

Міністерство освіти і науки України
Дрогобицький державний педагогічний університет
імені Івана Франка
Кафедра фізики та інформаційних систем

«До захисту допускаю»
завідувач кафедри фізики
та інформаційних систем,
кандидат фіз.-мат. наук, доцент
_____Віталій ГОЛЬСЬКИЙ
« ____ » _____ 2024 р.

**Розробка системи голосового
керування персональним комп'ютером**

Спеціальність: 122 Комп'ютерні науки

Магістерська робота

на здобуття кваліфікації – магістр з комп'ютерних наук

Автор роботи: Котів Владислав Романович

підпис

Науковий керівник: доцент кафедри
фізики та інформаційних систем,
кандидат технічних наук
Григорович Андрій Геннадійович

підпис

Дрогобич – 2024

ЗАХИСТ МАГІСТЕРСЬКОЇ РОБОТИ

Розробка системи голосового керування персональним комп'ютером

Оцінка за стобальною шкалою: _____

Оцінка за національною чотирибальною шкалою: _____

Коротка мотивація захисту:

[illegible]

Дата

Голова ЕК _____
підпис

прізвище, ім'я

Секретар ЕК _____
ніднус

прізвище, ім'я

АНОТАЦІЯ

У магістерській роботі досліджено розробку системи голосового керування персональним комп'ютером. Основною метою є створення інструменту, який забезпечує зручний та інклюзивний доступ до функцій комп'ютера, зокрема для користувачів із обмеженими фізичними можливостями, а також оптимізує взаємодію з системою для широкого кола користувачів. Розробка системи базується на використанні сучасних технологій, таких як Python, бібліотека SpeechRecognition для розпізнавання голосу та Pyttsx3 для синтезу мовлення. Інтеграція системи з ключовими функціями операційної системи забезпечує виконання голосових команд, таких як відкриття програм, пошук інформації та виконання рутинних задач. Завдяки цим технологіям система дозволяє зменшити залежність від фізичних пристроїв введення, покращує доступність функціоналу комп'ютера та забезпечує інноваційний підхід до взаємодії з персональним комп'ютером.

ABSTRACT

The master's thesis explores the development of a voice control system for a personal computer. The primary goal is to create a tool that provides convenient and inclusive access to computer functions, particularly for users with physical disabilities, while optimizing system interaction for a wide range of users. The system's development is based on modern technologies such as Python, the SpeechRecognition library for speech recognition, and Pyttsx3 for speech synthesis. The integration of the system with key operating system functionalities enables the execution of voice commands, such as opening applications, searching for information, and performing routine tasks. By leveraging these technologies, the system reduces dependence on physical input devices, enhances accessibility to computer functionality, and offers an innovative approach to interacting with personal computers.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ГОЛОСОВОГО КЕРУВАННЯ ПЕРСОНАЛЬНИМИ КОМП'ЮТЕРАМИ	8
1.1. Важливість і перспективи голосового керування комп'ютерами	8
1.2. Аналіз існуючих рішень і їхніх недоліків	9
1.3. Обґрунтування потреби у власній розробці	11
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ СИСТЕМИ	13
2.1. Функціональні можливості системи	13
2.2. Цільова аудиторія	13
2.3. Принципи роботи	15
2.4. Архітектура системи	6
2.5. Інтерфейс та дизайн	17
2.6. Зберігання даних	18
2.7. Очікувані результати	18
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	20
3.1. Діаграма прецедентів	20
3.2. Діаграма діяльності	21
3.3. Використання технології та інструментів	23
3.4. Архітектура програми	25
3.5. Розгортання проєкту	28
3.6. Переваги і можливості для розвитку	28
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	30
ДОДАТКИ	31

ВСТУП

Актуальність. Сьогодні у світі інформаційні технології стрімко розвиваються, і впровадження нових винаходів у наше повсякденне життя є важливою частиною процесу автоматизації та покращення взаємодії з користувачем. Однією з таких технологій є голосове керування. Siri і Google Assistant вже змінюють правила гри взаємодії користувача з технікою. Користувачі можуть виконувати безліч завдань, не торкаючись клавіатури чи миші.

Однак на ринку все ще є незаймана можливість розробки інтерфейсів з голосовим керуванням для персональних комп'ютерів, а точніше для користувачів, які через фізичні обмеження мають дуже обмежену взаємодію з традиційними пристроями.

Як би це не було важливо, але створення системи голосового керування для персонального комп'ютера є значним кроком у напрямках забезпечення доступності та зручності використання комп'ютерних технологій різними категоріями користувачів, і, зокрема, категорією людей з обмеженими можливостями. оскільки всі користувачі зацікавлені в оптимізації своєї взаємодії з комп'ютером за допомогою людської мови.

Мета дослідження: Метою роботи є розробка голосової системи на базі персонального комп'ютера, що дозволяє голосово керувати основними операціями ПК. Система повинна мати простий для розуміння інтерфейс користувача, через який можна буде відкривати програми, виконувати основні функції (наприклад, перевіряти час, погодні умови, запускати програми), а також взаємодіяти з іншими службами через Інтернет.

Для реалізації поставленої мети поставлені наступні завдання:

1. Складення алгоритму розпізнавання голосу для розуміння команд

користувача.

2. Інтеграція голосового помічника для обробки запитів і виконання операцій (наприклад, час, погода, виконання програми).

3. Створення інтерфейсу користувача для візуалізації голосових команд і результату дій.

Об'єкт дослідження: процеси голосового керування персональним комп'ютером, що включає програмні засоби для обробки голосових команд, розпізнавання мови, синтезу мови та інтеграції з іншими сервісами комп'ютера для виконання операцій.

Предмет дослідження: методи і засоби розпізнавання голосових команд та їх інтеграція з операційною системою для автоматизації виконання завдань, таких як запуск програм, управління вікнами, отримання інформації (погода, час тощо) за допомогою голосу.

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ГОЛОСОВОГО КЕРУВАННЯ ПЕРСОНАЛЬНИМИ КОМП'ЮТЕРАМИ

1.1. Важливість і перспективи голосового керування комп'ютерами

Голосове керування ПК відноситься до одного з перспективних напрямів розвитку інформаційних технологій, оскільки спрощує все роботу з технікою. Адже, в умовах, коли технології окремою частиною нашого повсякденного життя постає питання не лише ефективності, але й доступності всім сучасним технологіям до всіх користувачів. Це також зручно в тому плані, що голос сказує руки та можна керувати ПК в режимі наполовину навіть в одному відділенні.

Поява мовних технологій відбулася в 20 столітті; однак вони стали доступними для широкого кола користувачів лише в останнє десятиліття. Причини цього — удосконалення алгоритмів розпізнавання мовлення, зростання потужності обчислювальної техніки та активне використання нейронних мереж. Зараз голосові помічники, такі як Siri, Google Assistant або Amazon Alexa, є стандартним аксесуаром для багатьох мобільних пристроїв, розумних колонок і багатьох інших гаджетів. Сфера їх застосування в кадрових комп'ютерах невелика.

Голосове керування персональними комп'ютерами викликало інтерес з кількох причин. По-перше, це розширює можливості комп'ютера, оскільки дозволяє виконувати численні рутинні завдання набагато швидше та ефективніше. Наприклад, користувач може запустити програму на свій вибір без використання клавіатури та миші, знайти файл або навіть переглянути прогноз погоди. По-друге, голосове керування має величезний потенціал, щоб зробити обчислення доступними для тих, хто має обмежені фізичні можливості та не може використовувати стандартні пристрої введення.

Ще один важливий аспект — багатозадачність. Голосове керування дозволяє працювати над декількома справами одночасно, оскільки ваші руки залишаються вільними. Наприклад, текст для документа можна продиктувати під час приготування кави або надсилання електронного листа під час іншої роботи.

Зберігаючи всі ці переваги, поточні системи голосового керування мають певні перешкоди, які обмежують їх широке впровадження. Найвідомішою проблемою є умова, яка вимагає підключення до Інтернету. Голосові помічники потребують підключення до хмарних серверів для обробки голосових команд, тому вони стають неактуальними за відсутності стабільного Інтернету. Ще одна серйозна проблема – низька функціональність. Ці системи голосового керування розраховані на невелику кількість функцій і не забезпечують повної взаємодії з персональним комп'ютером.

З огляду на все це, голосове управління персональними комп'ютерами має велике майбутнє. Розвиток цих технологій несе з собою можливість для подальшого застосування, зокрема, у пов'язаній з роботою діяльності, процесах у навчальних закладах та домашньому використанні.

1.2. Аналіз існуючих рішень і їхніх недоліків

На сьогоднішній день, на ринку можна знайти безліч голосових помічників на будь-який смак і не тільки. Одним із найпопулярніших є Siri – голосовий помічник Apple. Siri охоплює всіх користувачів iPhone, iPad або Mac так як мова помічника має гнучку інтеграцію з Apple. Для використання Siri необхідно виконати декілька простих завдань. Наприклад: встановлення нагадувань, запуск додатків або отримання необхідної інформації. Але, включити цю функцію можна тільки на апаратах Apple, а тому користувачі інших платформ дуже

страждають. Багато хто з нас може так і не зрозуміти, що таке Siri, адже це не є повноцінною системою, і вони намагалися керувати своїм комп'ютером зі смартфона.

Чимало українців цікавиться активним підвищенням ефективності системи Cloud-M, що переважно стосується Google Assistant. Проте один з представників Android операційних систем продовжує залишатися активним серед учнів та користувачів в цілому. Google Assistant є таким, що домінує на пристроях з операційною системою android. Які ж можливості має google асистент? Його ім'я говорить само за себе – за допомогою голосу він здатний управляти будь-якими додатками. Працює Google Assistant на основі штучного інтелекту, так що його можливості значно розширені. Проте Google Assistant переважно призначений для використання на мобільних пристроях, з базою для комп'ютера. Але так само як і мова йшла раніше, домінуюче місце займає підключення до інтернету.

Схожі можливості має також Amazon Alexa, найвідоміший гучномовець та асистент, практично без аналогів на ринку. Засоби Alexa можна з легкістю інтегрувати в роботи вашого розумного дому: встановили лампу, термостат, камеру, і працюйте з розумним домом через Alexa. Тепер залишається питання - для чого ж все це? Так, Alexa має вузьку спеціалізацію, адже високі технології користувачів комп'ютерних систем не підходять.

Dragon NaturallySpeaking охарактеризовує одну з слідуючих частин програмного забезпечення. Вона добре відома на ринку, адже здатна безумовно перетворювати текст на його власну озвучку, а навколишні голоси не заважають функціоналу. Це має велику популярність серед таких сегментів на ринку, як поняття медицина та юриспруденція. Проте варто зазначити, що ця програма є специфічною для її налаштувань, але так само як і у випадку з комбінуванням ноутбука.

Після проведення аналізу таких ринкових рішень слід зазначити, що так і не вдалося знайти універсальну програму.

1.3. Обґрунтування потреби у власній розробці

Аналіз існуючих комп'ютерних систем із голосовим управлінням свідчить про те, що наразі на ринку немає універсального продукту, який би задовольняв усі вимоги сучасного користувача. Існуючі системи часто мають обмеження в своїх можливостях, технічних характеристиках або цільовій платформі, що створює перешкоди для їх широкого впровадження. Це породжує термінову потребу в розробці нової спеціалізованої системи, яка подолатиме ці обмеження і забезпечить нову якість взаємодії користувача з особистими комп'ютерами.

Однією з основних проблем, з якими стикаються в існуючих рішеннях, є дизайн інтерфейсу користувача, який спрощує використання рішення ширшим колом користувачів. Більшість популярних помічників, таких як Siri, Google Assistant або Amazon Alexa, призначені для виконання базових завдань, таких як пошук інформації або контроль програм. Однак для більш складних випадків використання, де очікується, що кінцевий користувач буде виконувати продуктивну роботу на ПК, ці системи недостатньо гнучкі. Ці системи часто вимагають додаткової конфігурації, встановлення певних застосунків або навіть певного рівня комп'ютерної грамоти від користувача.

Питання конфіденційності і захисту даних вимагають окремої уваги. Обробка голосових команд на хостингах може бути безпосередньою загрозою порушення або неоправданого зловживання особистою інформацією користувача. Це є особливо актуальним для сфери бізнесу або державного управління, де інформація має конфіденційний характер. Отже, нові рішення також повинні забезпечити високу безпеку при обробці даних з найменшою залежністю від

зовнішніх серверів.

З іншого боку, розробка системи голосового управління з можливістю інтерфейсу також дозволить врахувати ці всі аспекти і запропонувати більш універсальний і гнучкий інструментарій. Це, поряд з іншими перевагами, дозволить адаптувати систему Voice.exe під конкретні потреби користувачів Центру і його фахівців. Система повинна мати простий і зрозумілий інтерфейс, щоб навіть особам, які не мають жодного досвіду роботи з технікою, не було важко впровадити її в практику.

Забезпечення високої продуктивності при низьких вимогах до обладнання є важливим завданням. Впровадження техніки у вітчизняні комп'ютери нової ери цим буде перешкоджати. Середня вартість комп'ютерного обладнання є вигідною, з низькою вартістю графіки, що охоплює всіх любителів музики.

У новому продукті слід враховувати питання масштабування, дозволяючи інтегрувати додаткові функції та оновлення без необхідності внесення значних змін до основної архітектури. Це дозволить оперативно реагувати на мінливі потреби та сприятиме його тривалому використанню.

Створення власної системи голосового управління персональним комп'ютером - це актуальне завдання, яке під силу сучасним рішенням. Він має бути конкурентоспроможним на ринку та бути корисним для широкого кола користувачів.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ СИСТЕМИ

2.1. Функціональні можливості системи

Багатоцільовий програмний продукт, відомий як система голосового керування персональними комп'ютерами, був розроблений для забезпечення ефективної та простої взаємодії користувача з комп'ютером. Забезпечення комфорту користувача, збільшення можливостей комп'ютерної системи та максимально спрощене повсякденне виконання завдань – ось основні цілі.

Система дозволяє виконувати наступні важливі завдання.

- Керування програмами та файлами: використовуйте голосові команди для відкриття, редагування, копіювання, переміщення та видалення файлів.
- Пошук в Інтернеті: використання пошукових систем для здійснення голосового інтерфейсу запитів інформації.

Отримуйте довідкову інформацію, отримуючи доступ до поточної інформації про погоду, час, дату та новини.

- Керування мультимедіа: призупиняйте, відновлюйте та змінюйте відтворення музики чи відео.
- Автоматизація завдань: використовуйте сценарії для виконання повторюваних завдань, створення списків справ або встановлення нагадувань.

2.2. Цільова аудиторія

Система голосового керування комп'ютером зорієнтована на декілька категорій людей, щоб допомогти їм керувати та краще організовувати роботою. Серед цих категорій перебувають:

1. **Звичайні користувачі комп'ютерів**, котра хочуть оптимізувати свою звичайну роботу. Система дає можливість набагато прискорити виконання

простих завдань, таких як пошук інформації в інтернеті, запуск програм або виконання простих операцій голосовими командами. Це дає змогу користувачам економити час і зосереджуватись на більш важливих аспектах своєї роботи чи життя.

2. **Люди з обмеженими фізичними можливостями.** Для людей, які мають труднощі з фізичним введенням даних (наприклад, через обмежену рухливість рук або інші фізичні обмеження), голосове керування є важливим засобом доступу до технологій. Така система може стати незамінним інструментом для забезпечення більшої незалежності та зручності в повсякденному житті.
3. **Бізнес-користувачі,** які прагнуть автоматизувати рутинні процеси. У бізнес-середовищі автоматизація допомагає знижувати витрати часу на виконання стандартних завдань, таких як планування зустрічей, надсилання звітів або пошук документів. Голосове керування дозволяє бізнес-користувачам зосередитись на стратегічних завданнях, залишаючи операційні задачі на плечах інтелектуальних систем.
4. **Студенти та освітяни,** які потребують швидкого доступу до ресурсів і створення нотаток без необхідності фізичного введення тексту. Для студентів, які часто стикаються з необхідністю швидко занотовувати інформацію під час лекцій або обговорень, система голосових команд може стати чудовим інструментом для запису думок або пошуку відповідної інформації в Інтернеті.

Його простота у використанні та здатність інтегруватися з різними платформами робить його широким застосуванням у сфері охорони здоров'я, освіти та офісів.

У таких ситуаціях важливо мінімізувати час взаємодії з комп'ютером, щоб

система працювала максимально ефективно та продуктивно.

2.3. Принципи роботи

Основні принципи системи забезпечують ефективну обробку голосових команд, а також інтерактивність і точність виконання запитів.

Система працює на базі хмарних технологій, що дозволяє використовувати потужний штучний інтелект і сервіси обробки голосу.

Нижче наведено основні кроки, з яких складається процес взаємодії користувача з системою:

1. Розпізнавання голосу.

Першим кроком є аналіз звукового сигналу, що надходить від користувача.

Для цього система використовує спеціалізовані сервіси обробки мови, такі як OpenAI або Google Speech-to-Text, для перетворення голосових команд у текст.

Це дозволяє системі зрозуміти, яку саме команду або запит висловив користувач.

2. Обробка замовлень.

Після отримання тексту від системи розпізнавання мовлення наступним кроком є аналіз і порівняння отриманого тексту з попередньо визначеним набором команд.

Якщо текст відповідає певній команді, система виконає відповідну дію.

Якщо запит користувача складніший або потребує додаткової інтерпретації, система може звернутися до більш складних моделей для розпізнавання контексту.

3. Виконайте запит.

На основі обробленої команди система виконає відповідні дії.

Це може бути відкриття програми, виконання сценарію або пошук інформації в Інтернеті.

Наприклад, система може відкрити певний документ, запустити програму підготовки звіту або виконати пошук новин.

4. Коментарі.

Після завершення запиту система надасть відповідь користувачеві.

Це може бути у формі текстових повідомлень на екрані або голосових відповідей через синтезатор мовлення.

Коментарі дозволяють користувачеві отримати підтвердження того, що запит було виконано успішно, або повідомлення про помилку, якщо команда не була зрозуміла або не була виконана через певні обмеження.

Оскільки всі обчислення виконуються на стороні сервера, системі потрібен стабільний доступ до Інтернету для обробки запитів і забезпечення безперебійної роботи.

2.4. Архітектура системи

Архітектура системи побудована з кількох важливих модулів, кожен з яких виконує окрему функцію для забезпечення ефективної роботи програми. Ось основні модулі, що складають систему:

- 1. Модуль розпізнавання мовлення.** Цей модуль відповідає за отримання аудіосигналу від користувача через мікрофон. Після отримання сигналу він передає його на хмарний сервіс для обробки, який, в свою чергу, надає текстовий результат, що відповідає сказаному користувачем.
- 2. Модуль аналізу команд.** Після того, як текстова команда була

отримана, цей модуль здійснює її обробку та порівняння з вбудованим набором команд, що можуть бути виконані системою. Цей процес включає в себе аналіз контексту і, при необхідності, звернення до зовнішніх API для отримання додаткової інформації.

3. **Модуль генерації відповідей.** Після того, як команда оброблена, цей модуль створює відповідь для користувача. Це може бути текстова відповідь, яка виводиться на екран, або голосова, яку система озвучує за допомогою синтезатора мови. Це важливий крок для забезпечення зворотного зв'язку з користувачем і підтвердження виконання команди.
4. **Інтерфейс користувача.** Це частина системи, яка надає користувачеві зручний доступ до всіх її функцій. В інтерфейсі передбачено кілька вкладок, через які користувач може взаємодіяти з асистентом, переглядати історію запитів або налаштовувати параметри системи.

Ця модульна архітектура дозволяє кожному з етапів виконуватись незалежно, що забезпечує масштабованість і гнучкість системи.

Кожен модуль може бути легко оновлений чи покращений без потреби змінювати інші частини програми, що дає можливість для подальшого розвитку та оптимізації системи.

2.5. Інтерфейс та дизайн

- Графічний інтерфейс системи простий у використанні. Основне вікно включає:
- Поле для відображення розпізнаних голосових команд і результатів обробки.
- Кнопку для активації голосового керування.

- Панель налаштувань, що дозволяє обрати мову та персоналізувати функції.

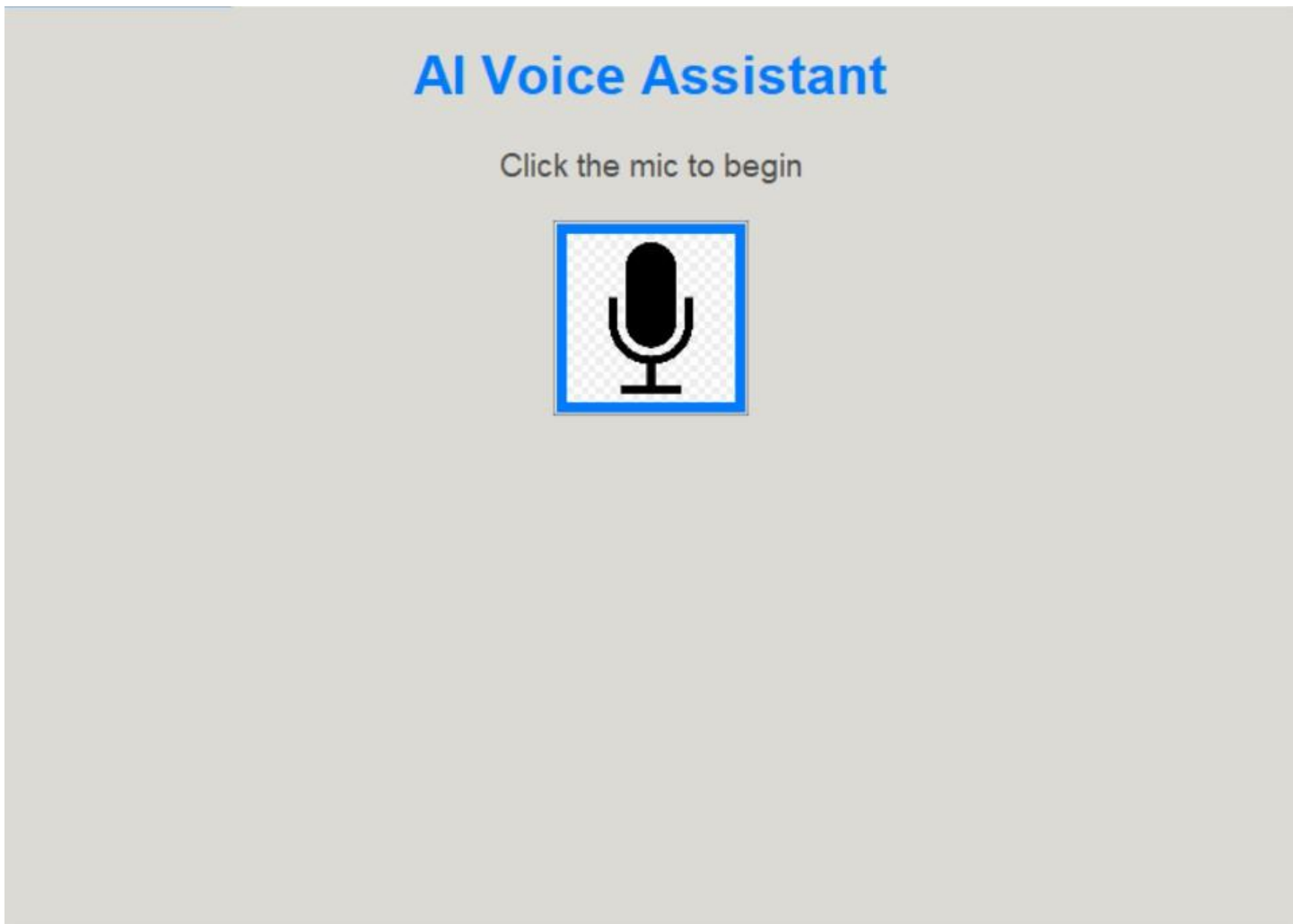


Рис. 2.1. Вигляда програми

2.6. Зберігання даних

Система не зберігає жодних даних користувача на локальному пристрої.

Усі голосові команди передаються на хмарний сервер для обробки, після чого результати негайно повертаються користувачеві.

Це забезпечує високу продуктивність і гнучкість, але вимагає дотримання правил конфіденційності та захисту даних.

2.7. Очікувані результати

Розроблена система голосового керування ПК буде інтуїтивно зрозумілою для використання користувачем, спрощуючи виконання щоденних завдань для нього. Її онлайн-функціонал дозволяє користувачам отримувати найточніші результати обробки команд, а також інтегрувати систему з іншими веб-сервісами. Ця розробка стане корисною як для звичайних користувачів, так і для професіоналів, що шукають інноваційні способи автоматизації роботи.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Діаграма прецедентів

Діаграма прецедентів відображає взаємодію користувача із системою голосового керування персональним комп'ютером. Єдиним актором є користувач, який взаємодіє з системою для виконання різноманітних завдань. Система дозволяє користувачу здійснювати операції за допомогою голосових команд, таких як відкриття програм, керування файлами, пошук інформації або налаштування параметрів. Крім того, система забезпечує зворотний зв'язок для підтвердження виконання команд чи інформування про можливі помилки. Всі дії користувача пов'язані зі зрозумілим і доступним інтерфейсом, що полегшує роботу з системою.

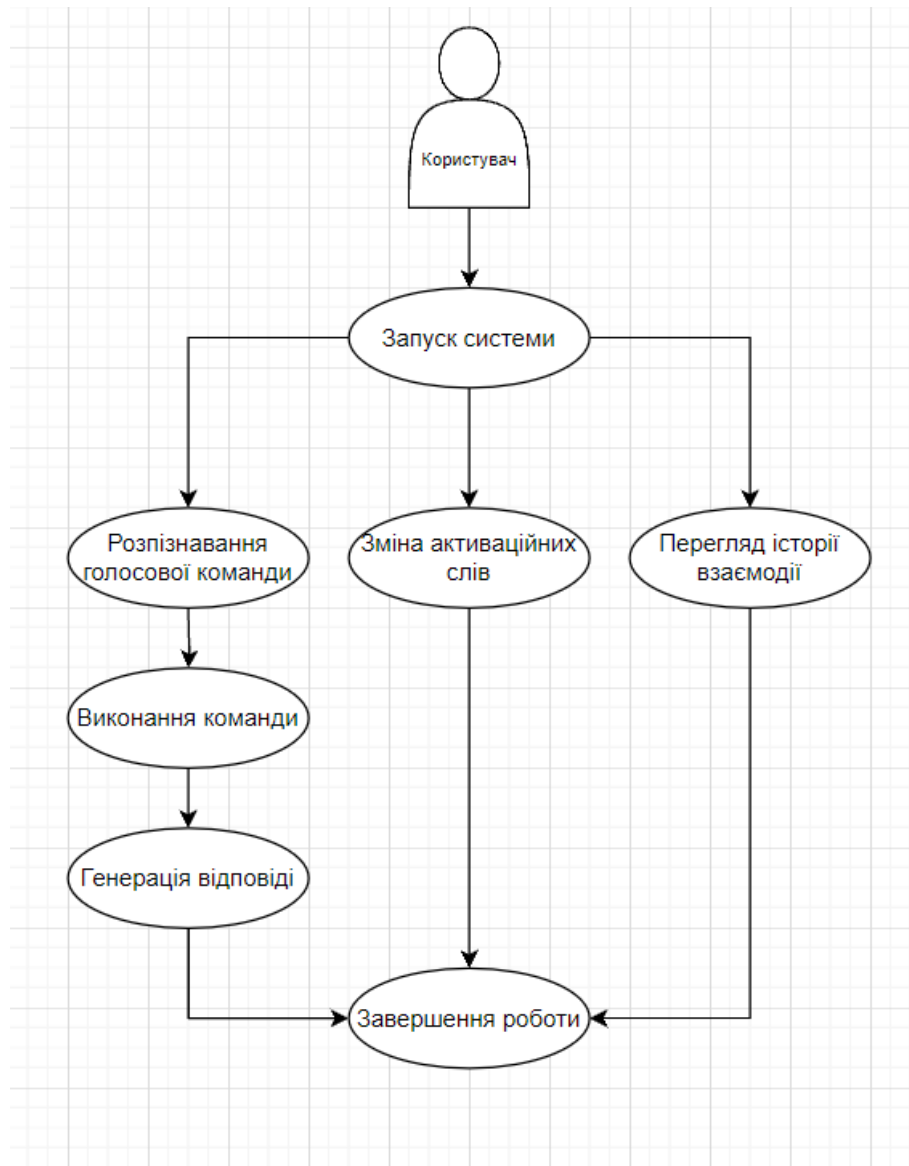


Рис. 3.1. Діаграма прецедентів

3.2. Діаграма діяльності

Діаграма діяльності ілюструє основні етапи взаємодії користувача з системою.

Процес починається зі стартового стану, що позначає ініціацію системи. Першим кроком є активація системи через натискання кнопки для початку взаємодії. Після цього система чекає на голосову команду користувача, яка ініціює розпізнавання мовлення.

Наступний крок — це розпізнавання голосу, в якому система перетворює отриманий аудіосигнал на текст. Після цього система обробляє команду,

порівнюючи отриманий текст з набором заздалегідь визначених команд. Якщо команда правильна, система виконує відповідну дію, наприклад, відкриває програму чи запускає запит в браузері.

Якщо користувач надає додаткові голосові інструкції чи запити, система продовжує виконувати команди. В кінці, коли користувач завершує взаємодію, система робить зворотний зв'язок у вигляді тексту і голосового повідомлення, підтверджуючи виконання запиту.

Цей процес завершується, коли всі запити виконано або система отримує команду на завершення. Діаграма показує весь цикл роботи з системою, починаючи з активації, розпізнавання голосу та виконання запитів, до завершення сеансу взаємодії.

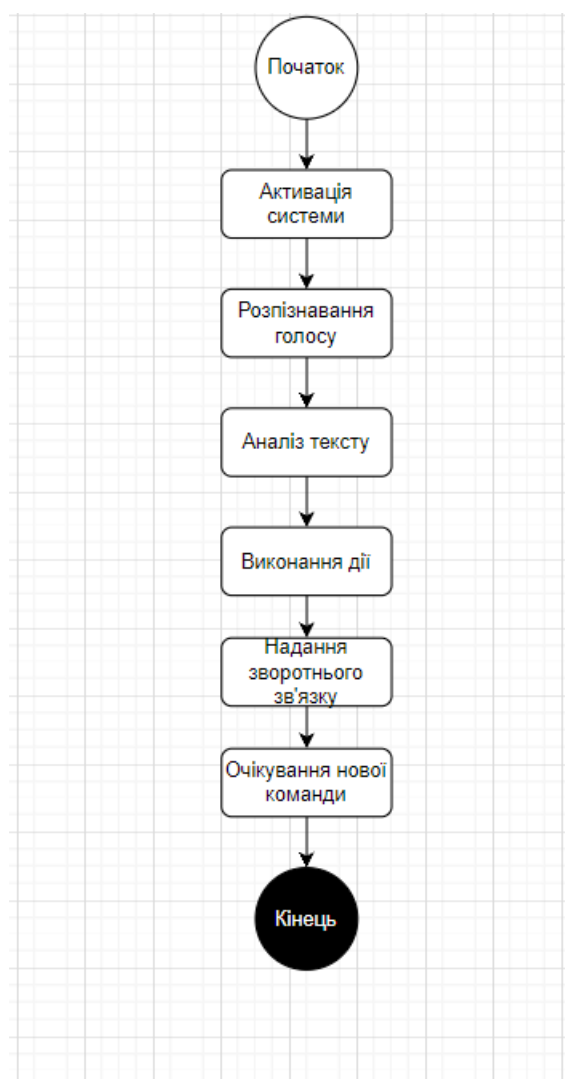


Рис. 3.2. Діаграма діяльності

3.3. Використані технології та інструменти

Розробка системи голосового управління персональним комп'ютером базується на використанні сучасних мов програмування, бібліотек та технологій, які дозволяють ефективно вирішувати поставлені завдання. Основною мовою програмування для даного проекту є Python. Ця мова була обрана через її універсальність, простоту використання, а також наявність великої кількості бібліотек, які спрощують роботу з обробкою звуку, побудовою штучного інтелекту та створенням графічного інтерфейсу.

Унікальні технології, котрі були використані в проєкті: **SpeechRecognition**: ця бібліотека має вирішальне значення для функції перетворення голосу в текст. Це дозволяє помічнику слухати мову користувача через мікрофон, обробляти аудіо та перетворювати його на текст. У коді функція `распознавателя.listen()` використовується для захоплення звуку з мікрофона, а `распознаватель.recognize_google()` використовується для перетворення вимовлених слів у текст.

gTTS (Google Text-to-Speech): gTTS використовується для створення мовлення з тексту. Після того, як помічник обробить команду користувача, він перетворює відповідь на мову та відтворює її для користувача. Це робиться за допомогою бібліотеки gTTS, яка перетворює текст на аудіофайл, а бібліотека `playsound` використовується для відтворення цього аудіофайлу.

OpenAI API: OpenAI API використовується для обробки введених користувачем даних і створення відповідей за допомогою моделі GPT-3.5. Ця інтеграція дозволяє помічнику брати участь у динамічному розмовному обміні, надаючи інформацію або виконуючи завдання на основі голосових команд користувача. Надсилаючи мову користувача OpenAI API, помічник може отримати доступ до потужної мовної моделі, яка генерує релевантні відповіді.

Запити: Бібліотека запитів використовується для отримання даних із зовнішніх API, наприклад для отримання інформації про погоду або пошуку IP-адреси користувача. Це дає змогу помічнику надавати дані в реальному часі, як-от поточну погоду чи час, на основі запитів користувача.

Playsound: Ця бібліотека використовується для відтворення аудіофайлів. Після перетворення тексту відповіді на мовлення за допомогою gTTS playsound використовується для відтворення згенерованого аудіо для користувача.

Потокова обробка. Потокова обробка використовується для забезпечення безперебійної роботи голосового помічника навіть під час виконання трудомістких завдань, таких як розпізнавання мовлення або запит до зовнішніх API. Запускаючи певні функції в окремих потоках, основний інтерфейс залишається чуйним, а помічник може слухати команди, не блокуючись іншими процесами.

Команди ОС: код також містить команди системного рівня, які дозволяють помічнику виконувати такі дії, як вимикання або перезавантаження комп'ютера. Ці команди виконуються через функцію Python `os.system()`, що дозволяє помічнику взаємодіяти з операційною системою.

Ці технології разом створюють безперебійний голосовий інтерфейс, який дозволяє користувачам взаємодіяти зі своїм комп'ютером у режимі «вільні руки», надаючи функціональні можливості для таких завдань, як перевірка часу, погоди або навіть керування системними операціями, наприклад вимиканням або перезавантаженням комп'ютера.

Графічний інтерфейс системи побудований за допомогою бібліотеки Tkinter, яка входить до стандартного пакету Python. Tkinter надає простий і водночас функціональний інструментарій для створення зручного графічного інтерфейсу користувача. Програма використовує структуру вкладок, яка забезпечує логічний поділ функцій на різні розділи, такі як голосовий помічник, історія запитів та налаштування.

Для реалізації функції розпізнавання голосу була використана бібліотека SpeechRecognition, яка дозволяє взаємодіяти з мікрофоном і отримувати текстову версію того, що говорить користувач. Для генерації звукових відповідей використовується бібліотека Google Text-to-Speech (gTTS). Це дозволяє системі відповідати користувачеві не тільки текстом, а й голосом, забезпечуючи інтерактивність.

В основі логіки роботи асистента лежить використання OpenAI API. Цей API надає можливість використовувати передові моделі штучного інтелекту для розуміння запитів користувача та генерації відповідей. Завдяки цьому система може вести природний діалог і відповідати на складні запитання.

Для отримання додаткових даних, наприклад, інформації про погоду, використовується бібліотека запитів, яка дозволяє надсилати HTTP-запити до зовнішніх API. Це забезпечує доступ до актуальної інформації в режимі реального часу.

Історія діалогів, а також налаштування системи зберігаються у форматі JSON. Використання цього формату дозволяє зберігати дані в простому і зручному для обробки вигляді.

3.4. Архітектура програми

Програма організована таким чином, щоб забезпечити максимальну зручність у використанні та розширюваність. Інтерфейс користувача представлений у вигляді вкладок, кожна з яких виконує окрему функцію. Перша

вкладка є основним інструментом для взаємодії з асистентом. Тут користувач може активувати систему, задавати запитання або команди і отримувати відповіді. Друга вкладка дозволяє переглядати історію попередніх діалогів. Це корисно для відстеження відповідей або повторного використання інформації. Третя вкладка присвячена налаштуванням, зокрема зміні ключового слова активації.

Логіка роботи асистента базується на потоковій обробці даних. Це означає, що голос користувача розпізнається в окремому потоці, що дозволяє програмі продовжувати працювати навіть під час тривалого розпізнавання або обробки команд.

```
def toggle_listening(self):

    global listening

    listening = not listening

    if listening:

        self.status_label.config(text="Listening started...")

        threading.Thread(target=self.recognize_and_process, daemon=True).start()

        self.start_button.config(text="Stop Listening")

    else:

        self.status_label.config(text="Listening stopped.")

        self.start_button.config(text="Start Listening")
```

Після розпізнавання голосу фраза передається на аналіз у OpenAI API, де генерується відповідь. Відповідь повертається у текстовій формі, після чого вона озвучується за допомогою gTTS.

```
# Показуємо результат поступово і читаємо його голосом

self.gradually_display_text(response)

try:

    tts = gTTS(text=response, lang=language, slow=False)

    tts.save("response.mp3")
```

```

playsound("response.mp3")

os.remove("response.mp3")

except Exception as e:

    messagebox.showerror("Audio Error", f"Could not play audio: {e}")

```

Робота програми організована таким чином, щоб мінімізувати затримки і забезпечити інтерактивність. Наприклад, якщо користувач вимовляє фразу, яка включає ключове слово активації, програма негайно реагує і виконує відповідну дію.

```

if activation_word in phrase: # Перевіряємо чи є ключове слово

    if "time" in phrase:

        response = self.get_time()

    elif "weather" in phrase:

        response = self.get_weather()

    elif "joke" in phrase:

        response = self.get_joke(phrase)

    elif "shutdown" in phrase: # Команда для вимкнення комп'ютера

        response = self.shutdown_computer()

    elif "restart" in phrase: # Команда для перезавантаження комп'ютера

        response = self.restart_computer()

    else:

        response = self.get_com(phrase)

# Додаємо нову відповідь до історії діалогу

dialog_history.append({"role": "user", "content": phrase})

dialog_history.append({"role": "assistant", "content": response})

self.update_dialog_history()

else:

    response = "No activation word detected."

```

3.5. Розгортання проєкту

Проєкт розгортається на локальному комп'ютері і потребує лише базового набору інструментів. Для запуску програми користувачеві потрібно встановити Python версії 3.8 або новішої, а також усі необхідні бібліотеки, які вказані у файлі вимог. Для цього використовується команда `pip install -r requirements.txt`. Після встановлення бібліотек користувач може запустити програму за допомогою файлу `main.py`. Додатково необхідно зареєструватися у сервісі OpenAI, отримати API-ключ і вказати його у конфігураційному файлі.

3.6. Переваги і можливості для розвитку

Система має кілька важливих переваг, які роблять її зручною і ефективною. Вона є легкою для використання, має мінімальні вимоги до обладнання і забезпечує інтерактивність завдяки голосовому зворотному зв'язку. У майбутньому проєкт може бути значно розширений. Наприклад, можна додати підтримку декількох мов для голосових команд і відповідей, інтегрувати систему з іншими платформами, такими як розумний дім або мобільні пристрої. Також можливе впровадження елементів машинного навчання для покращення точності розпізнавання голосу і розуміння контексту запитів. Хорошим варіантом буде запровадження можливості купівлі планів на обмеження часу чи запитів для розмов – тобто капіталізація продукту. З елементів системи котрі можна додати чи покращити також – це поглиблена інтеграція з ОС. Тобто, можливість змінювати файли на усьому комп'ютері.

Таким чином, розроблена система є не лише сучасним і функціональним інструментом, але й має великий потенціал для подальшого розвитку та адаптації до потреб користувачів.

ВИСНОВКИ

Розробка системи голосового керування є важливим кроком у розвитку інтерфейсів людина-комп'ютер, оскільки надає користувачам зручніший і більш інтуїтивно зрозумілий спосіб взаємодії з технічними пристроями. Особливо важливим є застосування такої технології для різних категорій користувачів, зокрема людей з обмеженими фізичними можливостями, студентів, бізнес-користувачів і звичайних користувачів, які хочуть оптимізувати свою роботу та зекономити час.

Розроблена система використовує сучасні хмарні технології для обробки голосових команд, що забезпечує високу точність і швидкість розпізнавання мови. Всі основні етапи її роботи, включаючи розпізнавання голосу, аналіз команд, виконання запитів і надання зворотного зв'язку, реалізовані через інтеграцію з потужними сервісами обробки мовлення, такими як Google Speech-to-Text або OpenAI.

Завдяки модульній архітектурі система є гнучкою та масштабованою, що дозволяє легко адаптувати її для різних потреб і умов. Кожен модуль виконує чітко визначену задачу, що забезпечує ефективну обробку запитів користувачів, а також можливість подальшого розвитку і вдосконалення системи.

Перспективи для подальшого розвитку системи включають додавання нових мов, покращення алгоритмів розпізнавання та обробки команд, а також інтеграцію з іншими платформами і пристроями, що дозволить значно розширити можливості її застосування. Система може бути адаптована для використання в різних сферах, таких як освіта, медицина, бізнес і багато інших.

В цілому, розробка голосової системи управління відповідає сучасним вимогам автоматизації та забезпечення доступу до технологій для різних категорій користувачів, що робить її корисним і перспективним інструментом у багатьох сферах діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OpenAI. GPT-3.5 API Documentation [Електронний ресурс] — Режим доступу: URL: <https://platform.openai.com/docs/models/gpt-3.5>
2. Google Cloud. Speech-to-Text API [Електронний ресурс] — Режим доступу: URL: <https://cloud.google.com/speech-to-text>
3. gTTS (Google Text-to-Speech) [Електронний ресурс] — Режим доступу: URL: <https://pypi.org/project/gTTS/>
4. Tkinter Documentation [Електронний ресурс] — Режим доступу: URL: <https://docs.python.org/3/library/tkinter.html>
5. Python Programming Language [Електронний ресурс] — Режим доступу: URL: <https://www.python.org/>
6. OpenWeather API Documentation [Електронний ресурс] — Режим доступу: URL: <https://openweathermap.org/api>
7. Requests Library Documentation [Електронний ресурс] — Режим доступу: URL: <https://docs.python-requests.org/en/master/>
8. SpeechRecognition Python Library [Електронний ресурс] — Режим доступу: URL: <https://pypi.org/project/SpeechRecognition/>
9. Python's os Module [Електронний ресурс] — Режим доступу: URL: <https://docs.python.org/3/library/os.html>
10. Microsoft Azure Cognitive Services Speech API [Електронний ресурс] — Режим доступу: URL: <https://learn.microsoft.com/en-us/azure/cognitive-services/speech-service/>

ДОДАТКИ

1. Функція для розпізнавання голосових команд

```
def recognize_and_process(self):
    """Розпізнає голосову команду і обробляє її."""
    global listening
    while listening:
        response = ""
        try:
            with sr.Microphone() as source:
                self.status_label.config(text="Listening...")
                self.root.update()
                audio = recognizer.listen(source)
                phrase = recognizer.recognize_google(audio).lower()
                self.status_label.config(text=f"You said: {phrase}")
                self.root.update()

            if activation_word in phrase: # Перевіряємо чи є ключове слово
                if "time" in phrase:
                    response = self.get_time()
                elif "weather" in phrase:
                    response = self.get_weather()
                elif "joke" in phrase:
                    response = self.get_joke(phrase)
                elif "shutdown" in phrase: # Команда для вимкнення комп'ютера
                    response = self.shutdown_computer()
                elif "restart" in phrase: # Команда для перезавантаження комп'ютера
                    response = self.restart_computer()
                else:
                    response = self.get_com(phrase)

            # Додаємо нову відповідь до історії діалогу
            dialog_history.append({"role": "user", "content": phrase})
            dialog_history.append({"role": "assistant", "content": response})
            self.update_dialog_history()
        except sr.UnknownValueError:
            response = "No activation word detected."
```

```

        response = "I didn't catch that. Please try again."
except sr.RequestError as e:
    response = f"Error with the speech recognition service: {e}"
except Exception as e:
    response = f"An error occurred: {e}"

# Показуємо результат поступово і читаємо його голосом
self.gradually_display_text(response)
try:
    tts = gTTS(text=response, lang=language, slow=False)
    tts.save("response.mp3")
    playsound("response.mp3")
    os.remove("response.mp3")
except Exception as e:
    messagebox.showerror("Audio Error", f"Could not play audio: {e}")

```

Ця функція відповідає за основну логіку розпізнавання голосу. Вона постійно слухає мікрофон, коли система активована. Після отримання аудіо сигналу за допомогою бібліотеки `speech_recognition`, текст розпізнається через Google Speech-to-Text API. Після цього програма перевіряє, чи містить розпізнаний текст ключове слово активації. Якщо так, програма аналізує команду (наприклад, час, погода, жарт, тощо) і виконує відповідну дію. Крім того, генерується голосова відповідь, яка програватиметься через аудіо пристрій.

2. Функція генерації голосових відповідей

```
def gradually_display_text(self, text): """Поступово виводить текст на екран""" self.result_label.config(text="")  
for i in range(len(text)): self.result_label.config(text=text[:i+1]) self.root.update_idletasks() time.sleep(0.05)
```

Ця функція відповідає за поступове відображення текстових відповідей користувачу. Це дає змогу створити враження, що асистент “пише” відповідь по буквах, що робить взаємодію більш інтерактивною. Функція використовує методи для оновлення інтерфейсу користувача в реальному часі.