

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

***«Розробка програмної оболонки системи акустичного
розпізнавання та обробки аудіо-інформації»***

Студента 2 курсу, 6м групи,
спеціальності 121 «Інженерія
програмного забезпечення»,
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

**Костенко Артема
Андрійовича**

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

**Рассамакін
Володимир Якович**

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

**Криворучко Олена
Володимирівна**

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

8 листопада 2019 р.

Завдання

на випускний кваліфікаційний проект студентів

Костенко Артему Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка програмної
оболонки системи акустичного розпізнавання та обробки аудіо-інформації»

Затверджена наказом ректора від "13" грудня 2019 р. № 4304

2. Строк здачі студентом закінченого проекту 01 грудня 2020 р.

3. Цільова установка та вихідні дані до проекту

Мета проекту досягти розпізнавання комп'ютером мовлення користувача та
допомога користувачу виводити його мовлення в текст.

Об'єкт дослідження процес розробки програмної оболонки розпізнавання
мовлення

Предмет дослідження розробка комп'ютерної програми для виведення
мовлення користувача у текст.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМИ

1.1. Загальні принципи розпізнавання мовлення

1.2. Аналіз програмних засобів розпізнавання мовлення

1.3. Висновки до першого розділу

РОЗДІЛ 2. АНАЛІЗ ТА ВИБІР МОВИ ПРОГРАМУВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

2.1. Мова програмування Python

2.2. Python та його модулі

2.3. Використання PyQt5 в середовищі програмування Python

2.4. Висновки до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Перелік програмних модулів

3.2. Розробка програмного коду функціональної частини

3.3. Створення зовнішньої оболонки програми

3.4. Поеднання логіки програми та оболонки.

3.5. Створення інсталятора програми

3.6. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	20.09.2019	20.09.2019
2.	<i>Розробка та затвердження завдання на проект магістра</i>	08.11.2019	08.11.2019
3.	<i>Вступ та перелік літературних джерел</i>	24.01.2020	24.01.2020
4.	<i>Наукова стаття</i>	01.09.2020	01.09.2020
5.	<i>Технічне завдання</i>	28.02.2020	28.02.2020
6.	<i>Розділ 1. Обґрунтування вибору програмної платформи CRM-систем</i>	25.06.2020	25.06.2020
7.	<i>Розділ 2. Проектування системи</i>	07.09.2020	07.09.2020
8.	<i>Розділ 3. Реалізація системи</i>	19.10.2020	19.10.2020
9.	<i>Програма та методика тестування</i>	21.10.2020	21.10.2020
10.	<i>Керівництво користувача</i>	23.10.2020	23.10.2020
11.	<i>Висновки та пропозиції</i>	30.10.2020	30.10.2020
12.	<i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i>	05.11.2020	05.11.2020
13.	<i>Підготовка автореферату та презентації доповіді</i>	05.11.2020	05.11.2020
14.	<i>Попередній захист випускного кваліфікаційного проекту</i>	25.11.2020- 27.11.2020	25.11.2020 -27.11.2020
15.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.12.2020	01.12.2020
16.	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	08.12.2020- 09.12.2020	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту Рассамакін В. Я.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Криворучко О. В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Костенко А. А.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист _____ Рассамакін В. Я.
(ПІБ, підпис, дата)

Випускний кваліфікаційний проект студента Костенко А. А
(прізвище, ініціали)

« » 20 p.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці комп'ютерної програми, що дозволить перетворення аудіоінформації до тексту.

Розробка програмного продукту виконана у середовищі розробки Python, з допомогою Google Speech Api. Обрана мова програмування в обох випадках – Python. Для напису коду програмного продукту, та його компілювання, було використано програму Sublime Text 3.

Готовий програмний комплекс «Розпізнавання мовлення» було успішно опротестовано відповідно до функціональних вимог.

ABSTRACT

In accordance with the purpose of the study, the work is devoted to the development of a computer program that will convert audio information into text.

Software development is performed for Python development environments, using Google Speech Api. The chosen programming language in both cases is Python. Sublime Text 3 was used to write the product program code and compile page.

The finished software program "Speech Recognition" was successfully tested in accordance with the functional requirements.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ОС – операційна система

RAM (англ. Random Access Memory) – оперативна пам'ять, об'єм ОЗП

USB (англ. Universal Serial Bus) – універсальний послідовний інтерфейс для з'єднання комп'ютерів і периферійних пристроїв

IDE (англ. Integrated Development Environment) – інтегроване середовище розробки

SDK (англ. Software Development Kit) – набір із засобів розробки, утиліт і документації до певної технології або платформи

XML (англ. Extensible Markup Language) – розширювана мова розмітки

CMD (англ. Command Promt) – командний рядок, за допомогою якого можна керувати операційною системою Windows та вносити зміни у її налаштування.

					<i>КНТЕУ 121 06-12.МР</i>		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		19.10.20	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	<i>Стадія</i>	<i>Аркуш</i>
Керівник		Рассамакін В.Я.		19.10.20		<i>ПС</i>	<i>2</i>
Гарант		Криворучко О.В.		19.10.20		<i>Факультет інформаційних технологій</i>	
Розробив		Костенко А.А.		19.10.20			
					<i>Перелік умовних скорочень</i>	<i>2м курс, 6 група</i>	

ЗМІСТ	
ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМИ.....	6
1.1. Загальні принципи розпізнавання мовлення.....	6
1.2. Аналіз програмних засобів розпізнавання мовлення.....	7
1.3. Висновки до першого розділу	11
РОЗДІЛ 2. АНАЛІЗ ТА ВИБІР МОВИ ПРОГРАМУВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	12
2.1. Мова програмування Python.....	12
2.2. Python та його модулі.....	13
2.3. Використання PyQt5 в середовищі програмування Python	15
2.4. Висновки до розділу 2.....	16
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	17
3.1. Перелік програмних модулів.....	17
3.2. Розробка програмного коду функціональної частини	17
3.3. Створення зовнішньої оболонки програми	21
3.4. Поєднання логіки програми та оболонки.....	23
3.5. Створення інсталятора програми.....	26
3.6. Висновок до розділу 3	28
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
ТЕХНІЧНЕ ЗАВДАННЯ	32
ДОДАТКИ	34

					<i>КНТЕУ 121 06-12.МР</i>		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		19.10.20	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	<i>Стадія</i>	<i>Аркуш</i>
Керівник		Рассамакін В.Я.		19.10.20		<i>Зміст</i>	<i>31</i>
Гарант		Криворучко О.В.		19.10.20		<i>Факультет інформаційних технологій</i>	
Розробив		Костенко А.А.		19.10.20			
					<i>Зміст</i>	<i>2м курс, 6 група</i>	

ВСТУП

Актуальність. Потужний успіх таких продуктів із підтримкою мовлення, як Amazon Alexa, не лише довела, що певна ступінь мовної підтримки буде важливим аспектом побутових технологій у найближчому майбутньому. Якщо задуматися, то є причини, чому це досить очевидно. Включення розпізнавання мови пропонує рівень інтерактивності та доступності, якому може відповідати лише мала кількість технологій.

Розпізнавання мови дозволяє людям похилого віку та людям з вадами зору швидко та природно взаємодіяти з ультрасучасними продуктами та послугами - графічний інтерфейс не потрібен!

Як працює розпізнавання мови, які пакети доступні на PyPI, як встановити та використовувати пакет SpeechRecognition - повнофункціональна та проста у використанні бібліотека розпізнавання мови Python.

Мета дослідження: аналіз створених програмних продуктів та створення програмної оболонки для розпізнавання.

Об'єкт дослідження: процеси та програмні засоби розпізнавання мовлення.

Предмет дослідження: розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації.

У відповідності з метою дослідження поставлені наступні завдання:

- 1) дослідити принципи роботи вже існуючих програмних рішень;

					<i>КНТЕУ 121 06-12.МР</i>		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		19.10.20	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	<i>Стадія</i>	<i>Аркуш</i>
Керівник		Рассамакін В.Я.		19.10.20		<i>В</i>	<i>4</i>
Гарант		Криворучко О.В.		19.10.20		<i>Факультет інформаційних технологій</i>	
Розробив		Костенко А.А.		19.10.20			
					<i>Вступ</i>	<i>2м курс, 6 група</i>	

- 2) визначити методи створення програмного продукту на базі мови Python;
- 3) обґрунтувати актуальність створеного програмного продукту;
- 4) проаналізувати та обґрунтувати вибір програмного забезпечення та технологій, що використовуватимуться у процесі розробки;
- 5) провести тестування готового рішення на реальних пристроях та розробити інструкцію користувача.

Методи дослідження, що були використані у роботі: аналіз, абстрагування, програмування, порівняння, графічний метод і моделювання.

Наукова новизна дослідження полягає в простоті використання, порівняно з аналогічними рішеннями.

Практичне значення дослідження: готове альтернативне рішення у вигляді програмного додатку операційного середовища Windows, що здатне на відтворення акустичних сигналів мовлення у вигляді текстової інформації.

					КНТЕУ 121 06-12.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМИ

1.1. Загальні принципи розпізнавання мовлення

Розпізнавання мови, здатність пристроїв реагувати на вимовлені команди. Розпізнавання мовлення дає можливість вільного керування різними пристроями та обладнанням (особливо корисною є для багатьох інвалідів), забезпечує автоматичний переклад та створює готові до друку диктанти.

Серед найбільш ранніх програм для розпізнавання мови були автоматизовані телефонні системи та програмне забезпечення для медичного диктування. Вони часто використовуються для диктування, запитів до баз даних і для передачі команд комп'ютерним системам, особливо в професіях, які покладаються на спеціалізовані словникові запаси. Це також забезпечує особистих помічників у транспортних засобах та смартфонах, таких як Apple Siri.

Перш ніж будь-яка машина зможе інтерпретувати мову, мікрофон повинен перевести вібрації голосу людини у хвилеподібний електричний сигнал. Цей сигнал, у свою чергу, перетворюється апаратним забезпеченням системи - наприклад, звуковою картою комп'ютера - у цифровий сигнал. В свою чергу цифровий сигнал аналізує програма розпізнавання мови, щоб розпізнати окремі фонemi, основні будівельні блоки мови.

Потім фонemi поєднуються у слова. Однак багато слів звучать однаково, і, щоб вибрати відповідне слово, програма повинна спиратися на

					<i>КНТЕУ 121 06-12.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			25.06.20	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Рассамакін В.Я.			25.06.20		<i>РІ</i>	<i>6</i>	<i>31</i>
Гарант	Криворучко О.В.			25.06.20		<i>Факультет інформаційних технологій 2м курс, 6 група</i>		
Розробив	Костенко А.А.			25.06.20				

контекст. Багато програм встановлюють контекст за допомогою триграмного аналізу - методу, заснованого на базі даних частих кластерів із трьох слів, у яких присвоюється ймовірність того, що за будь-якими двома словами слідуватиме дане третє слово. Наприклад, якщо доповідач каже « who I am », з англ « хто я такий », наступне слово буде розпізнано як займенник « I », а не схоже, але менш вірогідне « eye » з англійської « око ». Тим не менше, для виправлення помилок часом потрібне втручання людини.

Програми для розпізнавання кількох ізольованих слів, такі як телефонні голосові навігаційні системи, працюють майже для кожного користувача. З іншого боку, програми безперервного мовлення, такі як програми для диктантів, повинні бути навчені розпізнавати мовленнєві зразки людини; навчання передбачає читання користувачем вголос зразків тексту. Сьогодні, із зростанням потужності персональних комп'ютерів та мобільних пристроїв, точність розпізнавання мови помітно покращилася. Рівень помилок зменшився приблизно до 5 відсотків у словникових запасах, що містять десятки тисяч слів. Ще більшої точності досягають обмежені словникові запаси для таких спеціалізованих програм, як диктування рентгенологічних діагнозів.

1.2. Аналіз програмних засобів розпізнавання мовлення

Першою програмою стане Google Docs Voice Typing. Більшість авторів щодня покладаються на Документи Google, це безкоштовний, але потужний інструмент диктування. Це не тільки дозволяє вам друкувати своїм голосом у Документах Google, але ви також можете використовувати понад 100 голосових команд для редагування та форматування вашого документа майже всіма можливими способами, наприклад, нанесенням маркерів, курсивом або виділенням жирного тексту та переміщенням курсора до різних частин вашого документа.

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата		

Щоб використовувати голосовий ввід у Документах Google, все, що вам потрібно зробити, це натиснути на «Інструменти», «Голосовий набір» і дозволити Google отримати доступ до мікрофона вашого ноутбука.



Рис.1.1. Зображення Google Docs Voice Typing.

Dragon Professional Individual є найпопулярніше програмне забезпечення для диктувань у своєму просторі, Dragon Professional Individual дозволяє виконувати більшу частину своєї роботи, навіть не торкаючись клавіатури.

Використовуючи свою технологію глибокого навчання, яка адаптується до вашого конкретного голосу, жаргону та робочого середовища, ви можете створювати та редагувати документи з 99% точністю, швидко та точно перекладати записи в текст та використовувати голосові команди для керування діями вашого комп'ютера над програмами та веб-браузерерами, також надсилання електронного листа або створення звіту.

Dragon також інтегрується з Microsoft Office, iWork та іншими популярними бізнес-додатками, завдяки чому користувач може виконувати роботу в режимі "вільні руки", не залишаючи улюблених програм.

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

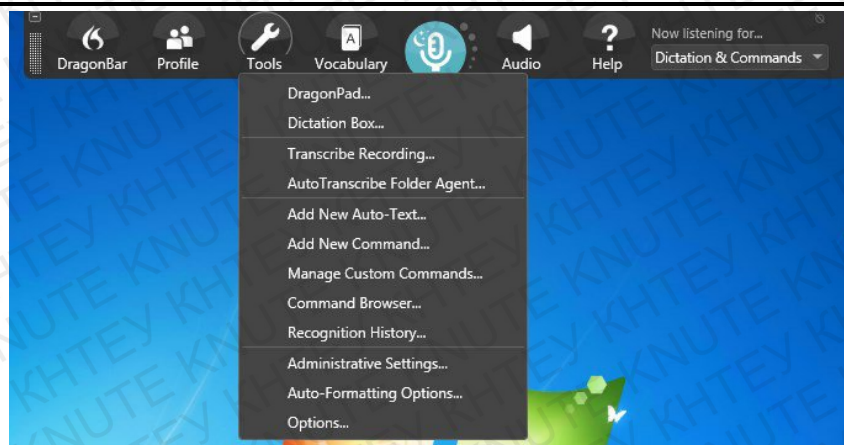


Рис.1.2 Зображення інтерфейсу програми Dragon Professional Individual

Braina Pro - це більше, ніж програмне забезпечення для розпізнавання мови - це персональний віртуальний помічник, який працює на основі штучного інтелекту. Окрім здатності диктувати понад 100 мов, Braina Pro може автоматизувати різні комп'ютерні завдання, встановлювати будильники та нагадування, повідомляти про поточні події, виконувати функції словника та тезауруса, відтворювати пісні та відео, читати електронні книги вголос, шукати файли в папках на комп'ютері тощо.

Braina Pro також підтримує вбудовані мікрофони більшості ноутбуків та має мобільний додаток, завдяки чому ви можете працювати з гарнітурою безкоштовно та далеко від комп'ютера.

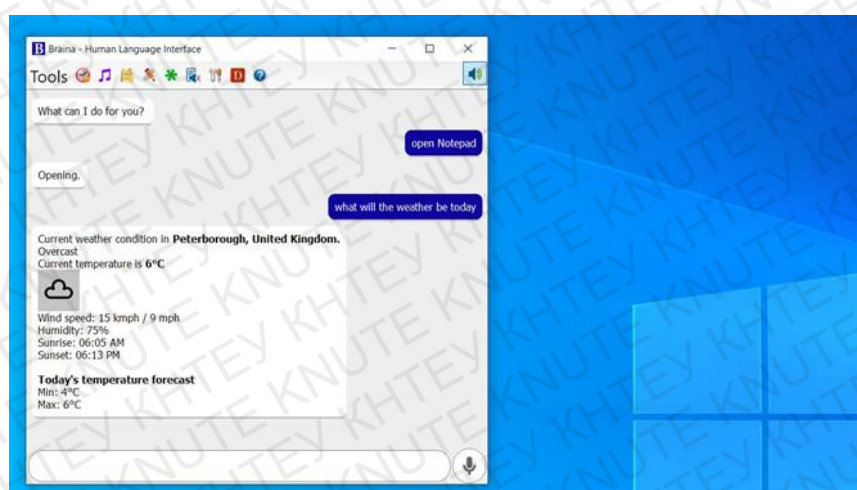


Рис. 1.3 Зображення роботи з програмою Braina Pro

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

Speechnotes ще одна програма для розпізнавання мовлення. Розроблений на механізмах розпізнавання мовлення Google, Speechnotes - це простий, чистий Інтернет-інструмент для диктування, який допомагає користувачам перекладати свою мову в текст із точністю понад 90%. Оскільки вам не потрібно завантажувати, встановлювати чи реєструвати Speechnotes, це один із найдоступніших інструментів для диктування.

Speechnotes також неймовірно зручний для користування - він автоматично пише великі літери на початку речень, автоматично зберігає документи та дозволяє одночасно диктувати та друкувати. Закінчивши диктувати Speechnotes, ви можете надіслати документи електронною поштою або роздрукувати, експортувати їх на Google Диск або завантажити на свій комп'ютер.

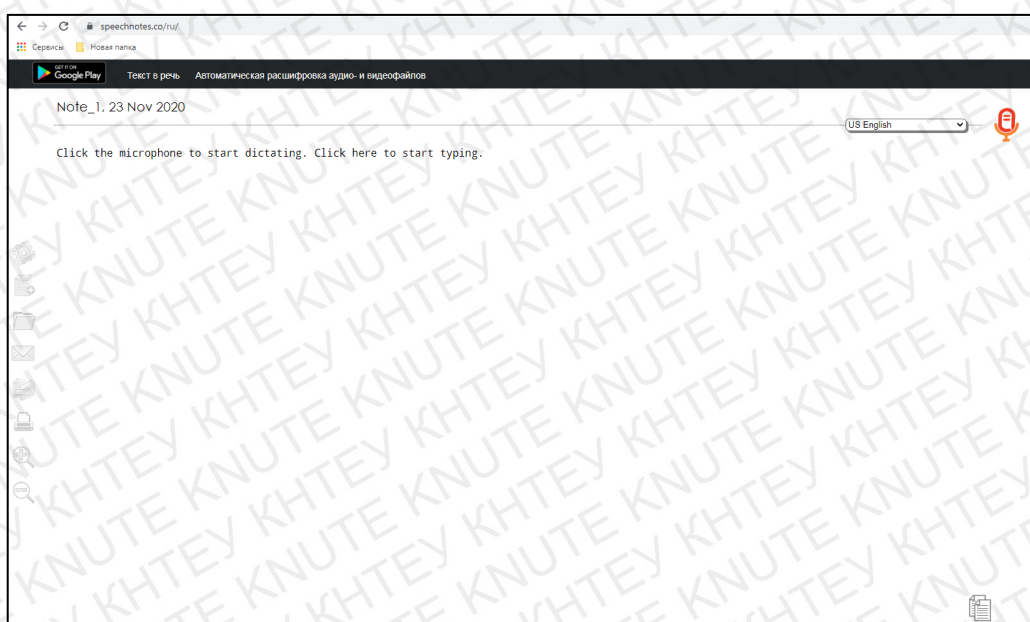


Рис. 1.4 Зображення сервісу Speechnotes

За допомогою інтерфейсу мовної програми Microsoft та .Net Framework e-Speaking дозволяє за допомогою голосу керувати діями комп'ютера, диктувати документи та електронні листи та змусити комп'ютер читати текст

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

вголос. Програмне забезпечення має понад 100 вбудованих команд, таких як «відкрити Інтернет» або «відкрити Excel». Ви також можете тренувати слова та додавати більше команд на e-Speaking, щоб отримати ще більше контролю над своїм комп'ютером.

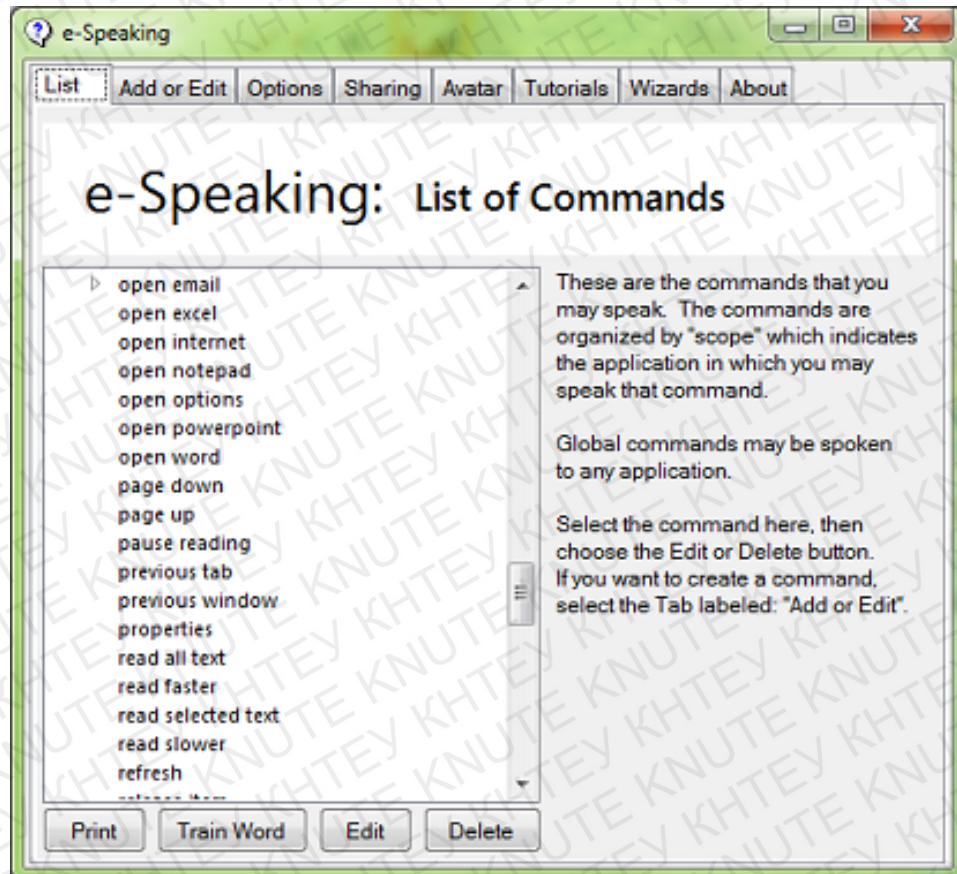


Рис. 1.5 Зображення інтерфейсу програми на e-Speaking.

1.3. Висновки до першого розділу

Отже, в цьому розділі було розглянуто не тільки принцип роботи розпізнавання мовлення, а також були представленні відомі продукти, які в свою чергу, в багатьох випадках є платними. При проведенні аналізу було вирішено створити свій програмний продукт для розпізнавання мовлення і особливістю його буде в тому, що він безкоштовний і простий у застосуванні.

					КНТЕУ 121 06-12.МР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

АНАЛІЗ ТА ВИБІР МОВИ ПРОГРАМУВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

2.1. Мова програмування Python

Python - інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Високорівневі вбудовані структури даних у поєднанні з динамічним набором тексту та динамічним прив'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів.

Простий, легкий у вивченні синтаксис Python підкреслює читабельність і, отже, зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно поширюватися.

Часто програмісти закохуються в Python через підвищену продуктивність, яку він забезпечує. Оскільки не існує кроку компіляції, цикл редагування-тестування-налагодження неймовірно швидкий. Налагодження програм Python дуже проста: помилка або неправильний ввід ніколи не спричинить помилку сегментації. Натомість, коли інтерпретатор виявляє помилку, виникає виняток. Коли програма не вловлює виняток, інтерпретатор друкує трасування стека. Налагоджувач рівня джерела дозволяє перевіряти локальні та глобальні змінні, оцінювати довільні вирази, встановлювати точки

					<i>КНТЕУ 121 06-12.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			07.09.20	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Рассамакін В.Я.			07.09.20		<i>P2</i>	<i>12</i>	<i>33</i>
Гарант	Криворучко О.В.			07.09.20		<i>Факультет інформаційних технологій 2м курс, 6 група</i>		
Розробив	Костенко А.А.			07.09.20				
					<i>Вибір програмних засобів та проектування</i>			

зупинки, переходити через код за рядком за один раз тощо. Налагоджувач написаний на самому Python, що свідчить про інтроспективну силу Python.

Python - це мова програмування загального призначення, що є ще одним способом сказати, що її можна використовувати майже для всього. Найголовніше, що це інтерпретована мова, а це означає, що написаний код фактично не перекладається в машиночитаний формат під час виконання. У той час як більшість мов програмування виконують це перетворення ще до запуску програми. Цей тип мови також називають "мовою сценаріїв", оскільки спочатку він задумувався для використання в тривіальних проектах.

Поняття "мови сценаріїв" значно змінилося з моменту її створення, оскільки Python тепер використовується для написання великих комерційних програм, а не просто банальних. Ця залежність від Python зросла ще більше, оскільки Інтернет набув популярності. Переважна більшість веб-додатків та платформ покладаються на Python, включаючи пошукову систему Google, YouTube та систему веб-транзакцій Нью-Йоркської фондової біржі (NYSE). Ви знаєте, що мова повинна бути досить серйозною, коли вона керує біржовою системою.

NASA насправді використовує Python, коли вони програмують своє обладнання та космічну техніку.

Python також може використовуватися для обробки тексту, відображення чисел або зображень, вирішення наукових рівнянь та збереження даних. Коротше кажучи, він використовується за лаштунками для обробки багатьох елементів, які можуть вам знадобитися або зустрітись на ваших пристроях, включаючи мобільний.

2.2. Python та його модулі

Модульне програмування - це процес розбиття великого, громіздкого завдання програмування на окремі, менші, більш керовані підзадачі або

					КНТЕУ 121 06-12.МР	Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата		

модулі. Потім можна створити окремі модулі, як будівельні блоки, щоб створити більший додаток.

Модуляризація коду у великому додатку має кілька переваг:

- *Простота:* Замість того, щоб зосередитись на цілій проблемі, що розглядається, модуль, як правило, фокусується на одній відносно невеликій частині проблеми. Якщо ви працюєте над одним модулем, у вас буде менше проблемних ділянок, які можуть з'явитися. Це полегшує розробку та меншу схильність до помилок.
- *Ремонтпридатність:* Модулі, як правило, розробляються таким чином, що вони забезпечують логічні межі між різними проблемними ділянками коду. Якщо модулі написані таким чином, щоб мінімізувати взаємозалежність, то зменшується ймовірність того, що модифікації одного модуля вплинуть на інші частини програми. (Ви навіть можете вносити зміни до модуля, не маючи жодних знань про програми поза цим модулем.) Це робить команду багатьох програмістів більш життєздатною для спільної роботи над великим додатком.
- *Багаторазове використання:* Функціональність, визначена в одному модулі, може бути легко повторно використана (через відповідним чином визначений інтерфейс) іншими частинами програми. Це позбавляє від необхідності відтворювати повторюваний код.
- *Сфера дії:* Модулі зазвичай визначають окремий простір імен, що допомагає уникнути зіткнень між ідентифікаторами в різних областях програми.

Функції, модулі та пакети - це всі конструкції в Python, що сприяють модуляризації коду.

					КНТЕУ 121 06-12.MP	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

2.3. Використання PyQt5 в середовищі програмування Python

PyQt - це бібліотека, яка дозволяє використовувати фреймворк графічного інтерфейсу Qt від Python. Сам Qt написаний на C++. Використовуючи його з Python, ви можете створювати програми набагато швидше, не жертвуючи значною частиною швидкості C++.

PyQt5 відноситься до останньої версії 5 Qt. Ви все ще можете випадково згадати про (Py) Qt4 в Інтернеті, але він застарів і більше не підтримується.

Цікавим новим конкурентом PyQt є Qt для Python. Його API практично ідентичний. На відміну від PyQt, він ліцензований під LGPL і, отже, може безкоштовно використовуватися в комерційних проектах.

Оскільки API настільки подібні, ви можете легко переробити свої програми на Qt для Python пізніше.

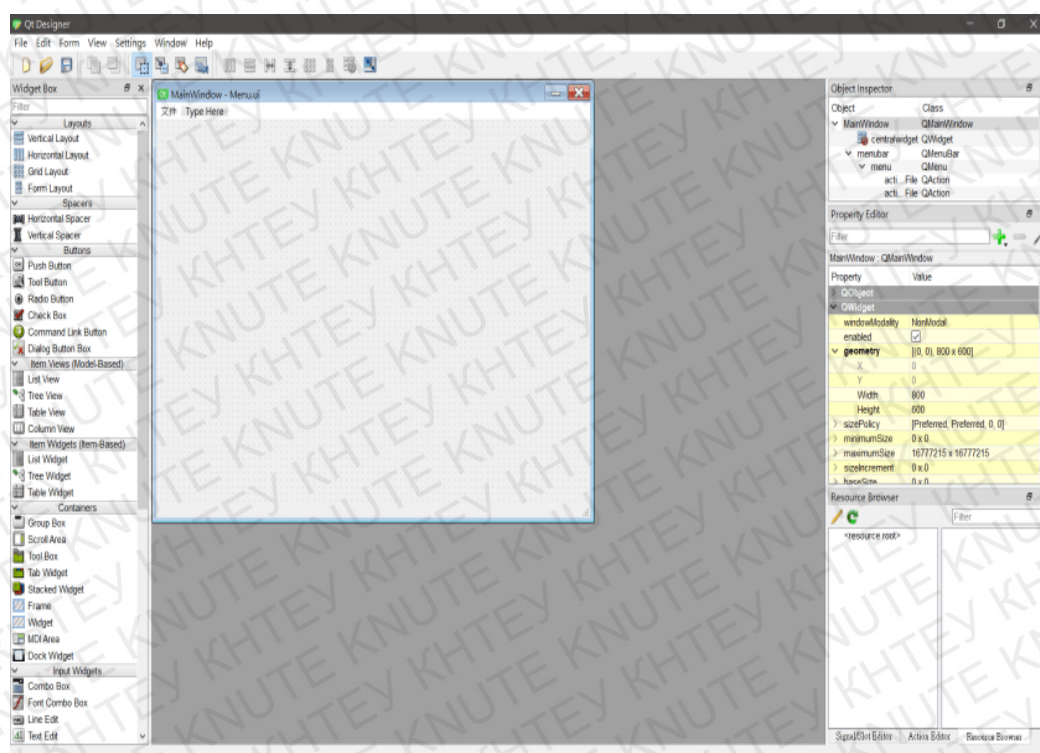


Рис. 2.1 Інтерфейс PyQt5

					КНТЕУ 121 06-12.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

2.4. Висновки до розділу 2

Отже, підсумовуючи вище викладене, можна визначити, що для створення програмного продукту буде використано модульне програмування на Python з використанням готових модулів, які складуть графічну частину та саму логіку програми.

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Перелік програмних модулів

Як вже зазначалося у попередніх розділах, робота програмного продукту для розпізнавання мовлення побудована на зв'язці мікрофона та модулю розпізнавання. Рисунок 3.1 схематично описує механізм роботи програми та обґрунтовують розробку окремих модулів, що представляють собою серверну частину та клієнтську частину програмного забезпечення.

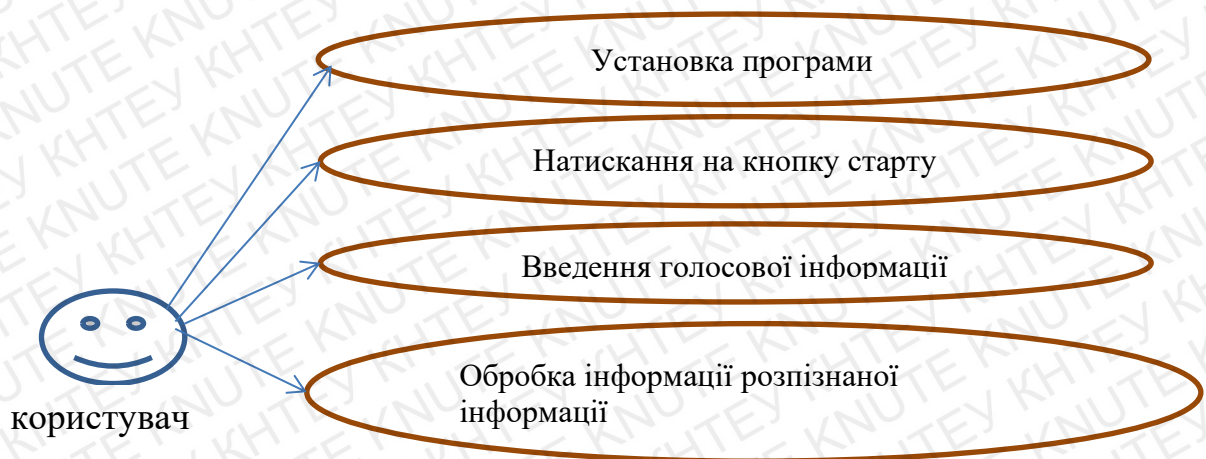


Рис. 3.1. Use-cases клієнтської сторони Розпізнавання мовлення

3.2. Розробка програмного коду функціональної частини

На початку, треба встановити Python, в нашому випадку була встановлена версія 3.8. Та, для комфортного введення коду та структурування, було встановлено SubLime Text 3.

					<i>КНТЕУ 121 06-12.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.			19.10.20		РЗ	17	33
Керівник	Рассамакін В.Я.			19.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант	Криворучко О.В.			19.10.20				
Розробив	Костенко А.А.			19.10.20				
					Реалізація програмного продукту			

Для правильної роботи програмного продукту, потрібно мати підключений мікрофон. Після цього треба виконати установку всіх необхідних бібліотек Python:

1. SpeechRecognition – ця бібліотека представляє собою обгортку для багатьох різних сервісів і бібліотек розпізнавання мовлення.

Для її підключення спочатку треба ввести в командну у cmd «pip install speechrecognition» рис. 3.1.

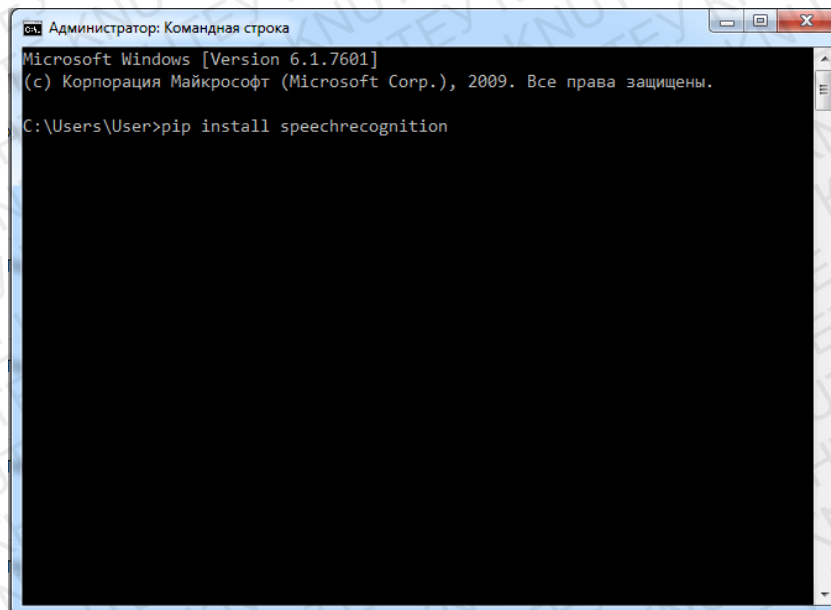


Рис. 3.1. Установка бібліотеки SpeechRecognition

2. PyAudio – це допоміжна бібліотека для ініціалізації та підключення мікрофона.

Для установки цієї бібліотеки треба ввести у cmd «pip install pyaudio»
Рис. 3.2.

					КНТЕУ 121 06-12.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

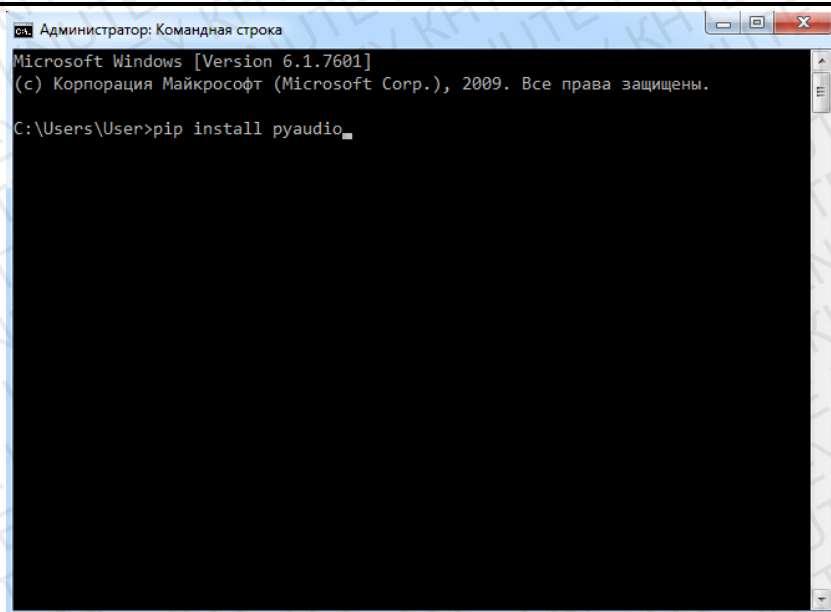


Рис. 3.2. Установка бібліотеки PyAudio

3. PySide2 – це прив’язка языка Python до інструментарію Qt, сумісна на рівні API с PyQt. На відміну від PyQt, PySide доступна для вільного використання як у відкритих, так і зачинених, в частності, комерційних проектах, оскільки ліцензована по LGPL.

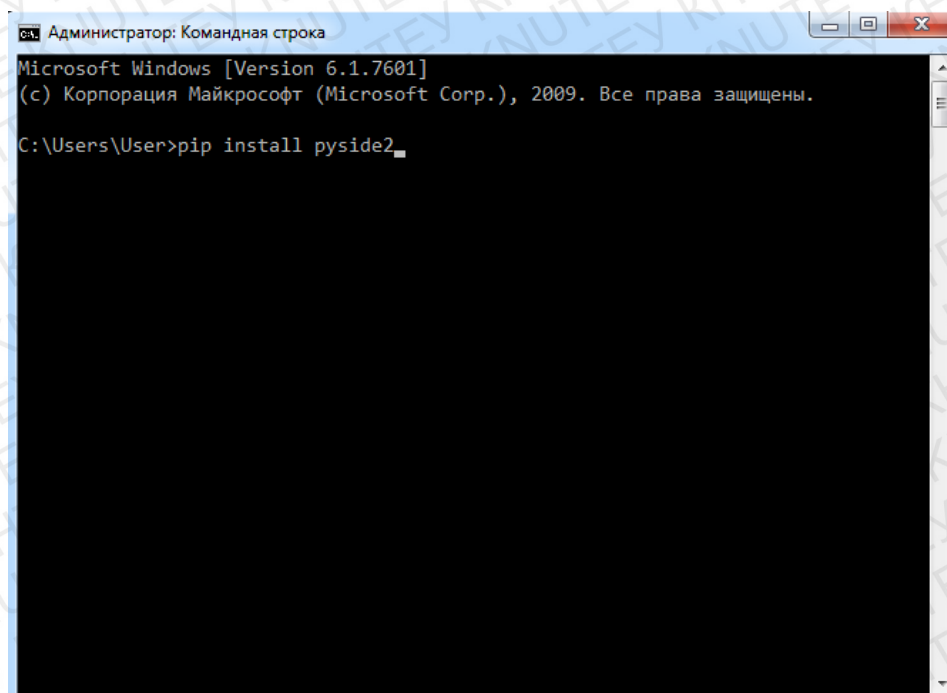


Рис. 3.3. Установка бібліотеки PySide2

					КНТЕУ 121 06-12.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

Тепер треба створити логіку програми, для цього треба в коді програми спочатку підключити всі бібліотеки.

Для підключення модулю розпізнавання мови вводим такі данні:

```
import sys  
import speech_recognition as sr
```

Також позначаємо наш модуль змінною «sr». Цих бібліотек буде достатньо для створення початкової логіки програми.

Створимо функцію «def record_volume():», ця функція відповідає за запис голосу, і не працює без встановленого модуля PyAudio. Прописуємо в функції нову перемінну «r» надаємо їй значення `r = sr.Recognizer()`, тобто підключаємо модуль розпізнавання з бібліотеки SpeechRecognition.

Додаємо перемінній значення:

```
with sr.Microphone() as source:  
    audio = r.listen(source)
```

Це підключає мікрофон користувача, і дає змогу програмі зчитувати його данні.

Далі створюємо нову перемінну «query» та присвоюємо їй значення:
`r.recognize_google(audio, language='uk-UA')`

Це підключає до цієї змінної модуль розпізнавання, та додає мову розпізнавання.

В залишку блоку логіки програми, ми повинні додати функціонал, для створення, та зберігання інформації, що виведе наша програма. Для цього введемо цей код:

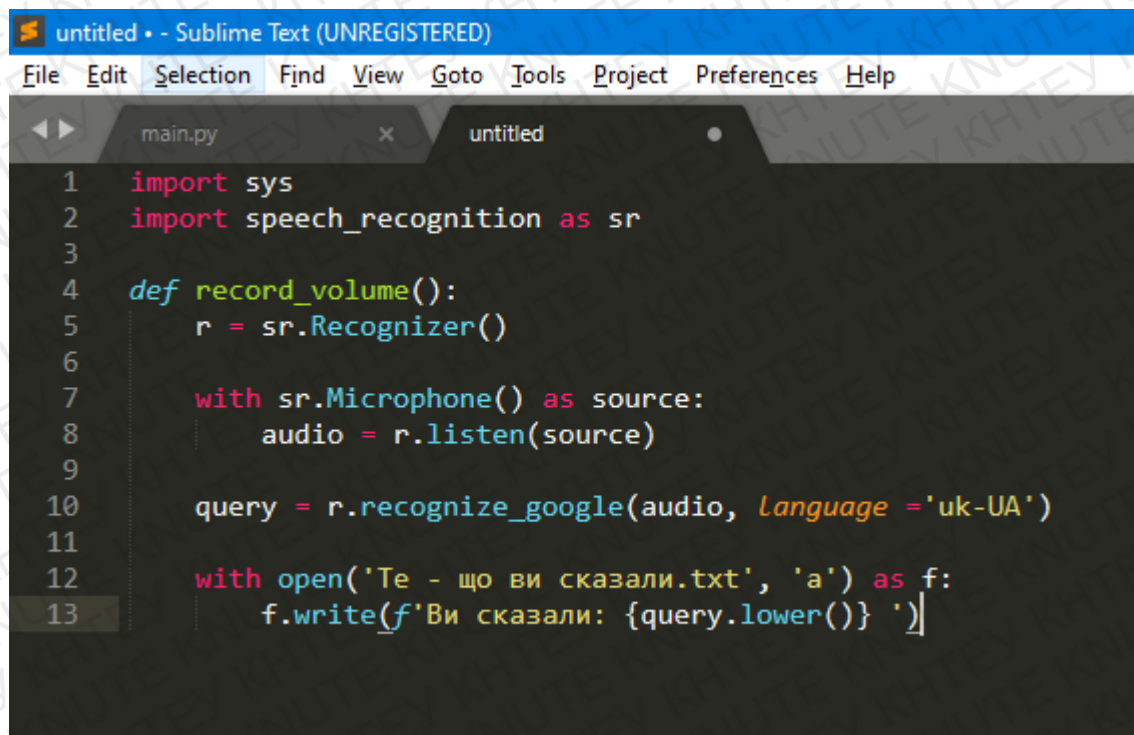
```
with open('Те - що ви сказали.txt', 'a') as f:  
    f.write(f'Ви сказали: {query.lower()} ')
```

Додається умова, що програма мусить зберегти данні, що виводить у файлі з ім'ям «Те - що ви сказали.txt», та якщо такого файла нема, він

					КНТЕУ 121 06-12.MP	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

створиться. Також додається надпис «Ви сказали:», перед кожним новим виводом програми, у створений файл.

На цьому логіка нашої програми закінчена.



```

1  import sys
2  import speech_recognition as sr
3
4  def record_volume():
5      r = sr.Recognizer()
6
7      with sr.Microphone() as source:
8          audio = r.listen(source)
9
10     query = r.recognize_google(audio, Language='uk-UA')
11
12     with open('Te - що ви сказали.txt', 'a') as f:
13         f.write(f'Ви сказали: {query.lower()} ')]

```

Рис. 3.4 Основна логіка програми.

3.3. Створення зовнішньої оболонки програми

Після того, як був створений, так званий, «двигун» програми, треба підключити функціонал, який допоможе оформити зовнішній вигляд програми. Для цього використаємо PyQt5, та влаштований в ньому інструмент під назвою Qt Designer.

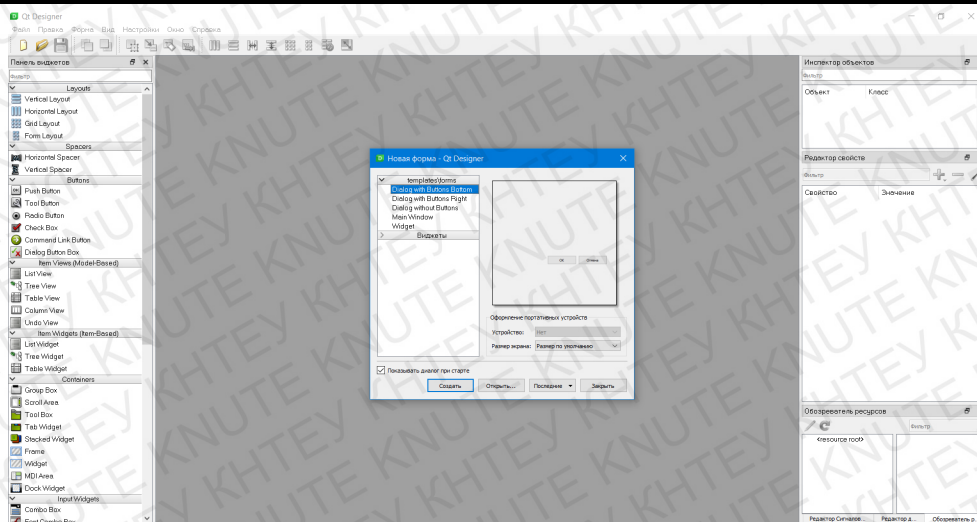


Рис. 3.5 Інструмент Qt Designer

Створимо звичайне пусте вікно, та додаємо на нього:

- QPushButton
- QLabel

Для роботи з цим інструментом, потрібно мати базові навички CSS, бо для редагування вікна програми використовують саме ці стилі.

Редагуємо QPushButton:

Змінюємо розмір, текст кнопки, його шрифт та зміну кольору кнопки під час наведення та натискання.

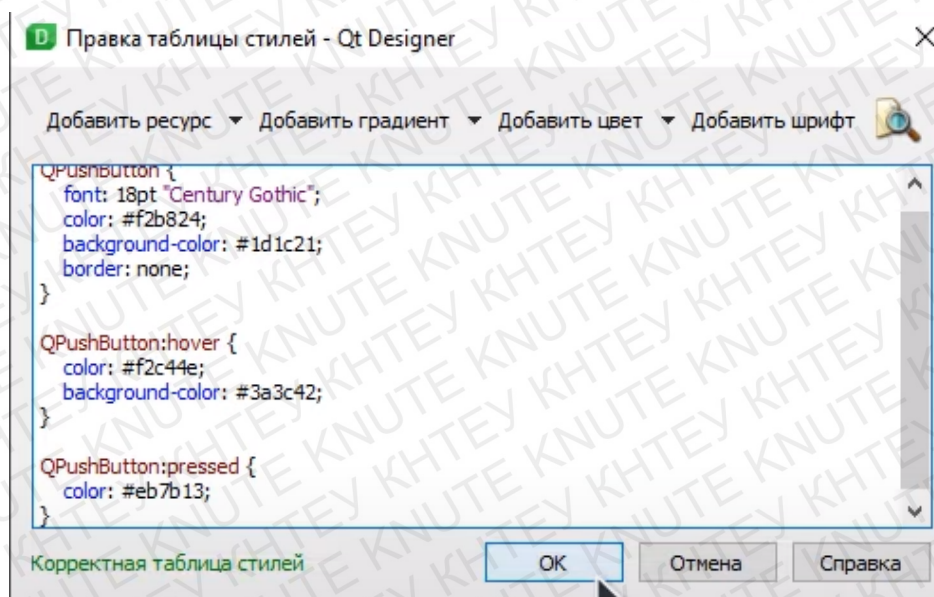


Рис. 3.6 Використання CSS для редагування кнопки програми

					Аркуш
					22
Зм.	Аркуш	№ докум	Підпис	Дата	

Також редагуємо BackGround, видаляємо текст з TextLabel та змінюємо його колір на інший.

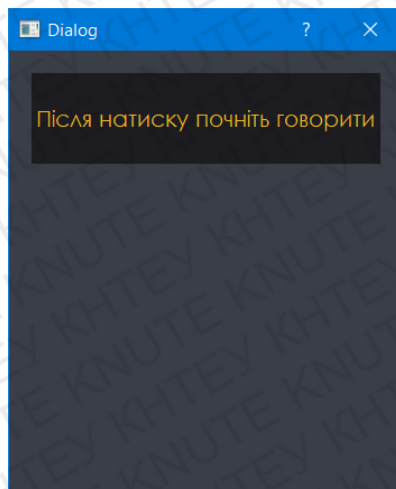


Рис. 3.7 Зовнішній вигляд програми після редагування.

3.4. Поєднання логіки програми та оболонки.

Після створення оболонки треба зберегти створений файл формату «.ui» під назвою «file». Потім треба конвектирувати цей файл з «xml-файла» в «.py-файл». Для цього потрібно відкрити cmd, та перейти туди, де зберігається сам файл, та ввести:

pyside2-uic “шлях до збереженого файла xml\file.ui” -x -o “шлях для збереження конвертованого .py-файлу\test.py ”

Створюємо новий файл main.py, та вирізаємо з файлу test.py цей код:

```
import sys
app = QtWidgets.QApplication(sys.argv)
Dialog = QtWidgets.QDialog()
ui = Ui_Dialog()
ui.setupUi(Dialog)
Dialog.show()
sys.exit(app.exec_())
```

					КНТЕУ 121 06-12.MP	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

Через те що ми вже маємо логіку своєї програми, потрібно поєднати її з об'єктом, тобто з самим вікном та його функціоналом.

Для цього імпортуємо в нашу програму декілька бібліотек:

```
from PyQt5 import QtCore, QtGui, QtWidgets
```

Ця команда витягує потрібні данні з PyQt5, щоб відобразити об'єкти вікна програми.

Також треба зробити витяг зі вже створеного файлу test.py, в якому вказані данні та оформлення вікна програми:

```
from test import Ui_Dialog
```

Тож наразі маємо програму з вже витягнутим функціоналом, але не підключеним до нашої логіки.

```
1  import sys
2  import speech_recognition as sr
3  from PyQt5 import QtCore, QtGui, QtWidgets
4  from test import Ui_Dialog
5
6  # Create app
7
8  app = QtWidgets.QApplication(sys.argv)
9
10 # init
11
12 Dialog = QtWidgets.QDialog()
13 ui = Ui_Dialog()
14 ui.setupUi(Dialog)
15 Dialog.show()
16
17 # Hook logic
18
19 def record_volume():
20     r = sr.Recognizer()
21
22     with sr.Microphone() as source:
23         audio = r.listen(source)
24
25         query = r.recognize_google(audio, language='uk-UA')
26
27 # Main loop
28
29 sys.exit(app.exec_())
```

Рис. 3.8 Блок програми main.py

					КНТЕУ 121 06-12.MP	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

Наша задача зробити так, щоб при натисканні кнопки почався процес запису та розпізнавання мовлення. Для цього використовуємо:

```
ui.pushButton.clicked.connect ( record_volume )
```

Тобто при натисканні кнопки, починається звертання до функції «record_volume», яка в свою чергу починає виконувати обробку запиту.

Наразі створена функція запустилася, але інформацію, яку вона повинна видати ще нікуди не виведена. Для її виведення використаємо таку функцію:

```
ui.label.setText (f'Ви сказали: {query.lower()}')
```

Після додавання цієї функції виведена інформація має такий вигляд рис.

3.9

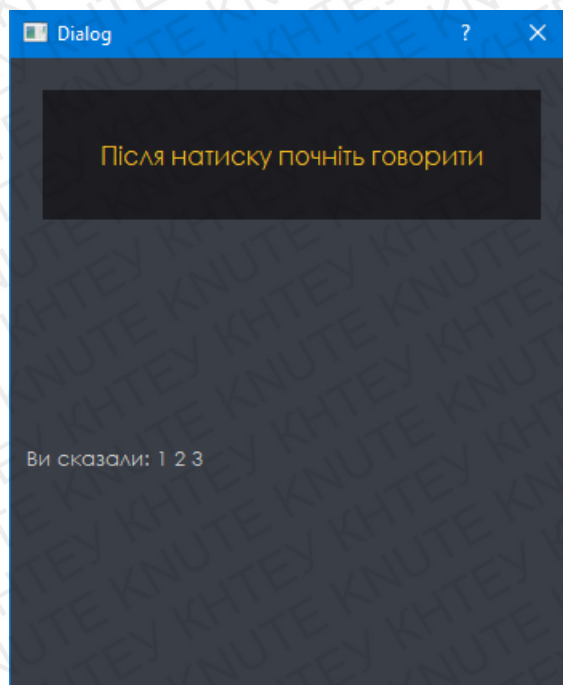


Рис. 3.9 Зображення виведеної інформації після обробки мовлення у вікні програми.

Для збереження даних виводу знадобиться така функція:

```
with open('Те - що ви сказали.txt', 'a') as f:
```

```
f.write(f'Ви сказали: {query.lower()} ')
```

					КНТЕУ 121 06-12.МР	Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

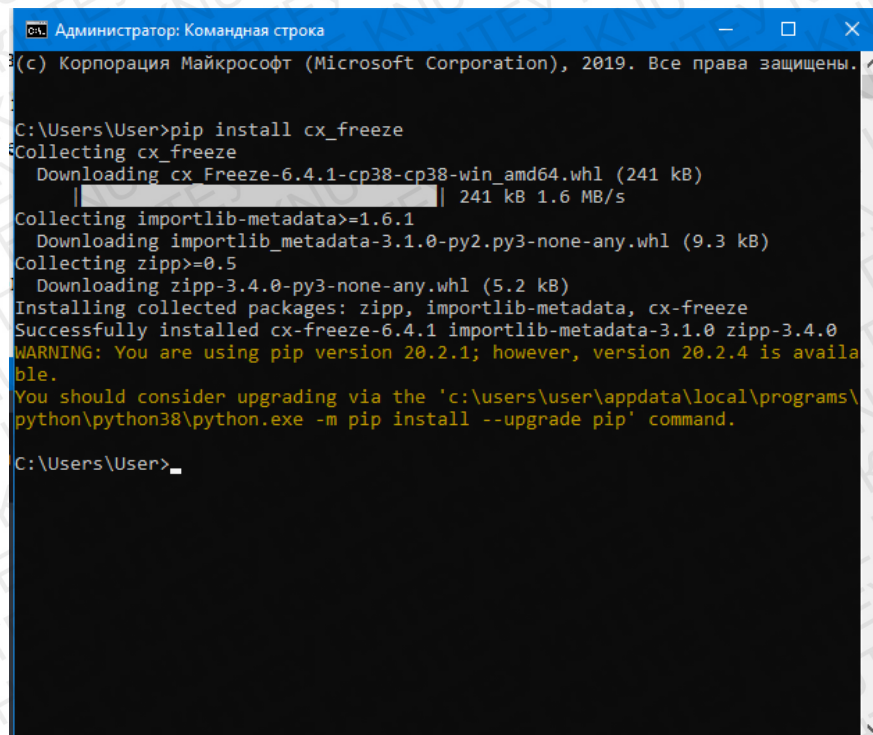
Вона означає, що програма створить файл формату «.ТХТ», під назвою «Те – що ви сказали.» та до дасть у нього данні, які були розпізнанні самою програмою.

3.5. Створення інсталятора програми

Для того, щоб програма працювала на різних комп'ютерах, треба створити її інсталятор. Для цього використаємо компілятор cx_freeze.

Вводимо у cmd наступну команду:

`pip install cx_freeze`



```
Администратор: Командная строка
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\User>pip install cx_freeze
Collecting cx_freeze
  Downloading cx_Freeze-6.4.1-cp38-cp38-win_amd64.whl (241 kB)
    | 241 kB 1.6 MB/s
Collecting importlib-metadata>=1.6.1
  Downloading importlib_metadata-3.1.0-py2.py3-none-any.whl (9.3 kB)
Collecting zipp>=0.5
  Downloading zipp-3.4.0-py3-none-any.whl (5.2 kB)
Installing collected packages: zipp, importlib-metadata, cx-freeze
Successfully installed cx-freeze-6.4.1 importlib-metadata-3.1.0 zipp-3.4.0
WARNING: You are using pip version 20.2.1; however, version 20.2.4 is available.
You should consider upgrading via the 'c:\users\user\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.
C:\Users\User>
```

Рис. 3.10 Установка компілятора.

Тепер потрібно створити файл для компіляції. Для цього в папці з проектом потрібно створити ще один файл з ім'ям «setup.py».

У цьому файлі треба розмістити наступне:

`from cx_Freeze import setup, Executable`

Цією строчкою програма бере готовий модуль «setup», «Executable» з компілятора `sx_Freeze`.

Наступним кроком буде створення ім'я програми, опис програми та основний виконавчий файл.

```
setup(  
    name = 'Розпізнавання_мовлення',  
    version = '1.0',  
    description = 'Speech_pyqt',  
    options = options,  
    executables = [Executable('main.py')]  
)
```

Рис. 3.11. Параметри для компіляції програми

Так, як в проєкті використовується декілька файлів, а саме «test.py» та «main.py» їх потрібно об'єднати в процесі компіляції. Для цього використаємо функцію «options», де прив'яжемо до головного файла «main» файл «test».

```
options = {  
    'build.exe': {  
        'includes': [  
            'test'  
        ]  
    }  
}
```

Рис. 3.12. Функція об'єднання файлів за допомогою `sx_Freeze`

Для створення інсталятора проєкту потрібно використати відповідні команди в `cmd`:

1. За допомогою команди `cd` потрібно перейти до папки поточного проєкту.
2. Вказати шлях до каталогу з інтерпретатором Python.
3. Написати виконуючий файл `setup.py` з командою для компіляції інсталятора `bdist_msi`.

					КНТЕУ 121 06-12.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

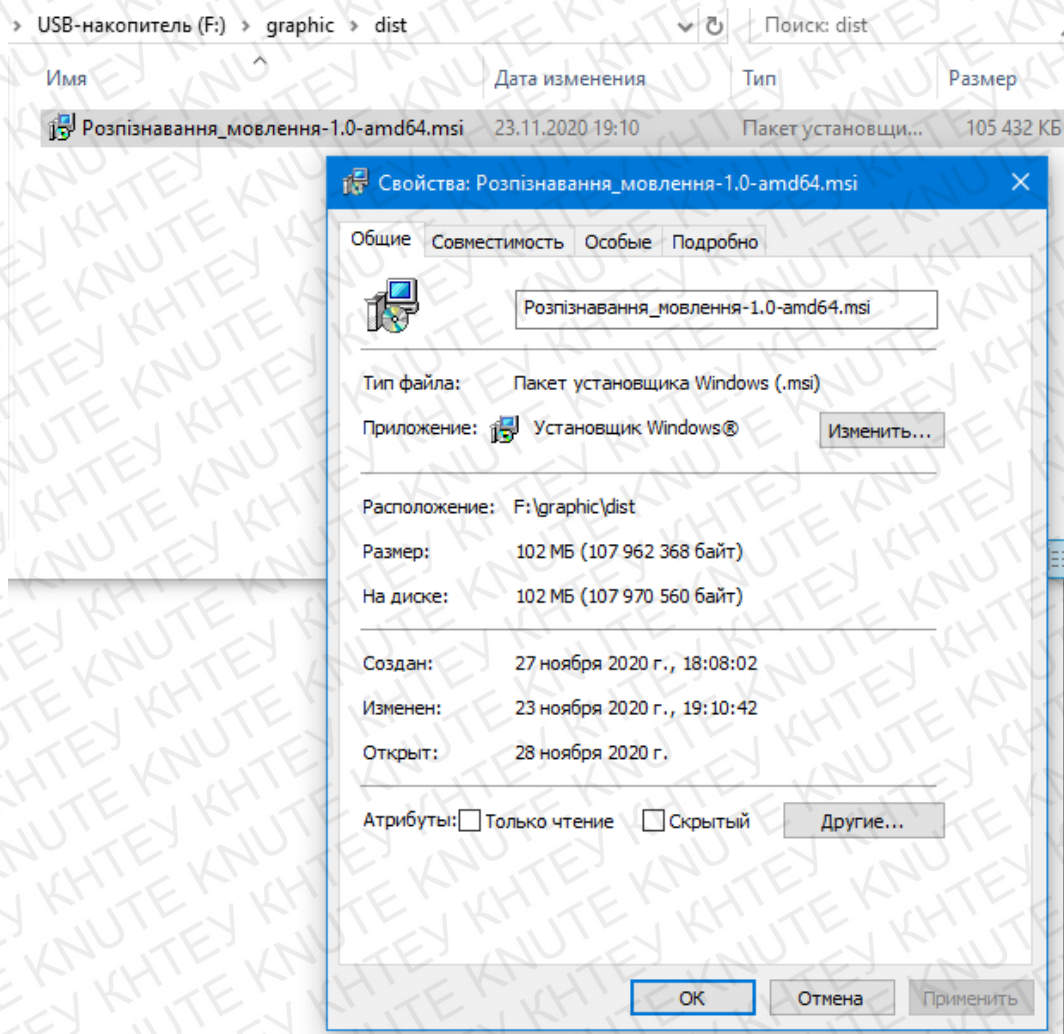


Рис. 3.13 Створений інсталятор.

3.6. Висновок до розділу 3

Отже, інтерпретатор Python, його програмні модулі та бібліотеки, дозволили створити єдиний програмний продукт для розпізнавання мовлення людини та зберігання тексту на ПК, та дозволяє вільно використовувати програму на будь-якому ПК за допомогою інсталятора.

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. Проведено аналіз існуючих аналогів та визначено клієнтську архітектуру застосунку, яка є оптимальною для реалізації поставлених завдань.
2. Визначена ОС Windows як найбільш розповсюджена та уживана на сучасному ринку операційних систем. Запропонована спрощена модель розпізнавання мовленої інформації, з використанням Windows 10 як базової комп'ютерної ОС з урахуванням її вдосконаленої системи швидкісної роботи та надійного функціонування цього операційного середовища.
3. Проведений аналіз засобів розробки показав, що найзручнішим для реалізації поставлених завдань є інтерпретатор Python та PyQt5.
4. В процесі виконання магістерської роботи розроблено програму розпізнавання мовлення людини та архівування мовленої інформації.
5. Розроблена система повністю задовольняє поставлені функціональні вимоги. Тестування продемонструвало повну відповідність функціональним вимогам та надійну роботу застосунку.
6. Робота пройшла апробацію на конференції з публікацією тексту доповіді.
7. За результатами проведених досліджень здійсненна публікація в збірнику наукових статей студентів, в збірнику магістерських наукових статей студентів «Software engeneric» комп'ютерних систем.

					<i>КНТЕУ 121 06-12.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			30.10.20	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	Стадія	Аркуш	Аркушів
Керівник	Рассамакін В.Я.			30.10.20		ВП	29	33
Гарант	Криворучко О.В.			30.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив	Костенко А.А.			30.10.20				
					<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Learning Python, Fifth Edition by Mark Lutz. Published by O'Reilly Media. ISBN: 978-1-449-35573-9
2. Programming Python, 4th Edition by Mark Lutz. Published by O'Reilly Media. ISBN: 978-0-596-15810-1
3. Python Programming for the Absolute Beginner, 3rd Edition by Michael Dawson. Publisher by Cengage Learning PTR. ISBN: 978-1-435-455-009
4. Python 3.6.4 documentation [Електронний ресурс] / Режим доступу до ресурсу: <https://docs.python.org/3/>
5. Бизли, Дэвид М. Язык программирования Python. Справочник. — К.: ДияСофт, 2000.
6. Дэвид М. Бизли. Python. Докладна справка, 4-е видання.
7. Марк Саммерфилд. Программування на Python 3. Детальне керівництво.
8. Ноа Гифт, Джереми М. Джонс. Python в системному адмініструванні.
9. Вінцюк Т.К. Аналіз, розпізнавання і інтерпретація звукових сигналів / Т. К. Вінцюк. - Київ: Наукова думка, 1987.
10. Рабинер Р. Теорія і застосування цифрової обробки сигналів / Р. Рабинер. - М.: Мир, 1978.
11. Оппенгейм А.В. Цифрова обробка сигналів / А.В. Оппенгейм. - М.: Связь, 1979.
12. Lindblom B. Durational patterns of Swedish Phonology: do they reflect short - term motor memory process / B. Lindblom, B. Lyberg, K. Holmgren. - Bloomington, 1981.

					<i>КНТЕУ 121 06-12.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.			30.10.20		ВП	30	33
Керівник	Рассамакін В.Я.			30.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант	Криворучко О.В.			30.10.20				
Розробив	Костенко А.А.			30.10.20				
					<i>Висновки та пропозиції</i>			

13. Rabiner L. Fundamentals of Speech Recognition / L. Rabiner, B.H. Juang. - Prentice Hall PTR, 1993.
14. Привалов І. І. Введення в теорію функцій комплексного змінного / І. І. Привалов. - М.: Наука, 1967.
15. L. E. Baum, T. Petrie, G. Soules, N. Weiss A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. - Ann. Math. Stat., 1970.
16. Davies K.H., Biddulph R., Balashek S. Automatic Speech Recognition of Spoken Digits. - J. Acoust. Soc. Am, 1952.
17. У. Лі. Методи автоматичного розпізнавання мови. - М.: Мир, 1983.

					<i>КНТЕУ 121 06-12.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		31

ТЕХНІЧНЕ ЗАВДАННЯ

1. Загальні відомості

Назва проекту: «Розпізнавання мовлення».

Планові строки розробки: серпень 2019 – вересень 2019.

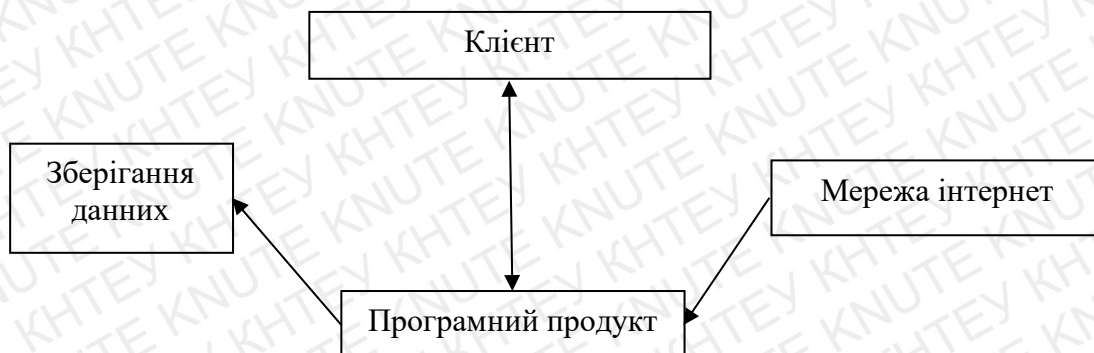
Потенційні користувачі: користувачі ПК.

Призначення програмного рішення: розробка програмної оболонки в діапазоні від розпізнавання мовлення до збереження розпізнаної мовленої інформації.

Мета створення програмного рішення: застосування можливостей програмування на Python 3.

2. Структура проекту

Структура проекту а саме самої системи вказана в пункті «Структура системи» . Вона має такий вигляд:



					<i>КНТЕУ 121 06-12.МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка програмної оболонки системи акустичного розпізнавання та обробки аудіо-інформації	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	
Зав. каф.	Криворучко О.В.			28.02.20		<i>ТЗ</i>	<i>32</i>	<i>33</i>	
Керівник	Рассамакін В.Я.			28.02.20		<i>Технічне завдання</i>	<i>Факультет інформаційних технологій 2м курс, 6 група</i>		
Гарант	Криворучко О.В.			28.02.20					
Розробив	Костенко А.А.			28.02.20					

3. Функціональні вимоги до додатку

Перелік функціональних можливостей визначений базовими діями, що виконує програмний продукт:

- Розпізнавання мовлення;
- Обробка інформації;
- Виведення результатів розпізнавання;
- Збереження розпізнаних даних.

4. Вимоги до програмного забезпечення

Розробка серверної та клієнтської частини програмного рішення здійснюватиметься із використанням мови програмування Python, доступом до бібліотек PyQt5 та розробки в середовищі Sublime Text 3.

5. Вимоги до технічного забезпечення

Найважливішою вимогою до задіяних пристроїв є наявність комп'ютера з операційною системою Windows 10.

Мінімальна конфігурація ПК для роботи програмного додатку:

- Microsoft Windows 10/8.
- RAM: 1 Гб і більше.
- Processor: Intel Celeron та краще.
- Наявність доступу до мережі інтернет, наявність монітору, мікрофона та мишки.

					КНТЕУ 121 06-12.МР	Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата		

ДОДАТКИ

Додаток А

Основний файл main.py

```
# -*- coding: utf-8 -*-

import sys
import speech_recognition as sr
from PyQt5 import QtCore, QtGui, QtWidgets
from test import Ui_Dialog

# Create app

app = QtWidgets.QApplication(sys.argv)

# init

Dialog = QtWidgets.QDialog()
ui = Ui_Dialog()
ui.setupUi(Dialog)
Dialog.show()

# Hook logic

def record_volume():
    r = sr.Recognizer()

    with sr.Microphone() as source:
        audio = r.listen(source)

        query = r.recognize_google(audio, language='uk-UA')
        ui.label.setText(f'Ви сказали: {query.lower()}')

        with open('Те - що ви сказали.txt', 'a') as f:
            f.write(f'Ви сказали: {query.lower()} ')

    ui.pushButton.clicked.connect ( record_volume )

# Main loop

sys.exit(app.exec_())
```

Допоміжний файл оформлення оболонки test.py

```

# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'file.ui'
#
# Created by: PyQt5 UI code generator 5.15.1
#
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(352, 399)
        Dialog.setStyleSheet("background-color: #3a3e47; ")
        self.pushButton = QtWidgets.QPushButton(Dialog)
        self.pushButton.setGeometry(QtCore.QRect(20, 20, 311, 81))
        self.pushButton.setStyleSheet("QPushButton {\n"
"font: 12pt \"Century Gothic\";\n"
"    color: #f2b824;\n"
"    background-color:#1d1c21;\n"
"    border:none;\n"
"}\n"
"QPushButton:hover {\n"
"    color:#f2c44e;\n"
"    background-color:#3a3c42;\n"
"}\n"
"QPushButton:pressed {\n"
"color: #eb7b13;\n"
"}")
        self.pushButton.setObjectName("pushButton")
        self.label = QtWidgets.QLabel(Dialog)
        self.label.setGeometry(QtCore.QRect(10, 110, 331, 281))
        self.label.setStyleSheet("color: #ccc;\n"
"font: 10pt \"Century Gothic\";\n"
"")
        self.label.setText("")
        self.label.setObjectName("label")

```

```
self.retranslateUi(Dialog)
QtCore.QMetaObject.connectSlotsByName(Dialog)

def retranslateUi(self, Dialog):
    _translate = QtCore.QCoreApplication.translate
    Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
    self.pushButton.setText(_translate("Dialog", "Після натиску почніть говорити"))
```

Допоміжний файл для створення інсталлера програми setup.py

```
# -*- coding: utf-8 -*-

from cx_Freeze import setup, Executable

options = {
    'build.exe': {
        'includes': [
            'test'
        ]
    }
}

setup(
    name = 'Розпізнавання_мовлення',
    version = '1.0',
    description = 'Speech_pyqt',
    options = options,
    executables = [Executable('main.py')]
)
```