додатка.

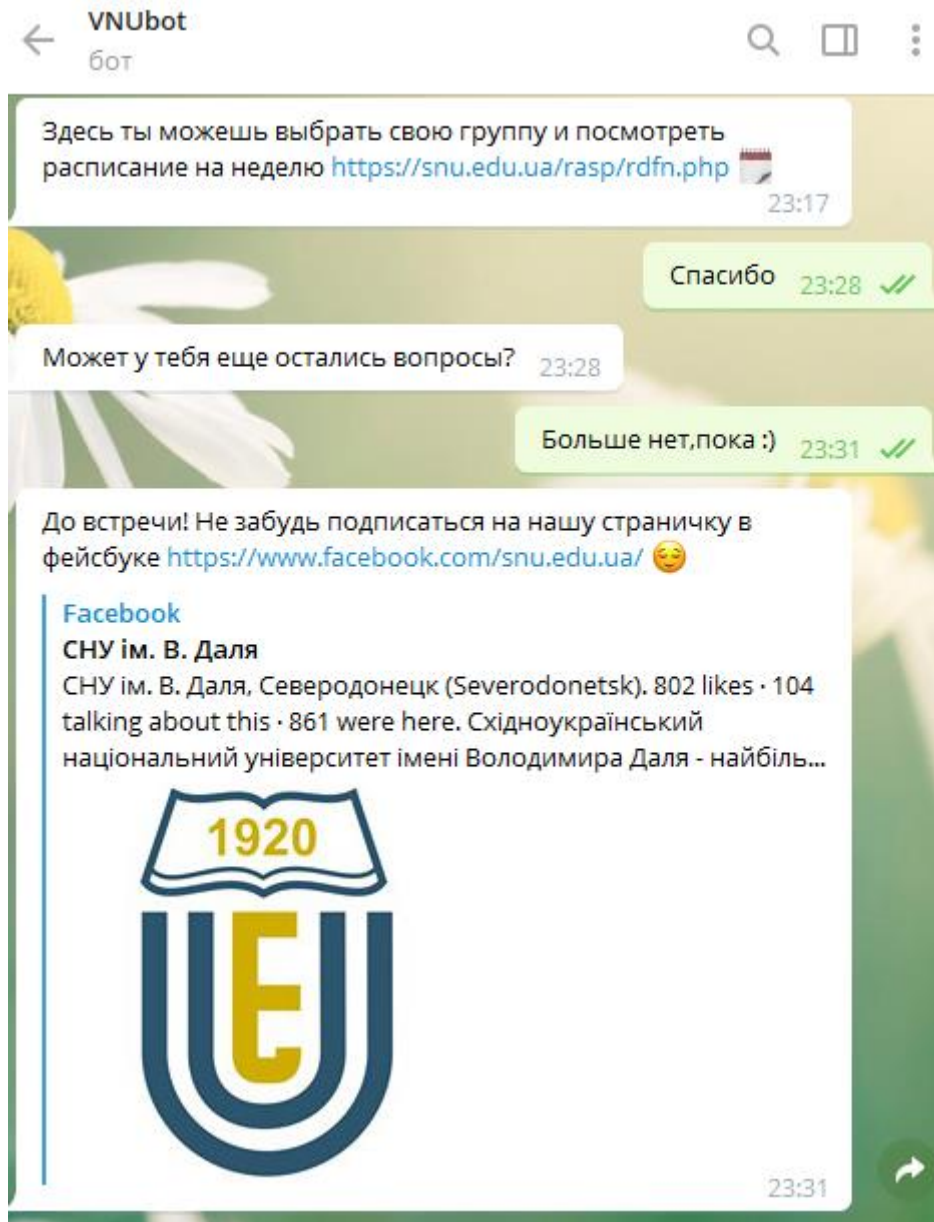


Рисунок 3.18 – Завершення розмови.

3.7 Висновки до 3 розділу

У цьому розділі була описана реалізація віртуального чат-боту із використанням платформи для створення та прототипування діалогових вікон Dialog Flow. Були представлені схеми роботи як усієї системи, так і її окремих модулів. Реалізований додаток може спростити роботу деканатів ВНЗ СНУ ім. В.Даля. Додаток може надавати абітурієнтам та студентам детальну інформацію про адреси, контакти викладачів, може виводити розклад занять та інше. Докладно описані всі використовувані при реалізації програмні засоби. Всі поставлені для розроблюваної системи вимоги виконані. На рисунках, що ілюструють роботу системи, показано, що підсистема працює коректно.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної магістерської роботи було використання інформаційних технологій для прототипування діалогових систем. За цією моделлю в подальшому розроблятиметься реальна діалогова система, яка буде використовуватися студентами ВНЗ. Так як в процесі проектування використовувалося ПК, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде використовуватися розроблена об'єктна модель для оцінки ризиків та аналізу кібербезпеки під час атак, розрахунку вразливостей системи, визначення зв'язків атак і захистів та прогнозування властивостей системи.

4.1 Загальні питання з охорони праці

В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці.

4.1.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім — під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Обов'язки працівників щодо додержання вимог нормативно-правових актів з охорони праці (ст. 14), відповідальність робітників всіх категорій за порушення вимог щодо охорони праці (ст. 44) та структура організації/виробництва системи управління охорони праці визначені у [1].

4.1.2 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [2].

Також впроваджені організаційні заходи з пожежної безпеки - навчання і перевірку знань відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях

України, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 29.09.2003 N 368, зареєстрованого в Міністерстві юстиції України 11.12.2003 за N 1148/8469 [3].

4.2 Аналіз стану умов праці

Робота над створенням об'єктної моделі забезпечення оцінки кібербезпеки, розрахунок уразливості системи і визначення зв'язків атак і захистів проходитиме в побутовому приміщенні. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 - Розміри приміщення

Найменування	Значення
Довжина, м	3
Ширина, м	3
Висота, м	2,5
Площа, м ²	9
Об'єм, м ³	22,5

Згідно з [4] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є дивани і журнальний стіл, обставлені живими квітами. Також робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [5] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ 800

4.2.3 Навантаження та напруженість процесу праці

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи наведені в [5].

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви:

- для операторів персональних комп'ютерів тривалістю 15 хв через дві години роботи;

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Роботу, пов'язану з ЕОМ з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання [6], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої.

Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220V \pm 5\%$;
- робочий струм $I=2A$;
- споживана потужність $P=350 \text{ Вт}$.

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [5].

За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами, шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3).

Таблиця 4.3 - Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
фізичні			
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	ДБН В.2.5-28:2015 [9]

Продовження таблиці 4.3

- підвищений рівень шуму на робочому місці	-//-	2	ДСН 3.3.6.042-99 [4]
- підвищена або знижена вологість повітря	-//-	2	ДСН 3.3.6.042-99 [4]
- підвищена або знижена рухливість повітря	-//-	1	ДСН 3.3.6.042-99 [4]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	ГОСТ 12.1.030-81 [7] ГОСТ 13109-97 [8]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	ДБН В.2.5-28:2015 [9]
- підвищена яскравість світла	порушення умов праці (організації місця праці - налагодження моніторів)	1	ДСанПіН 3.3.2.007-98[5]
- понижена контрастність	-//-	1	ДСанПіН 3.3.2.007-98[5]
психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	НПАОП 0.00-1.28-10[6] ДСанПіН 3.3.2.007-98[5]
- фізичні (статичне –	порушення умов праці	2	НПАОП 0.00-1.28-

сидіння)	(організації місця праці - сидіння користувача,) та організації робочого часу - безперервна робота)		10[6] ДСанПіН 3.3.2.007-98[5]
----------	--	--	----------------------------------

4.3.2 Пожежна безпека

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) системою запобігання пожежі,
- 2) системою протипожежного захисту,
- 3) організаційно-технічними заходами.

Згідно [10] таке приміщення, площею 9 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння. Відповідно до норм первинних засобів пожежогасіння пропонується використовувати:

- ручний вуглекислий вогнегасник ОУ-5 в кількості 1 шт.;
- ковпачок 1 м², ковшу 2×1,5 м² або азбестове полотно 2×2 м² в кількості 1 шт.

Виникнення пожежі можливе, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Вірогідність пожежної небезпеки приймається значною, якщо ймовірна взаємодія цих трьох чинників. Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, підлоги, двері, ізоляція силових, сигнальних кабелів і т.д.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія

електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [11] і наведені в табл. 4.4:

Таблиця 4.4 - Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

4.4.2 Освітлення

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 3 \cdot 3 = 9 \text{ м}^2,$$

$$S_{\text{вік}} = 1/10 \cdot 9 = 0,9 \text{ м}^2.$$

Приймаємо 2 вікна площею $S = 0,9 \text{ м}^2$ кожне.

Ліхтарі загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 9 \text{ м}^2$;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ;

$Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \times 9 \times 1.1 \times 1.5}{5400 \times 0.575 \times 2} = 0,7 \approx 1$$

Приймаємо освітлювальну установку, яка складається з 1 світильника, який складається з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.5 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення $> 40 \text{ м}^3$ на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.6 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Загальний опір захисного заземлення визначається за формулою:

$$R_{\text{зг}} = \frac{R_{\zeta} \cdot R_n}{R_n \cdot n \cdot \eta_{\zeta} + R_{\zeta} \cdot \eta_n} \quad (4.3)$$

де R_{ζ} - опір заземлення, якими можуть бути труби, опори, кути і т.п., Ом;

R_n - опір опори, яка з'єднує заземлювачі, Ом;

n - кількість заземлювачів;

η_{ζ} - коефіцієнт екранування заземлювача; приймається в межах $0,2 \div 0,9$; $\eta_n = 0,7$

η_n - коефіцієнт екранування сполучної стійки; приймається в межах $0,1 \div 0,7$; $\eta_n = 0,5$;

Опір заземлення визначається за формулою:

$$R_{\xi} = \frac{\rho}{2\pi \cdot l} \left(\ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right) \quad (4.4)$$

де ρ - питомий опір ґрунту, залежить від типу ґрунту, Ом·м;

для піску - $400 \div 700$ Ом·м; приймаємо $\rho = 400$ Ом·м;

l - довжина заземлювача, м; для труб - 2-3 м; $l = 3$ м;

d - діаметр заземлювача, м; для труб - 0,03-0,05 м; $d = 0,05$ м;

t - відстань від середини забитого в ґрунт заземлювача до рівня землі, м; $t = 2$ м.

$$R_{\xi} = \frac{400}{2 \cdot 3,14 \cdot 3} \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \ln \frac{4 \cdot 2 + 3}{4 \cdot 2 - 3} \right) = 110, \text{ Ом}$$

Опір смуги, що з'єднує заземлювачі, визначається за формулою:

$$R_n = \frac{\rho}{2\pi \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot t_1} \quad (4.5)$$

де L - довжина смуги, що з'єднує заземлювачі (м) і приблизно дорівнює периметру

будівлі: $P_{\text{буд.}} = 42 \cdot 2 + 38 \cdot 2 = 160$ м; $L = 160$ м;

b - ширина смуги, м; $b = 0,03$ м;

t_1 - глибина заземлення від рівня землі, м; $t_1 = 0,5$ м.

$$R_n = \frac{400}{2 \cdot 3,14 \cdot 160} \cdot \ln \frac{2 \cdot 160^2}{0,03 \cdot 0,5} = 5,99, \text{ Ом}$$

Кількість заземлювачів захисного заземлення визначається за формулою:

$$n = \frac{2 \cdot R_{\xi}}{4 \cdot \eta_{\xi}} = \frac{2 \cdot 110}{4 \cdot 0,7} = 79 \text{ шт} \quad (4.6)$$

де 4 - допустимий загальний опір, Ом;

2 - коефіцієнт сезонності.

Визначаємо загальний опір захисного заземлення:

$$R_{ззп} = \frac{110 \cdot 5,99}{5,99 \cdot 79 \cdot 0,7 + 110 \cdot 0,5} = 1,7, \text{ Ом}$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{ззп} < 4 \text{ Ом}$.

4.7 Охорона навколишнього природного середовища

4.7.1 Загальні дані з охорони навколишнього природного середовища

Діяльність за темою магістерської роботи, а саме: Дослідження та протипування діалогових систем для віртуальних клієнтів (ботів) в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

В процесі діяльності розробника об'єктної моделі за допомогою ПК виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

Відпрацьовані люмінесцентні лампи - I клас небезпеки

Акумулятор для джерел безперебійного харчування -III клас небезпеки

Змінні носії інформації - IV клас небезпеки

Макулатура - IV клас небезпеки

Побутові відходи - IV клас небезпеки

4.7.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог [12].

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця тимчасового зберігання відходів відповідно до затвердженої схеми їх розміщення.

Не допускається зберігання відходів у невстановлених схемою місцях, а також перевищення норм тимчасового зберігання відходів.

Способи тимчасового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- Відходи I класу небезпеки зберігаються в герметичній тарі (сталеві бочки, контейнери). У міру наповнення тари з відходами закривають герметично сталевий кришкою;

- Відходи II класу небезпеки в залежності від агрегатного стану зберігаються в поліетиленових мішках, бочках, сховищах та інших видах тари, яка запобігає поширенню шкідливих речовин;

- Відходи III класу небезпеки зберігаються в тарі, яка забезпечує локалізацію зберігання, дозволяє виконувати вантажно-розвантажувальні і транспортні роботи і виключає поширення в ОС шкідливих речовин;

- Відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автотранспортом перевантажують у самоскид і доставляють на місце утилізації або захоронення;

4.7.3 Визначення впливу та заходів щодо поводження з відходами ІТ галузі

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів. Відомості про місце утворення та місце розташування відходів зазначаються та наводяться у таблиці 4.5.

Таблиця 4.5 - Відомості про місце утворення та місце розташування відходів

№ з/п	Код та найменування відходів за ДК -005-96	Технологічний процес або виробництво, де утворюються відходи / клас небезпеки	Місце розташування відходу, тара та її кількість, місткість, розміри у разі наявності майданчиків розташування відходів необхідно зазначити тип покриття та наявність даху)
-------	--	---	---

1	2	3	4
1	7710.3.1.26 Лампи люмінесцентні, та відходи, які містять ртуть, інші зіпсовані або відпрацьовані (Відпрацьовані ртутьвмісні люмінесцентні лампи)	1	Буд.24, кв. 3
2	7710.3.1.01 Макулатура паперова та картонна (Макулатура)		буд.24, кв. 3
3	Акумулятор для джерел безперебійного живлення	3	буд.24, кв. 3

4.8 Висновки до 4 розділу

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними.

4.8.1 Перелік посилань до 4 розділу

1. НПАОП 0.00-6.03-93 «Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві»
2. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці»
3. НАПБ Б.02.005-2003 «Типове положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України»
4. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»
5. ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»
6. НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин»
7. ГОСТ 12.1.030-81 «ССБТ. Електробезпечність .Захисне заземлення. Занулення»
8. ГОСТ 13109-97 «Електрична енергія. Сумісність технічних засобів віелектромагнітних. Норми якості електроенергопостачання загального призначення »
9. ДБН В.2.5-28:2015 «Природне і штучне освітлення»
10. НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою»
11. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»
12. ДСанПіН 2.2.7.029 «Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення».

ВИСНОВКИ

В дипломному проекті було проведено огляд предметної області розробки і швидкого прототипування діалогових систем для віртуальних клієнтів (ботів). Проведено аналіз та дослідження сучасних засобів створення віртуальних систем. Обґрунтовано необхідність та актуальність створення віртуальної діалогової системи, призначеної для надання інформації студентам про розклад занять, адресу ВНЗ, контактних номерів деканатів або контактів кафедри. Розробка освітньо-інформаційного чат-боту є актуальною, оскільки вже в усіх популярних ВНЗ в світі є чат-боти, які допомагають в роботі деканатам та викладачам. Кожному ВНЗ необхідні якісь інструкції для початківців (абітурієнтів). На початку семестру у кожного абітурієнта та студента виникає питання «Як доїхати до ВНЗ?», «О котрій годині відкривається бібліотека?», «Де будуть заняття по веб-технологіям?». Зазвичай абітурієнти звертаються з такими питаннями до викладачів або, набравшись сміливості, до старших студентів. Це означає, що на початку кожного навчального року у співробітників університетів з'являється безліч додаткових обов'язків, пов'язаних з наданням всієї необхідної інформації і відповідями на однакові питання нових студентів.

Існує значна кількість аналогів діалогових систем або сайтів компаній з чат-ботами, але недоліком використання є їх складність, і внаслідок цього проблеми супроводження.

Ознайомившись з вимогами користувачів, були сформульовані основні задачі та описані методи та засоби, що були обрані для реалізації програмного продукту.

Для досягнення мети в роботі сформульовані і вирішені наступні завдання:

- проаналізовано методи розробки діалогових систем та існуючі програмні засоби, які їх описують;
- визначено основні принципи роботи чат-ботів;
- проведено критичний аналіз статей та публікацій з тематики дослідження;
- проаналізовано сучасні вимоги та зміни в роботі діалогових систем із віртуальними клієнтами;
- виконані проектування та розробка діалогової системи;
- розроблено інформаційно-освітній чат-бот.

Розроблений віртуальний бот може застосовуватись абітурієнтами, студентами та викладачами СНУ ім. В. Даля для швидкого отримання якісної інформації.

СПИСОК ПОСИЛАНЬ

Електронні ресурси:

1. Засоби розробки / Електронний ресурс <https://dialogflow.com/docs/getting-started/basics> [Перевірено: 23.11.2018].
2. Засоби розробки /Електронний ресурс <https://developers.google.com/maps/documentation/> - api google maps [Перевірено: 23.11.2018].
3. Документація /Електронний ресурс <https://code.visualstudio.com/docs/editor/whyvscode> [Перевірено: 03.12.2018].
4. <https://www.ipipe.ru/info/javascript.html> [Перевірено: 23.11.2018].
5. Документація / Електронний ресурс Telegram docs <https://core.telegram.org/bots/api> [Перевірено: 23.11.2018].-
6. Засоби розробки /Електронний ресурс <https://azure.microsoft.com/en-us/services/cognitive-services/language-understanding-intelligent-service/> [Перевірено: 03.12.2018].
7. Документація / Електронний ресурс <https://miningbusinessdata.com/dialogflow-vs-lex-vs-luis-vs-watson-vs-chatfuel/> [Перевірено: 03.12.2018].
8. Засоби розробки /Електронний ресурс <https://chatbotslife.com/dialogflow-fulfillment-webhook-tutorial-f7aa024ea475?gi=31da9f0b6424> [Перевірено: 03.12.2018].
9. Бізнес-процеси // Ключові рішення [Електронний ресурс] <https://chatbotslife.com/tagged/artificial-intelligence> [Перевірено: 03.12.2018].

Книги:

10. Understanding Machine Learning: From Theory to Algorithms **Автори:** Shai Shalev-Shwartz and Shai Ben-David стр. 10-23
11. NLP SECRETS: Upgrade Your Mind **Автор:** Pete Casale стр. 40-43 та стр. 47-58
12. NLPforBeginners **Автор:** Невідомо стр. 30-32
13. Chatbots: Development and Applications **Автор:** Jorin Vogel 3. Applications 14-20
14. Reiter and Dale (2000) стр. 20-29

ДОДАТОК А

Лістинг програми

Файл Hello.json

```
{
  "id": "70ca73d7-fbaa-4c4a-a880-4a2de5a103ea",
  "name": "Hello",
  "auto": true,
  "contexts": [],
  "responses": [
    {
      "resetContexts": false,
      "action": "input.welcome",
      "affectedContexts": [],
      "parameters": [],
      "messages": [
        {
          "type": 0,
          "speech": []
        },
        {
          "type": 0,
          "speech": [
            "Привет! Я тот самый бот помощник! Могу тебе подсказать контакты вуза, сайт, электронное расписание. Пиши что тебя интересует 😊",
            "Здравствуй! Я тот самый бот помощник! Могу тебе подсказать контакты вуза, сайт, электронное расписание. Пиши что тебя интересует 😊",
            "Добрый день! Я тот самый бот помощник! Могу тебе подсказать контакты вуза, сайт, электронное расписание. Пиши что тебя интересует 😊"
          ]
        }
      ]
    },
    {
      "defaultResponsePlatforms": {
        "telegram": false
      },
      "speech": []
    }
  ],
  "priority": 500000,
  "webhookUsed": false,
  "webhookForSlotFilling": false,
  "lastUpdate": 1546885076,
  "fallbackIntent": false,
}
```

```
"events": [
  {
    "name": "WELCOME"
  },
  {
    "name": "TELEGRAM_WELCOME"
  }
],
"userSays": [
  {
    "id": "b6854ce7-d48c-4169-92e5-c20c44677a49",
    "data": [
      {
        "text": "привет привет"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  },
  {
    "id": "8495e622-402a-4a1b-9722-4da23e63db4a",
    "data": [
      {
        "text": "здравствуй"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  },
  {
    "id": "f4f17259-e08c-4b3e-961a-5b9455b43aa1",
    "data": [
      {
        "text": "привет"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  },
  {
    "id": "f7209b75-2387-4a3a-882b-08e3750db2bd",
    "data": [
      {
```

```
    "text": "приветствую"
  }
],
"isTemplate": false,
"count": 0,
"updated": 0,
"isAuto": false
},
{
  "id": "ff1d0fe0-88ea-4c90-9355-f570874aabf2",
  "data": [
    {
      "text": "всем привет"
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "0f600d72-cc87-4c8a-9bec-4d870bc4a094",
  "data": [
    {
      "text": "хей"
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "8d038926-b75a-4079-bcc3-dabcec455288",
  "data": [
    {
      "text": "чао"
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "68893851-581f-4bc6-973f-748b1bf32d47",
  "data": [
    {
      "text": "приветствую тебя"
```

```
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "71b1a013-b429-498b-996d-0cda734fbdb2",
  "data": [
    {
      "text": "приветик"
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "3f7b3866-0c81-4f6c-85af-ab65a6e243cb",
  "data": [
    {
      "text": "и снова здравствуйте"
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "25d1767d-3972-414f-828e-a6c5ae88eeee",
  "data": [
    {
      "text": "здравствуйте"
    }
  ],
  "isTemplate": false,
  "count": 0,
  "updated": 0,
  "isAuto": false
},
{
  "id": "fbb76d8f-8214-45e0-a93f-71fe86b740f2",
  "data": [
    {
      "text": "давно не виделись"
    }
  ]
}
```

```
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  },
  {
    "id": "9f9e09df-c008-4deb-9d18-463ba8ba6a1a",
    "data": [
      {
        "text": "просто хотел поздороваться"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  },
  {
    "id": "94cbaee4-e6be-4433-9721-a064ed663578",
    "data": [
      {
        "text": "хотелось бы тебя поприветствовать"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  },
  {
    "id": "215d8495-d815-4e17-937d-976baa7d1298",
    "data": [
      {
        "text": "хочу сказать тебе привет"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 0,
    "isAuto": false
  }
],
"followUpIntents": [],
"liveAgentHandoff": false,
"endInteraction": false,
"templates": []
}
```


Файл geo.json

```

{
  "id": "27eb10e6-dca9-4260-a5c2-f79f4c42410f",
  "name": "Geo",
  "auto": true,
  "contexts": [],
  "responses": [
    {
      "resetContexts": false,
      "action": "",
      "affectedContexts": [],
      "parameters": [],
      "messages": [
        {
          "type": 0,
          "platform": "telegram",
          "speech": https://goo.gl/maps/5ovZfB38jZU2 #API GOOGLE MAPS
        },
        {
          "type": 3,
          "platform": "telegram",
          "imageUrl":
https://pp.userapi.com/c844720/v844720323/16c717/lAGMb7NlFcI.jpg,
          "lang": "ru"
        },
        {
          "type": 0,
          "speech": []
        }
      ],
      "defaultResponsePlatforms": {
        "telegram": true
      },
      "speech": []
    }
  ],
  "priority": 500000,
  "cortanaCommand": {
    "navigateOrService": "NAVIGATE",
    "target": ""
  },
  "webhookUsed": true,
  "webhookForSlotFilling": false,
  "lastUpdate": 1546890855,
  "fallbackIntent": false,

```

```
"events": [
  {
    "name": "TELEGRAM_WELCOME"
  }
],
"userSays": [
  {
    "id": "7c916cf6-b749-49ae-a5ca-d39ab5989132",
    "data": [
      {
        "text": "Адрес",
        "meta": "@sys.ignore",
        "userDefined": true
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546890855,
    "isAuto": false
  },
  {
    "id": "01d58e5f-7199-4a49-aeb8-f69a8294a1f0",
    "data": [
      {
        "text": "Где",
        "meta": "@sys.ignore",
        "userDefined": true
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546890855,
    "isAuto": false
  },
  {
    "id": "4d4efce7-bcea-458f-986c-ecc0f1aac0fe",
    "data": [
      {
        "text": "Где находится?",
        "meta": "@sys.ignore",
        "userDefined": true
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546890855,
    "isAuto": false
  }
],
```

```
"followUpIntents": [],
"liveAgentHandoff": false,
"endInteraction": false,
"templates": []
```

Файл Contacts.json

```
{
  "id": "ccf08980-8f1c-492d-b971-52ed70ed6d65",
  "name": "Contacts",
  "auto": true,
  "contexts": [],
  "responses": [
    {
      "resetContexts": false,
      "affectedContexts": [],
      "parameters": [],
      "messages": [
        {
          "type": 0,
          "platform": "telegram",
          "speech": "Здесь вы можете найти контакты ВУЗа  

https://snu.edu.ua/?page\_id=21"
        },
        {
          "type": 3,
          "platform": "telegram",
          "imageUrl": "https://pp.userapi.com/c846320/v846320316/171877/QdttkRD8uWo.jpg",
          "lang": "ru"
        },
        {
          "type": 0,
          "speech": []
        }
      ],
      "defaultResponsePlatforms": {
        "telegram": true
      },
      "speech": []
    }
  ],
  "priority": 500000,
  "cortanaCommand": {
    "navigateOrService": "NAVIGATE",
    "target": ""
  },
  "webhookUsed": false,
```

```
"webhookForSlotFilling": false,
"lastUpdate": 1546638340,
"fallbackIntent": false,
"events": [
  {
    "name": "TELEGRAM_WELCOME"
  }
],
"userSays": [
  {
    "id": "a7ae4318-f877-41d5-b905-0a8a1da095ef",
    "data": [
      {
        "text": "Контакты ВУЗа"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546638340,
    "isAuto": false
  },
  {
    "id": "6ff168bf-24c2-4d5f-a26a-a68384ff3890",
    "data": [
      {
        "text": "Контакты"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546638340,
    "isAuto": false
  }
],
"followUpIntents": [],
"liveAgentHandoff": false,
"endInteraction": false,
"templates": []
}
```

Файл address.json

```
{
  "id": "1b344839-3b4e-4eee-8124-c8645222606f",
  "name": "Site",
  "auto": true,
  "contexts": [],
  "responses": [
    {
      "resetContexts": false,
      "action": "input.site",
      "affectedContexts": [],
      "parameters": [
        {
          "id": "2d009109-d742-4e9e-ac2a-436067c7b879",
          "required": true,
          "dataType": "@sys.url",
          "name": "url",
          "value": "$url.original",
          "prompts": [
            "http://snu.edu.ua/"
          ],
          "isList": false
        }
      ],
      "messages": [
        {
          "type": 0,
          "platform": "telegram",
          "speech": []
        },
        {
          "type": 0,
          "speech": []
        }
      ],
      "defaultResponsePlatforms": {
        "telegram": false
      },
      "speech": []
    }
  ],
  "priority": 500000,
  "cortanaCommand": {
    "navigateOrService": "NAVIGATE",
    "target": ""
  }
}
```

```
},
"webhookUsed": false,
"webhookForSlotFilling": false,
"lastUpdate": 1546639833,
"fallbackIntent": false,
"events": [],
"userSays": [
  {
    "id": "8ba3da22-4d86-47d3-9a84-79ee5f3e9545",
    "data": [
      {
        "text": "вну СЧУ им даля"
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546630618,
    "isAuto": false
  },
  {
    "id": "afd1c100-102b-4b95-b287-e9e44653f4ac",
    "data": [
      {
        "text": "сайт",
        "alias": "url",
        "meta": "@sys.url",
        "userDefined": true
      }
    ],
    "isTemplate": false,
    "count": 0,
    "updated": 1546630618,
    "isAuto": false
  }
],
"followUpIntents": [],
"liveAgentHandoff": false,
"endInteraction": false,
"templates": []
}
```

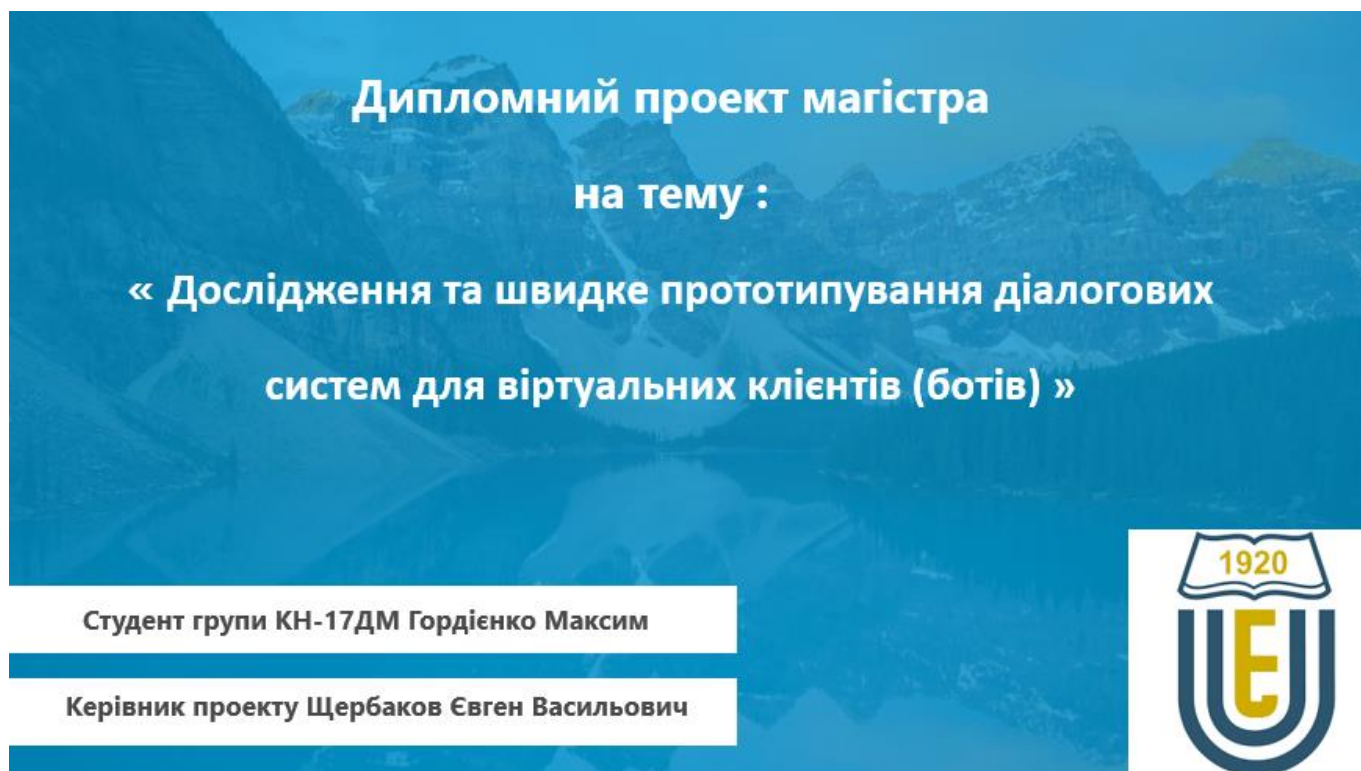
ДОДАТОК Б

Рисунок Б.1 – «Титульний лист»

Метою даного проекту є:

1. Проведення аналізу та обґрунтування необхідності розробки
2. Структура та архітектура роботи чат-боту
3. Порівняння та вибір діалогової системи
4. Штучний інтелект: наука та створення інтелектуальних програм
5. Створення інформаційного чат-боту
6. Реліз чат-боту для студентів із використанням діалогової системи Dialog Flow
7. Висновок

Рисунок Б.2 – «Мета проекту»

1. Проведення аналізу та обґрунтування необхідності розробки

Діалогові системи (агенти), орієнтовані на конкретні завдання - призначені для конкретного завдання і обробляють короткі бесіди, щоб отримати інформацію від користувача для виконання завдання. Вони включають в себе інтерактивні мобільні цифрові помічники (Siri, Cortana, Alexa, Google Assistant та інші).

Чат-боти - це системи, що імітують мовну поведінку людини при тривалому спілкуванні з одним або декількома співрозмовниками, на відміну від діалогових систем, не вирішують якусь конкретну задачу. Ці системи часто мають розважальну функцію або можуть використовуватись для допомоги студенту у отриманні інформації швидко та якісно.

Обґрунтовано необхідність та актуальність створення віртуальної діалогової системи, призначеної для надання інформації студентам про розклад занять, адресу вузу, контактних номерів викладачів (ел.пошти) тощо.

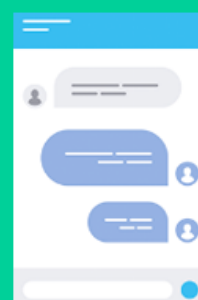


Рисунок Б.3 – «Проведення аналізу та обґрунтування необхідності розробки»

2. Структура та архітектура роботи чат-боту

Для того, щоб зрозуміти як працює чат бот, потрібно спочатку поглянути на загальну архітектуру.

Обробка природних мов (NLP) використовується для опису здатності машини приймати те, що сказано, розібрати, зрозуміти його значення, визначити відповідні дії та відповісти на мові, яку користувач зрозуміє. Розуміння мови (NLU) - це частина обробки природної мови, що стосується набагато вужчого, але не менш важливого аспекту того, як найкраще справлятися з неструктурованими входами та перетворювати їх у структуровану форму, яку машина може зрозуміти та відтворити. Генерація природної мови (NLG) – перетворення структурованих даних в текст. На рис. 1 зображена схема роботи чат-ботів.

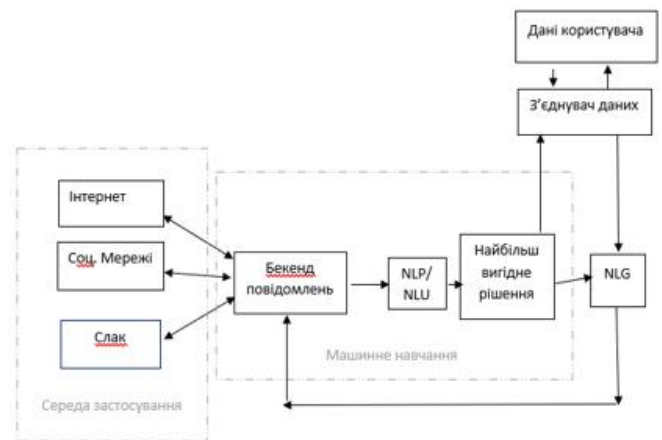


Рисунок Б.4 – «Структура та архітектура чат-боту»

3. Порівняння та вибір діалогової системи

Найпопулярніші діалогові системи:

- LUIS
- Watson Conversation
- API.ai (Dialog Flow)
- wit.ai
- Amazon Lex

NLU	Намір и	Об'єкти	Пакетний імпорт
LUIS	+	+	+
Watson	+	+	+
API.ai	+	+	+
wit.ai	+	+	0
Lex	+	0	-

На рис.2 зображено таблицю порівняння основної функціональності, що пропонується різними службами. Всі вони, за винятком Amazon Lex, поділяють одну і ту ж основну концепцію: на основі прикладних даних користувач може навчити класифікатор класифікувати так звані наміри (які представляють намір всього повідомлення і не пов'язані з певною позицією в межах повідомлення) і об'єктів (які можуть складатися з одного або декількох символів).

У якості діалогової системи було обрано платформу Dialogflow, так як вона безкоштовна для розробки діалогових систем для додатків, веб-сайтів, платформ обміну повідомленнями і пристроїв, а також Інструменти розуміння природної мови допомагають створювати унікальні сценарії діалогів, з відповідними діями, і аналізувати взаємодію з користувачами.

Рисунок Б.5 – «Порівняння та вибір діалогової системи»

4. Штучний інтелект: наука та створення інтелектуальних програм

Штучний інтелект (ШІ, англ. Artificial intelligence, AI) – це велике міждисциплінарний напрямок, завданням якого є моделювання і формалізація інтелектуальних дій людини. Застосування штучного інтелекту для посилення комунікативних здібностей інформаційних систем призвело до появи систем з інтелектуальним інтерфейсом, серед яких можна виділити наступні типи.



Інтелектуальні бази даних.
Дозволяють на відміну від традиційних БД забезпечувати вибірку необхідної інформації, що не присутньої в явному вигляді, а виводиться з сукупності збережених даних.



Природно-мовний інтерфейс.
Застосовується для доступу до інтелектуальних баз даних, контекстного пошуку документальної текстової інформації, голосового введення команд в системах управління, машинного перекладу з іноземних мов.

Рисунок Б.6 – «Штучний інтелект: наука та створення інтелектуальних програм»

1

5. Створення інформаційного чат-боту

Чат-бот розробляється на мові JavaScript за допомогою діалогової системи Dialog Flow для студентів та викладачів, котрим потрібно дізнатись ту чи іншу інформацію про ВУЗ використовуючи додаток на телефоні (наприклад адресу, контакти, електроний розклад занять, веб-сайт, соціальні сторінки, тощо).

Схема роботи чат-боту представлена на рис. 3



Діаграма варіантів використання чат-боту представлена на рис. 4

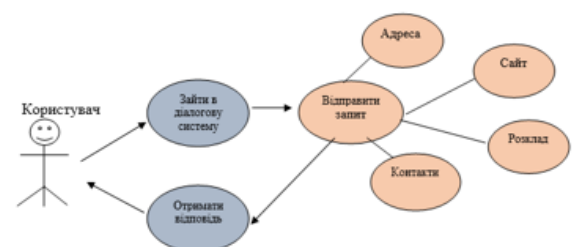



Рисунок Б.7 – «Створення інформаційного чат-боту»

6. Реліз чат-боту для студентів із використанням діалогової системи Dialog Flow


Dialog flow зручна, інтуїтивно зрозуміла платформа яка використовує нейро-лінгвістичне програмування і підходить для створення чат-ботів за допомогою інтеграції з Telegram.

Загальна схема роботи сервісу Dialog Flow представлена на рис. 5



The diagram illustrates the Dialog Flow architecture. It is divided into three main stages: APP/DEVICE, AI PLATFORM, and FULFILLMENT. A USER interacts with the APP/DEVICE, which provides INPUT (text and voice) and receives OUTPUT (text and voice). The INPUT is processed by the AI PLATFORM, which includes a QUERY step and an INTENT step. The INTENT step then triggers the FULFILLMENT stage, which involves EXTERNAL APIs and a DATABASE (DB). The FULFILLMENT stage also feeds back into the AI PLATFORM via ACTIONABLE DATA.

Приклад того як працює чат-бот в Telegram рис.6



The screenshot shows a Telegram chat interface with a bot named 'VNUbot'. The bot's profile picture is a daisy. The chat history includes:

- Bot: "Что может делать этот бот?"
- User: "Справочник для студентов ВНУ им. В.Даня"
- User: "У шкору"
- User: "привет 23:17 ✓"
- Bot: "Доброй день! Я тот самый бот помощник! Могу тебе подсказать контакты вуза, сайт, электронное расписание. Пиши, что тебя интересует 😊"
- User: "Покажи мне расписание на завтра 23:17 ✓"
- Bot: "Здесь ты можешь выбрать свою группу и посмотреть расписание на неделю <https://vnu.edu.ua/ru/diary/>"

Рисунок Б.8 – «Реліз чат-боту»

7. Висновок

В дипломному дослідженні було проведено огляд предметної області, що стосується теми проекту дослідження і швидкого прототипування діалогових систем для віртуальних клієнтів (ботів). Проведено аналіз та дослідження сучасних засобів створення віртуальних систем. Обґрунтовано необхідність та актуальність створення віртуальної діалогової системи, призначеної для надання інформації студентам про розклад занять, адресу вузу, контактних номерів. Розробка такого чат-боту є актуальною, оскільки в наш час всі користуються чат-ботами в месенджерах.

Рисунок Б.9 – «Висновки»



Б.10 – «Кінець»