

Хмельницький національний університет
Факультет програмування
та комп'ютерних і телекомунікаційних систем
Кафедра інженерії програмного забезпечення

ДИПЛОМНИЙ ПРОЕКТ

Програмне забезпечення для голосового управління

процесами комп'ютера

Назва теми

Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр ДППЗ.170115.01.14.ПЗ

Виконав студент IV курсу група ПЗ-17-1


Підпис

В.А. Павлюк
Ініціали, прізвище

Керівник канд. техн. наук, доцент
Науковий ступінь, звання


Підпис

Ю.В. Форкун
Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент


Підпис

Г. І. Радельчук
Ініціали, прізвище

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення


Підпис

Л. П. Бедратюк
Ініціали, прізвище


10 06 2021 р.

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри 
Л. П. Бедратюк

05 02 2021 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)

Павлюку Віталію Андрійовичу

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Програмне забезпечення для голосового управління процесами комп'ютера

Керівник проєкту (роботи) Форкун Юрій Вікторович, канд. техн. наук, доцент
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 05.02.2021 р. № 11

2. Строк подання студентом проєкту (роботи) на кафедру 01.06.2021 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики

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



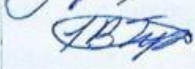
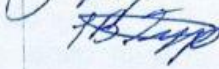
Дослідження та аналіз предметної області, проектування програмного забезпечення,

програмна реалізація та тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Радельчук Г.І., канд. техн. наук, доцент		
Антиплагіат	Гурман І.В., канд. техн. наук, доцент		

7. Дата видачі завдання « 05 » лютого 2021р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітки
1 Ознайомлення з тематикою дипломного проєктування (ДП), визначення та узгодження індивідуальних тем ДП	01.12 – 30.12.2020	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2021	
3 Проєктування програмного забезпечення	01.02 – 28.02.2021	
4 Програмна реалізація	01.03 – 10.04.2021	
5 Тестування програмного забезпечення	11.04 – 30.04.2021	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2021	
7 Попередній захист ДП	Травень 2021 (згідно графіка)	
8 Перевірка ДП на плагіат, нормконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки)	26.05 – 30.05.2021	
9 Підготовка до захисту та захист ДП	з 01.06.2021	

Студент


Підпис

В.А. Павлюк
Ініціали, прізвище

Керівник проєкту (роботи)


Підпис

Ю.В. Форкун
Ініціали, прізвище

АНОТАЦІЯ

Тема дипломного проекту: «Програмне забезпечення для голосового управління процесами комп'ютера».

Автор проекту: Павлюк Віталій Андрійович.

Керівник проекту: Форкун Юрій Вікторович.

Пояснювальна записка: 86 с., 25 рис., 7 табл., 3 дод., 23 джерела.

Графічна частина: 12 презентаційних слайдів.

ГОЛОСОВИЙ АСИСТЕНТ, API, TTS, PYTHON, RHVOICE, GOOGLE VOICE RECOGNITION, PUCHARM.

Метою проекту є розробка програмного забезпечення для голосового управління процесами комп'ютера, яке має зручний та мінімалістичний інтерфейс. У дипломному проекті виконано порівняння наявних програмних продуктів, опис модулів та процеси компілювання та тестування.

Для реалізації програмного забезпечення використано мову програмування Python, розпізнавання мовлення Google Voice Recognition та мовний рушій Anatol.

В результаті проектування здійснена програмна реалізація забезпечення для голосового управління процесами комп'ютера.



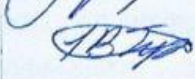
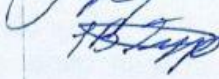
9.06.21

Дата



Підпис

6. Консультанти розділів дипломного проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Радельчук Г.І., канд. техн. наук, доцент		
Антиплагіат	Гурман І.В., канд. техн. наук, доцент		

7. Дата видачі завдання « 05 » лютого 2021р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Приміти
1 Ознайомлення з тематикою дипломного проєктування (ДП), визначення та узгодження індивідуальних тем ДП	01.12 – 30.12.2020	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2021	
3 Проєктування програмного забезпечення	01.02 – 28.02.2021	
4 Програмна реалізація	01.03 – 10.04.2021	
5 Тестування програмного забезпечення	11.04 – 30.04.2021	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2021	
7 Попередній захист ДП	Травень 2021 (згідно графіка)	
8 Перевірка ДП на плагіат, нормконтроль, отримання відгуків та рецензій. Брошурування (зшиття) пояснювальної записки)	26.05 – 30.05.2021	
9 Підготовка до захисту та захист ДП	з 01.06.2021	

Студент


Підпис

В.А. Павлюк
Ініціали, прізвище

Керівник проєкту (роботи)


Підпис

Ю.В. Форкун
Ініціали, прізвище

ЗМІСТ

Перелік скорочень	6
Вступ	7
1 Дослідження предметної області та постановка задачі	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	10
1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання	17
2 Проектування програмного забезпечення	20
2.1 Архітектура та функціональна структура додатка	21
2.2 Обґрунтування вибору технології для розробки	24
2.2.1 Вибір мови програмування для розробки	24
2.2.2 Вибір мовного рушія	26
2.2.3 Вибір бібліотек для розробки програмного забезпечення	28
2.2.4 Вибір засобів для розробки програмного забезпечення	29
2.3 Реалізація голосового інтерфейсу програми	31
3 Програмна реалізація	34
3.1 Детальне проектування даних	34
3.2 Реалізація модулів системи	35
3.2.1 Оператор import та функція def()	35
3.2.2 Словники	37
3.2.3 Бібліотека fuzzywuzzy	38
3.2.4 Модуль platform	39

ДППЗ.170115.01.14.ВД							
мн	Арх.	№ докум.	Підпис	Дата	Лист	Арх	Аркуші
					1	1	1
Виконав		Павлюк В.А.	<i>[Signature]</i>	9.06	ХНУ. ППЗ-17-1		
Керівник		Форжун Ю.В.	<i>[Signature]</i>	9.06			
Н. контро.		Рабельчук Г.І.	<i>[Signature]</i>	10.06			
Зав. каф.		Бабратюк Л.П.	<i>[Signature]</i>	10.06			

Програмне забезпечення для
голосового управління
процесами комп'ютера
Відомість документів

3.2.5 Модулі psutil та shutil.....	40
3.2.6 JSON.....	42
3.2.7 OpenCV.....	44
3.2 Компіляція проекту	45
3.3 Інтерфейс користувача	48
3.4 Інструкція користувача	49
4 Тестування програмного забезпечення.....	53
4.1 Вибір та обґрунтування методів тестування додатку	54
4.2 Доведення працездатності програми	55
4.3 Результати тестування програмного забезпечення	57
Висновки.....	62
Перелік джерел посилання	64
Додаток А Технічне завдання	66
Додаток Б Фрагменти коду програмного забезпечення.....	73
Додаток В Презентаційні слайди.....	78

ПЕРЕЛІК СКОРОЧЕНЬ

ВВ	–	Варіанти використання
ПЗ	–	Програмне забезпечення
API	–	Application programming interface
CLI	–	A command-line interface
GUI	–	The graphical user interface
IDE	–	Integrated development environment
SAPI	–	Microsoft Speech Application programming interface
STT	–	Speech-To-Text
TCP	–	The Transmission Control Protocol
TTS	–	Text-to-Speech
UDP	–	The User Datagram Protocol
UML	–	Unified Modeling Language
WWW	–	World Wide Web
ZUI	–	Zooming user interface

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		6

ВСТУП

У сучасному світі важко уявити людину без персонального комп'ютера, смартфона, планшета чи ноутбука. Всі ці пристрої полегшують людям життя, дозволяючи шукати інформацію в Інтернеті, спілкуватись з іншими людьми на відстані чи просто проводити відпочинок, переглядаючи відео. Але іноді для певних дій потрібна автоматизація. Для таких випадків було вигадано голосових помічників, які допоможуть спростити використання певних функцій при користуванні пристроєм.

На даний момент голосовий помічник відіграє велику роль у житті користувача смартфона чи персонального комп'ютера. Він дозволяє телефонувати чи відправляти повідомлення, шукати інформацію, дивитися погоду за допомогою голосового управління. З кожним роком голосові помічники отримують все більше функцій, а також їх стає більше на ринку. Але більшість з них потребують додаткове програмне забезпечення для правильної роботи.

Актуальність теми полягає у тому, що на сьогодні голосових помічників для персонального комп'ютера, які використовують українську мову, не існує. Голосовий помічник дозволяє виконувати не лише невеликі команди, але й, при можливості повноцінно керувати певними функціями персонального комп'ютера. Таке програмне забезпечення повинно мати мінімалістичний дизайн, інтуїтивно зрозумілий користувачу.

Метою проекту є розробка програмного забезпечення для голосового управління процесами комп'ютера, яке має мінімалістичний інтерфейс.

Для реалізації проекту необхідно виконати наступні завдання:

- визначити специфіку предметної області;
- провести аналіз подібного програмного забезпечення та визначити їх основні недоліки;
- встановити особливості використання модулів та бібліотек потрібних для розробки програмного забезпечення.

					ДППЗ.170115.01.14.ПЗ	Арк.
						7
Зм.	Арк	№ докум.	Підпис	Дата		

– провести тестування програмного забезпечення.

Результатом виконання дипломного проекту є готове програмне забезпечення для голосового управління процесами комп'ютера.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Голосовий помічник – програмне забезпечення, яке допомагає користувачу автоматизувати певні дії під час роботи з певним пристроєм за допомогою голосу. Сучасні голосові помічники дають змогу шукати інформацію в інтернеті, дізнаватись погоду, замовляти товар в інтернеті, телефонувати, а також навіть відповідати на дзвінки чи записувати користувача в перукарню. Існують такі види голосових помічників:

- для Android: Google Assistant, «Аліса» від Яндекс, «Маруся» від Mail.Ru Group, Vixby, Alexa;
- для iOS: Siri, Alexa, Google Assistant, «Аліса» від Яндекс, «Маруся» від Mail.Ru Group;
- для ПК: «Аліса» від Яндекс, Cortana.

Основною проблемою таких голосових помічників є безпека доступу до особистих даних сторонніми людьми. Деякі помічників вдавалось активувати за допомогою високочастотних звуків або шуму, який був схожий на людську мову, наприклад, слова в пісні.

Також голосовими помічниками не варто користуватись дітям. Деякі з голосових помічників мають змогу замовляти товари з магазину, наприклад, «Alexa» від Amazon, яка може випадково увімкнутись дитиною та виконати покупку без відома дорослих.

Опираючись на те, що голосових помічників для персональних комп'ютерів, які підтримують україномовний синтезатор мовлення не існує, доцільно буде реалізувати програмне забезпечення для голосового управління процесами комп'ютера.

					ДППЗ.170115.01.14.ПЗ	Арк.
						9
Зм.	Арк	№ докум.	Підпис	Дата		

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

На сьогоднішній день існує багато програм, які дозволяють користувачу автоматизувати певні дії за допомогою голосового управління.

На рисунку 1.1 зображено інтерфейс помічника від компанії «Google», який носить назву «Google Assistant». На даний момент це один з найпопулярніших голосових помічників. Він працює на пристроях, які управляються операційною системою Android та iOS. Помічник дозволяє керувати пристроєм, шукати інформацію в інтернеті, телефонувати, а також відповідати автоматично на дзвінки (ця функція недоступна в Україні і на даний момент працює лише на фірмових пристроях компанії Google Pixel). На даний момент помічник не вміє розпізнавати українську мову.

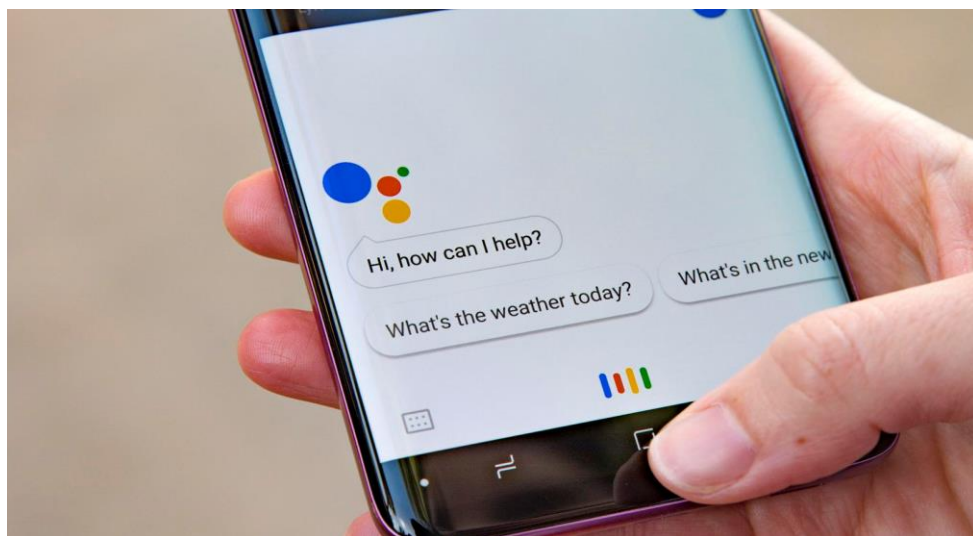


Рисунок 1.1 – Голосовий помічник «Google Assistant»

На рисунку 1.2 зображено інтерфейс помічника від компанії Apple, який носить назву «Siri». Даний помічник був створений у 2011 році і працює лише на пристроях від компанії Apple. Функціонал подібний з Google Assistant. Перевагою даного помічника є те, що він доступний «з коробки», в той час, як для коректної

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		10

роботи Google Assistant потрібно установити фірмовий додаток Google з магазину Play Store (Android) чи App Store (iOS).

Google Assistant має одну з основних особливостей – він доступний як і на Android, так і на iOS.

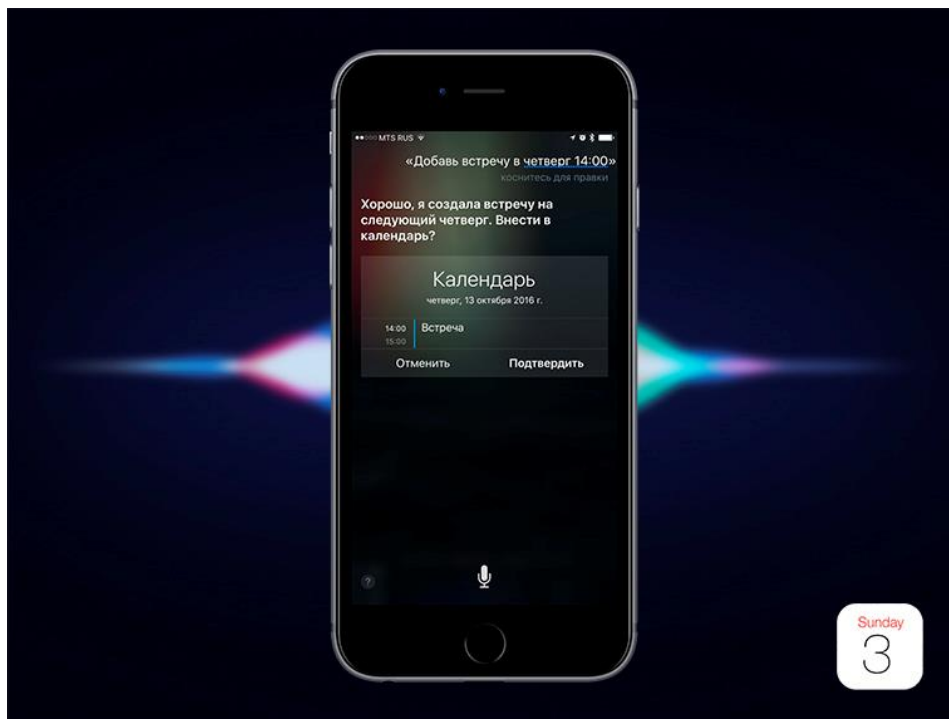


Рисунок 1.2 – Голосовий помічник «Siri»

На рисунку 1.3 зображено інтерфейс помічника «Алиса» від компанії «Яндекс» (версія для смартфонів), а на рисунку 1.4 – версію для персональних комп'ютерів. Даний помічник був створений у 2017 році. На відміну від Google Assistant та Siri даний помічник не вміє на даний момент управляти пристроєм, але відрізняється високоякісним синтезом мовлення, який компанія «Яндекс» створила спеціально для цього помічника. «Алиса» вміє шукати інформацію в інтернеті, читати новини або просто спілкуватись. На території України працює лише при ввімкненому VPN чи проксі. Для роботи даного помічника потрібен браузер чи лаунчер від компанії «Яндекс».

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

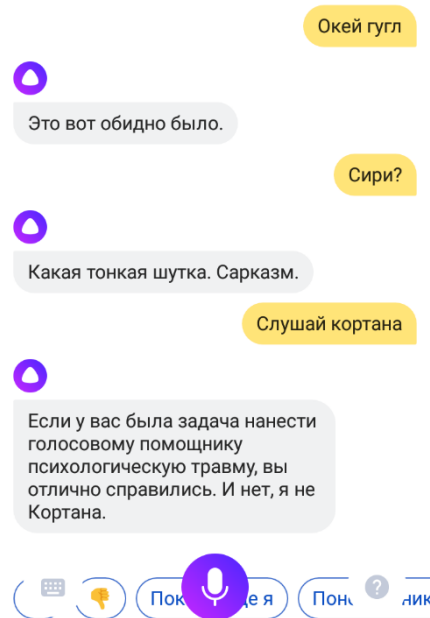


Рисунок 1.3 – Голосовой помічник «Алиса» (версія для смартфонів)

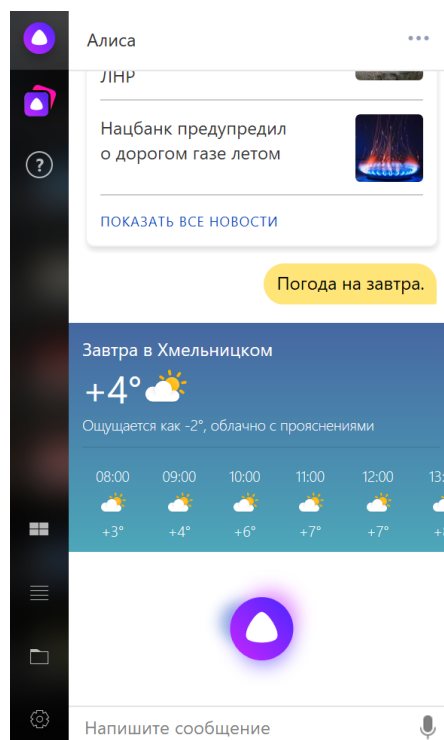


Рисунок 1.4 – Голосовой помічник «Алиса» (версія для ПК)

На рисунку 1.5 зображено інтерфейс помічника «Vixby» від компанії Samsung. Голосовой помічник був створений у 2017 році і працює лише на пристроях компанії Samsung. На початку 2020 року компанія Samsung

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

відмовилась від підтримки Vixby на пристроях під управлінням Android Oreo та нижче. Українська та російська мови недоступні.

Vixby поділяється на три частини, такі як «Vixby Voice», «Vixby Vision» та «Vixby Home». «Vixby Voice» – назва методу активації Vixby шляхом його виклику або довгого натискання спеціальної кнопки Vixby, розташованої під кнопками гучності. За деякий час до виходу телефону, кнопка Vixby була перепрограмовано і могла бути налаштована для відкриття інших додатків або помічників, таких як Google Assistant.

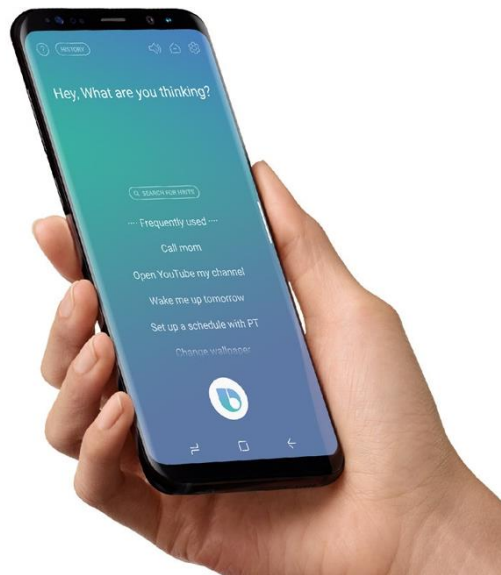


Рисунок 1.5 – Голосовий помічник «Vixby»

На рисунку 1.6 зображено зображено інтерфейс помічника «Маруся» від компанії «Mail.ru Group». Даний помічник був створений у 2019 році. Як і «Аліса», «Маруся» не вміє управляти пристроєм, але не потребує браузера чи лаунчера. «Маруся» вміє шукати інформацію в інтернеті, дізнаватись погоду, спілкуватись, запускати ігри. Основним недоліком помічника є тісний взаємозв'язок із сервісами «ВКонтакте», наприклад прослуховування музики. На території України даний голосовий помічник працює лише при увімкненому VPN чи проксі.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

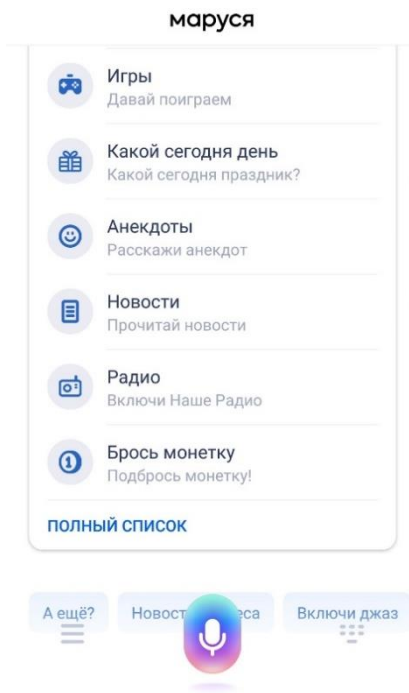


Рисунок 1.6 – Голосовой помічник «Маруся»

На рисунку 1.7 зображено інтерфейс помічника «Cortana» від компанії Microsoft. Помічник був створений у 2014 році і працює лише під управління операційної системи Windows від компанії Microsoft. Голосовой помічник на даний момент недоступний в Україні.

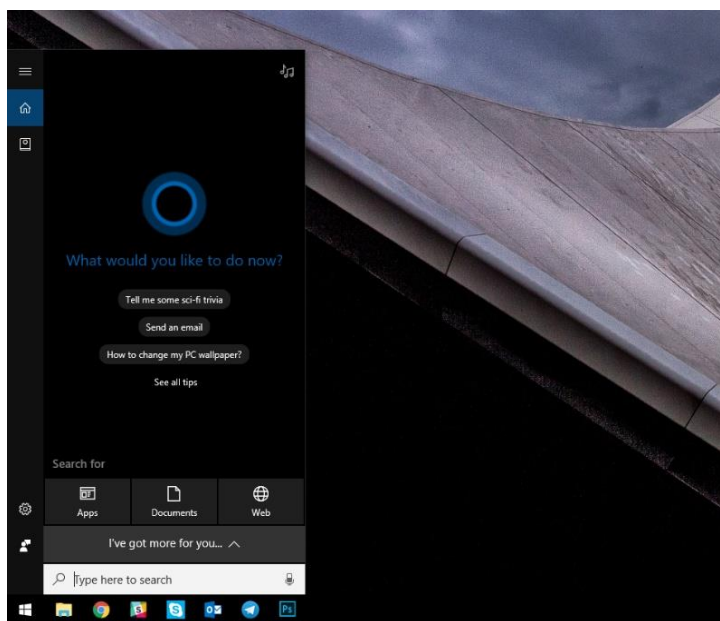


Рисунок 1.7 – Голосовой поміник «Cortana»

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14

Розглянемо особливості кожного з цих голосових помічників у порівняльній характеристиці ПЗ у таблиці 1.1:

Таблиця 1.1 – Порівняльна характеристика ПЗ

Назва	Платформа	Підтримка української мови	Ким розроблено	Особливості
1	2	3	4	5
Google Assistant	Android/iOS	Ні	Google	Потрібен Google Account. Не підтримує розпізнавання мовлення українською. В пристроях Pixel може автоматично відповідати на дзвінок та відповідати користувачу.
Siri	iOS	Ні	Apple	Працює тільки на пристроях компанії Apple. Української на даний момент не підтримує. Вміє телефонувати, шукати інформацію в інтернеті
Аліса	Android/iOS/PC	Ні	Яндекс	На смартфонах працює тільки при наявності браузера чи лаунчера від компанії «Яндекс». Вміє шукати інформацію, дізнаватись погоду, грати в ігри, а також зачитувати

Продовження таблиці 1.1

1	2	3	4	5
Аліса	Android/iOS/PC	Ні	Яндекс	новини. В Україні працює тільки при наявності VPN чи проксі. Українську мову не підтримує.
Bixby	Android	Ні	Samsung	Працює лише на пристроях компанії «Samsung». Активується за допомогою спеціальної кнопки на пристрої. Українська мова недоступна.
Маруся	Android/iOS	Ні	Mail.ru Group	На даний момент виконувати основні функції пристрою не вміє (наприклад, зателефонувати комусь і т.д.). Вміє шукати інформацію, новини, дізнаватись погоду. В Україні працює тільки при наявності VPN чи проксі. Українську мову не підтримує.
Alexa	Android/iOS	Ні	Amazon	Використовується для управління розумним будинком, для замовлення товарів з Amazon. Вміє шукати інформацію та новини. Не підтримує українську.

Кінець таблиці 1.1

1	2	3	4	5
Cortana	Windows	Ні	Microsoft	Вміє шукати інформацію, погоду, перевіряти пошту. Інтегрована в Windows, Недоступна в Україні

Згідно проведеному аналізу подібних програмних продуктів, програмне забезпечення повинно:

- підтримувати розпізнавання української мови;
- підтримувати український синтез мовлення;
- споживати мінімальну кількість ресурсів;
- бути функціональним;
- бути простим у користуванні.

1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання

Діаграма варіантів використання дозволяє визначити залежність між різними варіантами використання та акторами, що приймають участь у процесі.

Варіанти використання або як ще їх називають прецеденти показують поведінку програмної системи та її реакцію на зовнішні фактори.

Створення діаграми варіантів використання допоможе краще зрозуміти логіку програмного забезпечення та полегшити його розробку.

Наступним кроком буде опис акторів програмного забезпечення та варіантів використання.

У таблиці 1.2 показано опис акторів програмного продукту, а у таблиці 1.3 – варіанти використання.

Таблиця 1.2 – Опис акторів програмної системи

Актор	Короткий опис
Користувач	Має змогу вводити команди за допомогою голосового введення

Таблиця 1.3 – Опис варіантів використання програмної системи

Актор	Найменування ВВ	Опис ВВ
Користувач	Введення голосової команди	Користувач має змогу вводити команду за допомогою голосу
	Обробка голосової команди	Користувач отримує перетворену голосову команду у вигляді тексту
	Виконання команди	Користувач отримує виконану програмою команду

На рисунку 1.8 зображено діаграму прецедентів для голосового помічника:

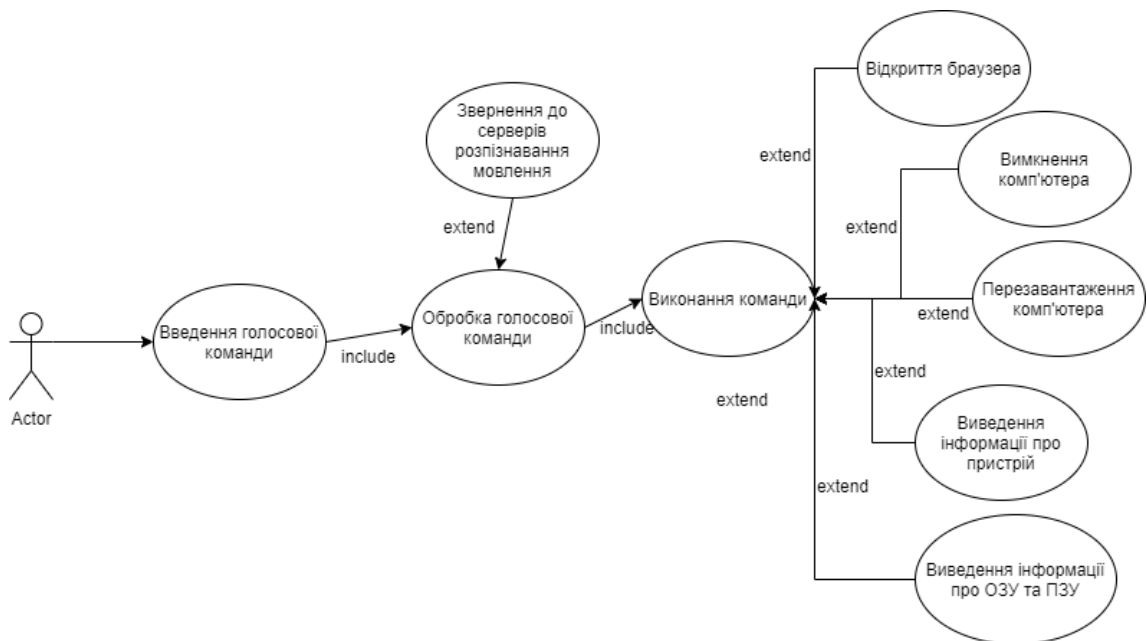


Рисунок 1.8 – Діаграма варіантів використання

Отже, було проведено аналіз предметної області та встановлено, що голосові помічники відіграють значну роль у житті користувачів смартфонів та персональних комп'ютерів, а також допомагають полегшити пошук інформації та автоматизувати певні дії. Усі вони повинні бути продуктивними та легкими у користуванні. Також у ході аналізу програмного продукту було розроблено технічне завдання (додаток А).

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		19

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Архітектура та функціональна структура додатка

Для розробки програмного забезпечення були поставлені такі задачі:

- введення даних;
- обробка даних;
- виведення даних;
- обробка помилок;
- мінімалістичний інтерфейс.

Програмне забезпечення, що розробляється, є десктопним додатком. Десктопний додаток – це програмне забезпечення, яке працює як єдина та незалежна програма. Десктопний додаток потребує наявності людини, яка ним управляє.

Додатки існують таких типів:

- веб-додаток;
- мобільні додатки;
- додатки для робочого столу (десктопні).

Десктопні додатки мають більш розширений функціонал, ніж мобільні, а також працювати як онлайн, так і оффлайн, на відміну від веб-додатків, які потребують швидкого з'єднання з мережею. Вони потребують інсталяції на персональний комп'ютер або можуть бути переносними (Portable)

Дуже часто цей тип додатків можна вважати уразливим від дій користувача, адже користувач має змогу змінювати та редагувати файли програмного забезпечення, що дуже часто призводить до його неправильного функціонування, через яке користувачу буде необхідно переустановлювати його.

Для кращого розуміння структури буде побудовано спрощену схему, яка показуватиме загальну роботу програмного продукту. Ця схема допоможе показати, як програма повинна працювати, не углиблюючись у деталі. Створення схеми допоможе спростити створення UML діаграм для програмного продукту.

На рисунку 2.1 показано схему роботи голосового помічника, який розробляється у рамках дипломного проекту.

					ДППЗ.170115.01.14.ПЗ	Арк.
						20
Зм.	Арк	№ докум.	Підпис	Дата		



Рисунок 2.1 – Схема роботи програмного забезпечення

Відштовхуючись від цієї схеми буде створено UML діаграми, які допоможуть при подальшому написанні коду зрозуміти алгоритм роботи програми, а також створити своєрідний каркас для цієї програми.

Для початку буде створено діаграму послідовностей, яка допоможе впорядкувати об'єкти та їх взаємодію за часом. На діаграмі послідовностей показують повідомлення, які передаються між наявними у програмному забезпеченні об'єктами.

Створення UML діаграм допоможе краще зрозуміти логіку програмного забезпечення, як воно повинно працювати, показати усі його залежні компоненти та об'єкти, їх взаємодію та функціонал програмного забезпечення.

Об'єктами, що взаємодіють між собою за допомогою повідомлень є користувач, програма та сервер.

На рисунку 2.2 зображена діаграма послідовностей для програмного забезпечення, що розробляється у рамках дипломного проекту.

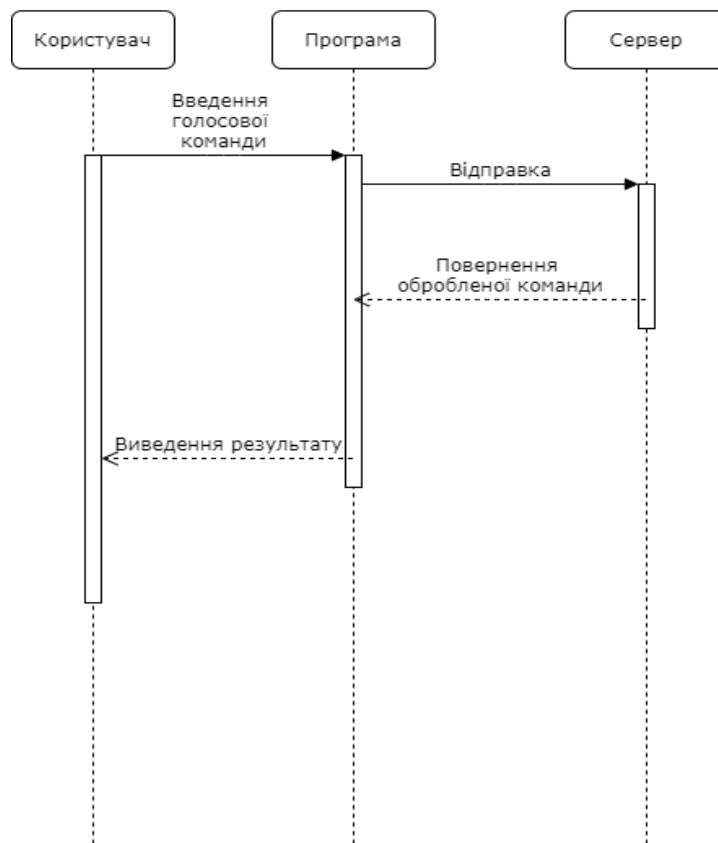


Рисунок 2.2 – Діаграма послідовностей

Діаграма діяльності представляється у вигляді графу або блок-схеми. Діаграма діяльності за графічною нотацією є дуже подібна на діаграму станів. Діаграму діяльності використовують при створенні послідовності дій.

Правильна робота програмного забезпечення залежить від наявності інтернету, мікрофону та голосового пакету, тому при побудові діаграми буде враховано їх наявність, а також їх стабільну роботу в вигляді умов, при дотриманні яких програма продовжує функціонувати і продовжує виконувати наступну дію, але при порушенні хоча б однієї з них та завершує роботу програми або лише виводить повідомлення про помилку, якщо цю помилку можна вирішити не вимикаючи програму і якщо програма не буде працювати в фоновому режимі, споживаючи зайві ресурси пристрою.

На рисунку 2.3 показана діаграма діяльності для програмного забезпечення.

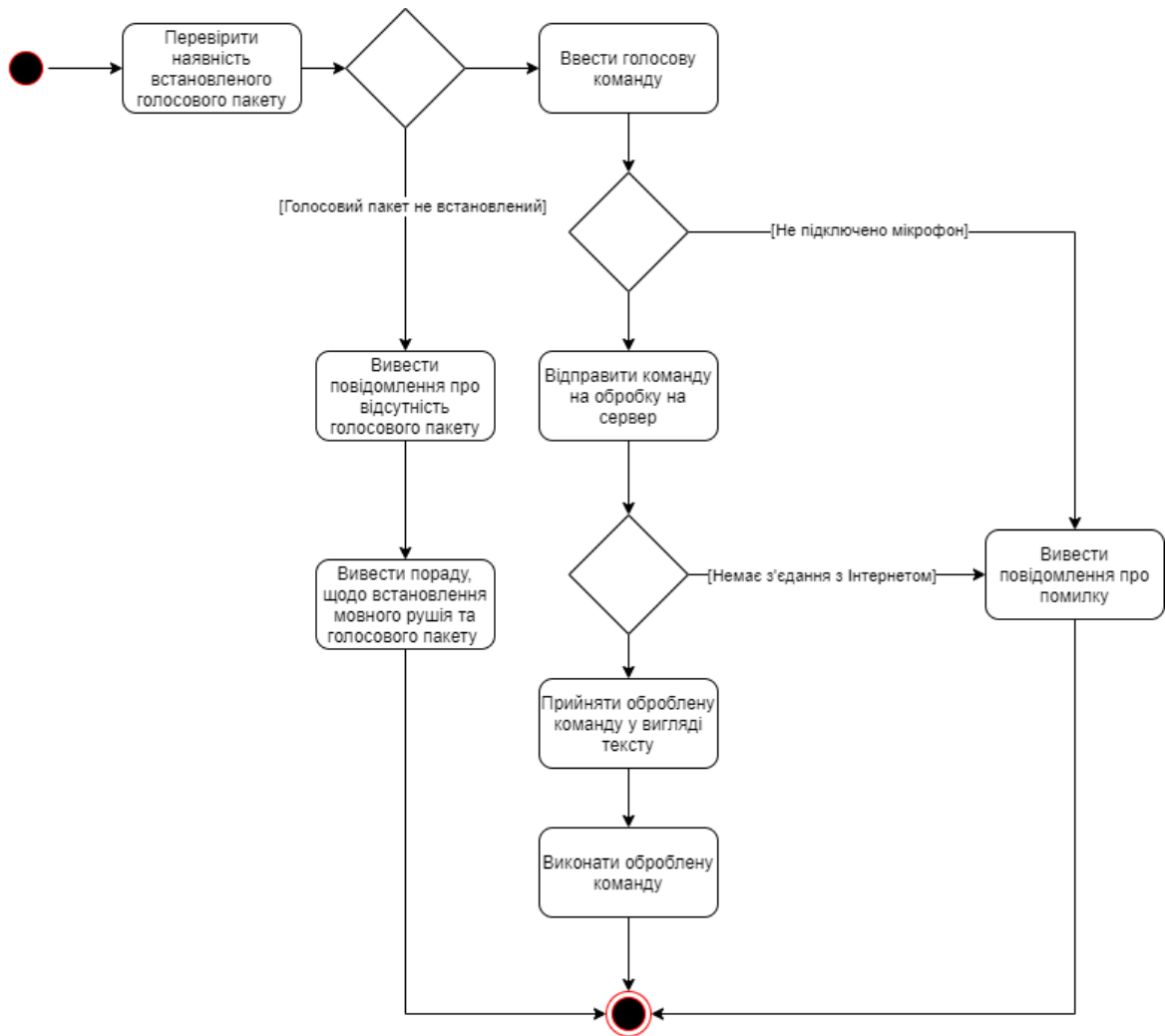


Рисунок 2.3 – Діаграма діяльності

Наступним кроком для побудови логіки програмного забезпечення є створення діаграми станів. Як вже було сказано вище, діаграма станів є дуже схожою за графічною нотацією на діаграму діяльності, але, на відміну від діаграми діяльності діаграма станів показуватиме зміну станів об'єктів програми.

Також дуже часто діаграму станів розробляють для опису поведінки об'єктів у програмі, а також вона може використовуватись, як функціональна специфікація варіантів використання.

На рисунку 2.4 показано діаграму станів, яку було створено для даного програмного продукту, що розробляється.

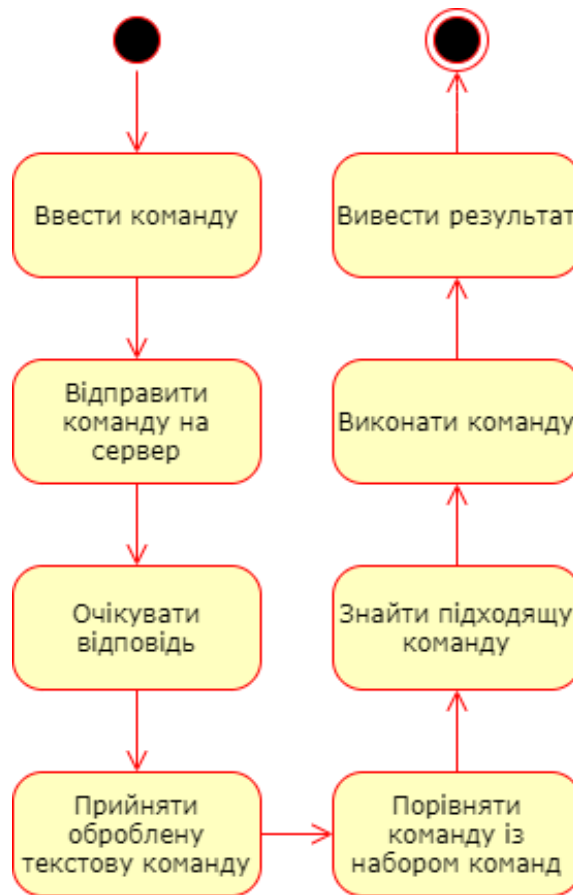


Рисунок 2.4 – Діаграма станів

2.2 Обґрунтування вибору технологій для розробки

2.2.1 Вибір мови програмування для розробки

Основними критеріями для вибору мови були: можливість взаємодіяти з комп'ютером та його комплектуючими та можливість підключення сторонніх мовних рушіїв.

Для даного проекту було обрано Python 3.8.6, яка цілком задовольняє критерії розробки дипломного проекту.

Python працює на різних платформах (Windows, Linux, MacOS і т.д.). Він має зручний синтаксис, що дозволяє розробникам скоротити кількість рядків для написання певних програм. Також Python працює на системі інтерпретатора. Це означає, що код може бути запущений відразу після його написання.

Остання версія Python – це Python 3, яка містить безліч нововведень:

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

2.2.2 Вибір мовного рушія

Синтез мовлення – це штучне створення людської мови. Для цих цілей використовують синтезатором мовлення і застосовують для розробки програмного забезпечення.

Голоси створюються за допомогою натурального мовлення. Це дозволяє їм використовувати менше місця, адже на комп'ютері зберігаються лише їх статистичні моделі.

Мовний рушій повинен задовольняти такі критерії:

- наявність української мови;
- висока якість голосових пакетів;
- безкоштовний продукт.

Для вибору мовного рушія для розробки програмного забезпечення були такі варіанти, як:

а) Microsoft TTS

Переваги:

- 1) підтримка API Microsoft SAPI 5.x;
- 2) є вбудованим у Windows.

Недоліки:

- 1) відсутня українська мова.

б) Sacrament TTS

Переваги:

- 1) підтримка API Microsoft SAPI 5.x.

Недоліки:

- 2) відсутність бібліотеки для Python;
- 3) відсутність української мови;
- 4) платний програмний продукт;
- 5) можливість виникнення труднощів при використанні під час реалізації коду програмного забезпечення.

					ДППЗ.170115.01.14.ПЗ	Арк.
						26
Зм.	Арк	№ докум.	Підпис	Дата		

в) Google TTS

Переваги:

- 1) підтримка української мови;
- 2) якісні голосові пакети;
- 3) наявність бібліотеки для Python.

Недоліки:

- 1) працює за допомогою збереження файлів на диску.

г) RNVoice

Переваги:

- 1) наявність декількох пакетів української мови (Anatol та Natalia);
- 2) висока якість голосу;
- 3) наявність бібліотеки для Python.

Недоліки:

- 1) можливість неправильної вимови іншомовних слів;

RNVoice найбільше задовольняє критерії мовного рушія, тому для перетворення тексту в мову було вибрано саме цей рушій.

RNVoice використовує статистичний параметричний синтез. Він спирається на існуючі відкриті енергетичні технології (наприклад HTS та подібне програмне забезпечення).

На даний момент існує 4 мовні пакети, які використовуються для синтезу української мови:

- CyberMova/VymovaPlus/VymovaPro (голоси Опанас, Дмитро, Слава та Наталка). Основним недоліком є те, що даний програмний продукт є платним;
- UkrVox (голос Ігор Мурашко);
- Розмовлялька (голоси Анатоль/Стьопа та Руся);
- RNVoice (голоси Анатоль та Наталка).

Для розробки програмного проекту було вибрано голос Anatol від компанії RNVoice. Даний голос з'явився у 2016 році завдяки диктору харківського суспільного радіо «UA:Українське радіо» Подорожка Анатолія, голос якого

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		27

використовується у мовному пакеті Anatol та Харківського соціально-реабілітаційний центр незрячих, яким вдалося залучити його до цього.

2.2.3 Вибір бібліотек для розробки програмного забезпечення

Для розробки системи буде використано декілька основних бібліотек Python, а саме PyAudio, PyTTSx3, SpeechRecognition, fuzzywuzzy.

PyAudio забезпечує прив'язування Python для PortAudio, крос-платформенної аудіобібліотеки вводу-виводу. Використання PyAudio допоможе легко використовувати Python для відтворення та запису аудіо на різних платформах та системах.

Для перетворення тексту в мову використовують бібліотеки PyTTSx3, ru-espeak-ng або Pocketsphinx.

PyTTSx3 — це зручна кросплатформенна бібліотека для реалізації TTS на Python 3. Використовує різні системи синтезу мови в залежності від поточної ОС:

- в Windows — SAPI5;
- в Linux — eSpeak;
- в Mac OS X — nsss.

Py-espeak-ng дуже часто служить альтернативою для PyTTSx3, у разі необхідності користування синтезатором eSpeak NG. На відміну від PyTTSx3, ru-espeak-ng використовує `sync=True`, замість `runAndWait()`, без якого він почне читати всі фрази одночасно. Одним із основних мінусів – відсутність української мови.

У даному випадку при розробці програмного забезпечення буде доцільніше використати `pyttsx3`.

Для коректної роботи PyTTSx3 необхідно встановити `pywin32`, який служитиме для взаємодії з об'єктами компонентів Windows.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		28

Для розпізнавання мовлення (STT) буде використовуватись бібліотека SpeechRecognition від компанії Google. Бібліотека працює онлайн, але завдяки цьому споживає невелику кількість ресурсів пристрою.

Основним аналогом Google Speech Recognition є CMU Sphinx Voice Recognition, але ця бібліотека для розпізнавання голосу працює оффлайн і для цього використовує значну кількість ресурсів.

Також для нечіткого зрівняння мовлення буде використано модуль FuzzyWuzzy - бібліотека Python, яка використовується для узгодження рядків. Нечітке порівняння рядків - це процес пошуку рядків, які відповідають заданому шаблону, у цьому випадку, введеній користувачем команді.

2.2.4 Вибір засобів для розробки програмного забезпечення

Інтегровані середовища розробки, тобто IDE, використовують для написання програмного коду. IDE використовуються для написання початкового коду для програмного продукту та дають змогу полегшити написання цього коду за допомогою розширеного функціоналу, наприклад, підказки щодо доповнення програмного коду.

Також IDE можуть бути багатомовними, а можуть використовуватись лише для одної мови програмування.

Серед варіантів для розробки програмного забезпечення є такі середовища:

- Eclipse з розширенням PyDev;
- Microsoft Visual Studio;
- Atom;
- SublimeText;
- PyCharm;
- IntelliJ IDEA.

					ДППЗ.170115.01.14.ПЗ	Арк.
						29
Зм.	Арк	№ докум.	Підпис	Дата		

Найбільш підходящим є інтегроване середовище розробки PyCharm, адже він не потребує додаткових розширень для написання коду на мові програмування Python, а також має інтуїтивно зрозумілий та зручний інтерфейс для розробки.

PyCharm є крос-платформним, з версіями для Windows, macOS та Linux. Видання Community випускається під ліцензією Apache. Існує також Professional Edition з додатковими функціями, яка випускається під власною ліцензією.

Для розробки програмного забезпечення буде використовуватись PyCharm Community Edition.

PyCharm має зручний редактор коду з усіма корисними функціями: підсвічуванням синтаксису, автоматичним форматуванням та доповненням коду.

PyCharm дозволяє легко переглядати документацію прямо у вікні редактора або в браузері.

Програма підтримує всі свіжі версії фреймворку для розробки веб-додатків Django, а також Cython, IronPython, Jython та багато інших інструментів.

PyCharm дозволяє швидко робити рефакторинг коду, а також використовувати зручний графічний налагоджувач.

Також PyCharm дає змогу проводити інтегроване Unit тестування для програмного забезпечення.

PyCharm має велику колекцію плагінів, і його можна використовувати в зв'язці з різними трекерами, наприклад, JIRA, Redmine, Youtrack, Trac і так далі.

Ці трекери допомагають відстежувати різні помилки, що з'являються у процесі написання та відлагодження коду.

Також PyCharm підтримує версії Python 2.x та 3.x.

Основні можливості PyCharm:

- підсвічування синтаксису і помилок коду;
- навігація серед проектів та коду: відображення файлової ієрархії проекту, швидкий перехід між файлами, класами і методами;
- рефакторинг коду;
- вбудований відлагоджувач для Python;

					ДППЗ.170115.01.14.ПЗ	Арк.
						30
Зм.	Арк	№ докум.	Підпис	Дата		

– підтримка систем контролю версій.

На рисунку 2.1 зображено інтерфейс програми PyCharm.

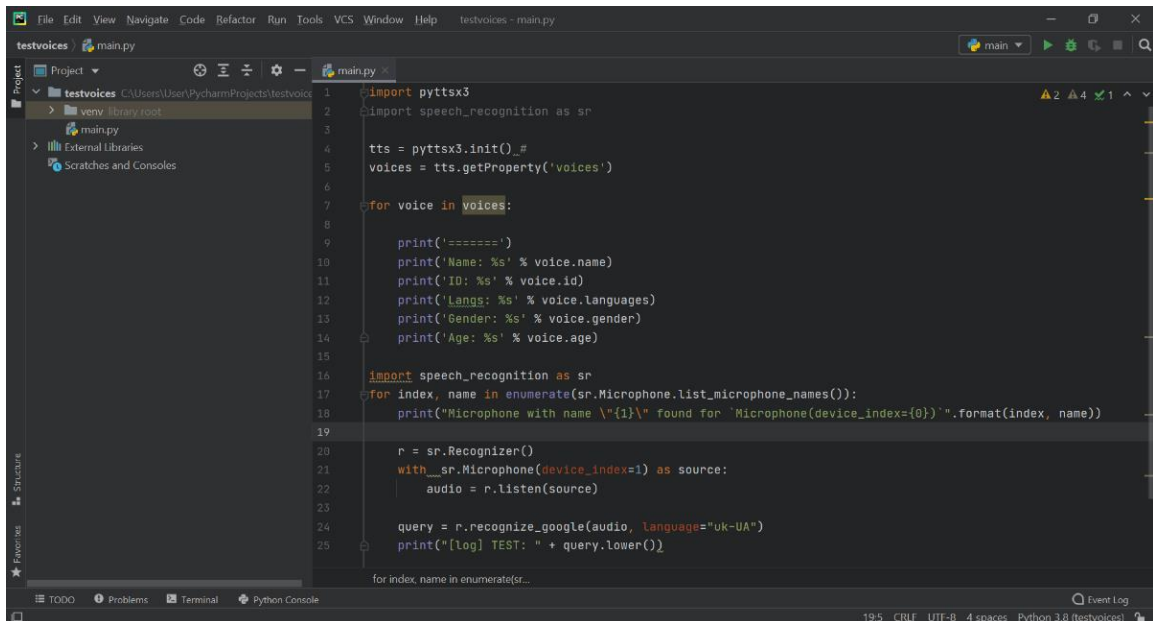


Рисунок 2.5 – Інтерфейс IDE PyCharm

2.3 Реалізація голосового інтерфейсу програми

Після вибору технологій для розробки програмного забезпечення потрібно визначити функціонал програми, тобто визначити найбільш потрібні для користувача команди та їх назви. Тому наступним кроком буде детальний опис функціональних команд для голосового помічника (таблиця 2.1).

Таблиця 2.1 – Опис голосових команд

Назва команди	Опис
1	2
ctime	Показуватиме поточний час

Продовження таблиці 2.1

1	2
browser	Запускатиме браузер, який користувач призначив у системі за замовчування, у разі відсутності операційна система запитає, який браузер потрібно запустити
shutdown	Вимикатиме персональний комп'ютер із затримкою (наприклад 5 або 10 секунд). Відсутність затримки може призвести до втрати незбережених даних
restart	Перезавантажує персональний комп'ютер із певною затримкою. Відсутність затримки може призвести до втрати незбережених даних
logoff	Виконує вихід із облікового запису, яким користувач користується
ping	Допоможе провести невелике тестування програмного забезпечення
sysinfo	Виводитиме детальну інформацію про апаратну та програмну частину (операційну систему, процесор, тип шини, архітектуру і т.д.)
meminfo	Виводитиме детальну інформацію про загальну, зайняту та вільну пам'ять, а також про оперативну пам'ять
location	Виводитиме інформацію про місцезнаходження користувача та його провайдера використовуючи інтернет-сервіси

Кінець таблиці 2.1

1	2
weather	Виводитиме на екран детальну інформацію про погоду (про температуру повітря, вологість та тиск і т.д.), використовуючи ресурс OpenWeatherMap та інші ресурси для визначення місцезнаходження користувача
cam	Запускатиме вбудовану у пристрій веб-камеру. Також буде реалізовано можливість збереження знімку та вимкнення веб-камери за допомогою натиснення певних клавіш.
quit	Завершуватиме роботу програмного забезпечення, закриваючи вікно і вимикатиме доступу до мікрофону

Отже, у результаті проектування програмного додатку було визначено поставлено основні задачі для розробки програмного забезпечення. Також було побудовано діаграму послідовності та діаграму діяльності для кращого розуміння логіки програмного продукту. Після цього було вибрано необхідні для розробки технології, а саме вибрано найбільш оптимальну мову для написання коду, вибрано бібліотеки для розробки, а також було вибрано середовище для розробки програмного забезпечення. Після вибору технологій було визначено основні команди для голосового помічника.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Детальне проектування даних

Для правильного функціонування голосового помічника, замість функції input(), яка відповідає за введення тексту з клавіатури, буде реалізовано введення даних за допомогою голосового введення.

Все, що потрібно користувачу для правильної роботи програмного забезпечення – це мікрофон із гарним подавленням шумів. Це може бути будь-який мікрофон, наприклад, петличний, динамічний чи конденсаторний.

При розробці програмного продукту у якості мікрофона буде використовуватись Android-пристрій. Основною відмінністю використання смартфона в якості мікрофона від вбудованого мікрофона є краща якість подавлення шуму, що допоможе при реалізації розпізнавання мовлення.

Також використання смартфона у якості мікрофона може бути корисним, якщо зовнішній та внутрішні мікрофони відсутні. Наприклад при написанні коду за допомогою настільного комп'ютера чи ноутбука, вбудований мікрофон якого не працює або працює недостатньо якісно.

Для таких цілей може використовуватись декілька програмних додатків, таких як DroidCam та Wo Mic. DroidCam можна використовувати у якості мікрофону та веб-камери, але через те, що він оптимізований під використання пристрою у якості камери, то буде доцільніше використовувати програму Wo Mic.

Для розробки програмного забезпечення буде використано транспортуванням даних по Wi-Fi. У разі відсутності безпроводних мереж можливе транспортування даних по інших джерелах, таких як Bluetooth або USB. У даному методі смартфон працює як сервер або точка доступу, тоді як комп'ютер працює у якості клієнта. Для з'єднання також можуть використовуватись пристрої під управлінням операційної системи iOS.

Для правильної роботи програмного забезпечення було встановлено значення 8125 для порту управління (TCP), та значення 49152 для медіа порту

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		34

(UDP). Значення для медіа порту встановлюється лише на пристрої-клієнті, у даному випадку на ноутбучі і синхронізується на смартфоні після з'єднання.

Для розміру буферу було встановлено значення 3, що допоможе зменшити затримку звуку та покращити його точність.

Також краще використовувати USB-з'єднання між пристроями, що забезпечить меншу затримку, а також забезпечить менше втрат даних.

Після того, як програма отримала голосову команду, вона переводить її у нижній регістр та відправляє на сервера Google для перетворення в текст. Також в якості аргументу вказується те, що голосовий запит буде перетворюватись в текст, як україномовний.

У лістингу нижче наведено фрагмент коду, який відповідає за обробку та відправлення голосового запиту та отримання результату у вигляді тексту:

```
voice = recognizer.recognize_google(audio, language="uk-UA").lower()
print("[log] Розпізнано: " + voice)
```

3.2 Реалізація модулів системи

3.2.1 Оператор import та функція def()

Під час розробки програмного забезпечення були підключенні бібліотеки, які були описані у попередньому розділі. Для підключення потрібних бібліотек була використана команда import. Імпорт дозволяє файлу Python або модулю Python отримати доступ до певного сценарію з іншого файлу або модуля Python. Це допомагає використовувати лише функції та властивості, до яких має доступ програма. Модуль import при використанні шукає модуль спочатку в певній області, викликаючи функцію `__import__()`. Потім значення, повернені функцією, відображаються у вихідних даних вихідного коду.

Конструкція `import ... as ...` дозволяє використовувати власні назви для бібліотек, якщо, наприклад, назва бібліотеки занадто довга. У наступному

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		35

фрагменті коду наведено приклад використанням власної назви при імпорті бібліотек Python:

```
import speech_recognition as sr
import webbrowser as wb
```

Функцією називають певний блок коду, який запускається тільки тоді, коли є потреба у його виклику.

Функції допомагають розділити програму на менші частини. У міру того як збільшується кількість програмного коду, функції допомагають зробити програму більш організованою.

Крім того функції дозволяють уникнути всіх непотрібних повторень у коді.

Для передавання інформації функціям використовують аргументи. Аргументи наводяться після назви функції, у круглих дужках.

При використанні функції можна використовувати декілька аргументів. Для цього їх перераховують, розділяючи комою.

Для розробки програмного забезпечення буде створено чотири функції: `callback`, `recognize_cmd`, `execute_cmd`, які будуть взаємодіяти між собою та виконувати основні функції, такі як розпізнавання мовлення, знаходження команди та виконання, та функція `speak`, за допомогою якої буде ініціалізовано мовний рушій.

Наступний приклад має функцію з одним аргументом (`sometext`). Коли функція викликана, ми передаємо ім'я, яке використовується всередині функції для друку повного імені:

```
def speak(sometext):
    print(sometext)
    voices = speak_engine.getProperty('voices') #отримує список
встановлених голосів
    for voice in voices: #вибір голосу за замовчуванням
        speak_engine.setProperty('voice', voice.id)
    speak_engine.say(sometext)
    speak_engine.runAndWait()
    speak_engine.stop()
```

					ДППЗ.170115.01.14.ПЗ	Арк.
						36
Зм.	Арк	№ докум.	Підпис	Дата		

3.2.2 Словники

Для реалізації звернення до голосового помічника та переліку команд, які він буде виконувати, буде використано словник. Дана структура даних, також відома як асоціативний масив. За допомогою словників можна зберігати та використовувати дані, як ключ та його значення.

У мові програмування Python словник може бути створений завдяки розміщенню послідовності елементів у дужках {}, розділених «комою». Словник містить пару значень, одним із яких є ключ, а іншим відповідним парним елементом є значення цього ключа. Значення у словнику можуть мати будь-який тип даних і можуть дублюватися, тоді як ключі не повинні повторюватись і мають бути незмінними.

У лістингу показано використання словника для реалізації команд для голосового помічника та різних варіантів для їх вимови, а також різних варіантів імені асистента:

```
param = {
    "name": ('алекс', 'бот', 'робот'),
    "cmds": {
        "ctime": ('скільки годин', 'котра година', 'яка година'),
        "browser": ('відкрий браузер', 'запусти браузер'),
        "shutdown": ('вимкни комп', 'вимкни пк', 'виключи комп', 'включи пк'),
        "restart": ('перезавантаж комп', 'перезагрузи пк', 'перезагрузи комп', 'перезавантаж пк'),
        "logoff": ('вийди з облікового запису', 'вийди з системи'),
        "ping": ('тест', 'пінг', 'відкрий тест'),
        "sysinfo": ('інформація про систему', 'характеристики пристрою', 'характеристики'),
        "meminfo": ('інформація про пам'ять', 'кількість вільної пам'яті'),
        "location": ('де я', 'моє розташування'),
        "weather": ('погода', 'яка зараз погода', 'погода зараз'),
        "cam": ('камера', 'зроби фото', 'ввімкни камеру'),
        "quit": ('бувай', 'пока', 'дякую', 'досить', 'стоп')
    }
}
```

									Арк.
									37
Зм.	Арк	№ докум.	Підпис	Дата	ДППЗ.170115.01.14.ПЗ				

3.2.3 Бібліотека fuzzywuzzy

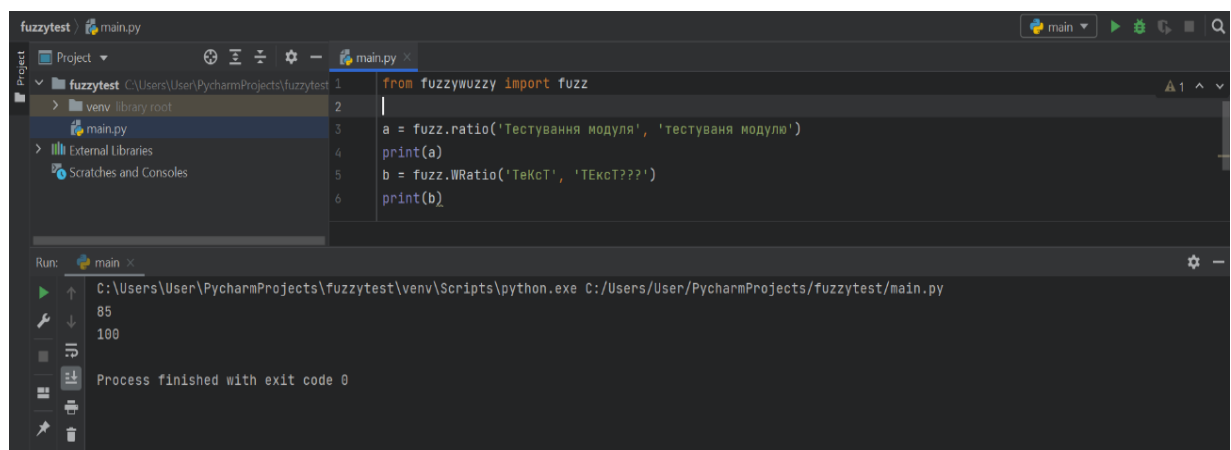
Також треба врахувати, що SpeechRecognition може іноді здійснювати похибки у розпізнаванні мовлення. Найчастіше це може відбуватись через сторонній шум. Тому, щоб уникнути цього буде використано порівняння рядків. У Python існує декілька способів для реалізації цієї ідеї:

- просте порівняння;
- використання regex;
- використання сторонніх бібліотек.

Найбільш підходящим методом буде використання бібліотеки fuzzywuzzy, який працює за принципом зрівнювання рядків, даючи індекс подібності. Якщо цей індекс дорівнює 100, то це свідчить про те, що рядки рівні.

FuzzyWuzzy – це бібліотека Python, яка використовується для порівняння рядків. Нечітке порівняння рядків дозволяє виконати пошук рядків, які відповідають заданому шаблону. В основному він використовує відстань Левенштейна для обчислення різниці між послідовностями.

На рисунку 3.1 показана невелика програма, яка показує роботу бібліотеки. WRatio, на відміну від ratio, дозволяє порівняти два рядки, з урахуванням літер верхнього регістру та нижнього, а також деяких інших правил.



```
1 from fuzzywuzzy import fuzz
2
3 a = fuzz.ratio('Тестування модуля', 'тестування модуль')
4 print(a)
5 b = fuzz.WRatio('ТеКсТ', 'ТЕКсТ??')
6 print(b)
```

Run: main ×
C:\Users\User\PycharmProjects\fuzzytest\venv\Scripts\python.exe C:/Users/User/PycharmProjects/fuzzytest/main.py
85
100
Process finished with exit code 0

Рисунок 3.1 – Приклад роботи бібліотеки fuzzywuzzy

Для реалізації алгоритму при розробці програмного забезпечення було створено окрему функцію, яка допоможе порівняти введenu команду зі командами зі словника та вибрати ту, команду, відсоток схожості якої найбільший.

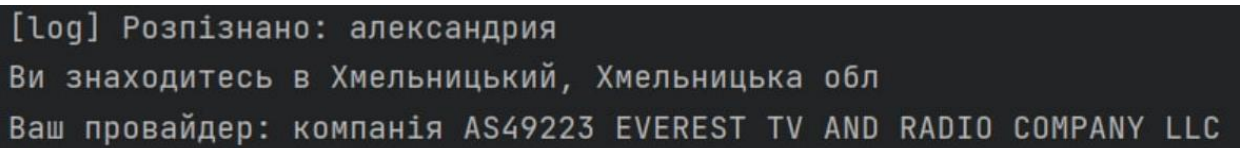
При реалізації алгоритму буде використано `ratio`, адже команди переводяться у нижній регістр, тому нема потреби враховувати верхній.

Реалізацію алгоритму порівняння введеної команди із готовими показано у наступному фрагменті коду:

```
def recognize_cmd(cmd):
    RC = {'cmd': '', 'percent': 0}
    for c, v in param['cmds'].items():

        for x in v:
            vrt = fuzz.ratio(cmd, x)
            if vrt > RC['percent']:
                RC['cmd'] = c
                RC['percent'] = vrt
```

На рисунку 3.2 показано результат роботи бібліотеки `fuzzywuzzy`, отримавши найбільший відсоток схожості з командою «Алекс, де я?».



```
[log] Розпізнано: александрія
Ви знаходитесь в Хмельницький, Хмельницька обл
Ваш провайдер: компанія AS49223 EVEREST TV AND RADIO COMPANY LLC
```

Рисунок 3.2 – Результат роботи алгоритму

3.2.4 Модуль platform

Дуже часто у користувача з'являється потреба дізнатись інформацію про певні характеристики персонального комп'ютера. Дуже часто для цього використовують стороннє програмне забезпечення, тому було прийнято рішення

реалізувати виведення інформації про персональний комп'ютер, на якому працює програмне забезпечення.

Для цих вирішення цієї потреби існує безліч варіантів, але найбільш підходящим у даному випадку є використання модуля platform. Даний модуль використовують для виведення загальних відомостей про апаратну частину персонального комп'ютера. До таких відомостей належить інформація про реальну назву процесора, загальні відомості про операційну систему, яка встановлена на пристрої, інформація про архітектуру комп'ютера, версія інтерпретатора, на якому працює програмне забезпечення.

Нижче наведено фрагмент коду команди, яка буде виводити інформацію про процесор, тип машини, платформу та операційну систему користувача:

```
elif cmd == 'sysinfo':
    m = platform.machine()
    print("Тип машини: " + m)
    ver = platform.version()
    print("Версія: " + ver)
    plt = platform.platform()
    print("Платформа: " + plt)
    sys = platform.system()
    print("Система: " + sys)
    pr = platform.processor()
    print("Процесор: " + pr)
    arc = platform.architecture()
    print("Архітектура: ")
    print(arc)
    pv = platform.python_version()
    print("Версія Python: " + pv)
    rel = platform.release()
    print("Версія Windows: " + rel)
    speak("Ось ваша детальна інформація про систему.")
```

3.2.5 Модулі psutil та shutil

Вслід за реалізацією команди для виведення інформації про персональний комп'ютера буде реалізовано команду для виведення інформації про постійну та оперативну пам'ять. Найбільш підходящим для цих цілей є модуль psutil. Даний

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

модуль дозволяє виводити інформацію про запущені процеси, а також виводити інформацію про процесор чи про використання пам'яті.

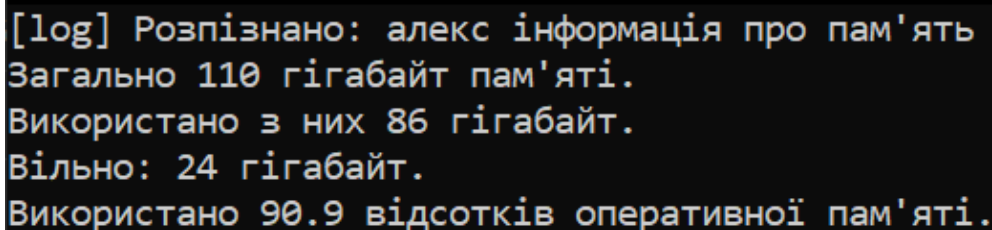
Разом з psutil у команді буде використано модуль shutil. Даний модуль працює з файлами та файловими системами, вмiє копіювати вміст файлу та його метадані та управляти директоріями. Також даний модуль може виводити статистику про використання пам'яті у байтах за вказаним шляхом.

Так як статистика використання пам'яті повертається у байтах, то буде правильно перевести ці дані у гігабайти для більш зручного використання команди користувачем.

У лістингу коду показано фрагмент використання модулів psutil та shutilЮ для виведення інформації про оперативну та постійну пам'ять:

```
elif cmd == 'meminfo':
    psutil.cpu_percent() #статистика використання процесора у
    відсотках
    psutil.virtual_memory() #статистика використання системної
    пам'яті
    dict(psutil.virtual_memory()._asdict())
    used_ram = psutil.virtual_memory().percent
    total, used, free = shutil.disk_usage("\\") #статистика
    використання дискового простору за вказаним шляхом
    speak("Загально %d гігабайт пам'яті.\n" % (total // (2 ** 30))
    +
        "Використано з них %d гігабайт.\n" % (used // (2 ** 30))
    +
        "Вільно: %d гігабайт.\n" % (free // (2 ** 30)) +
        "Використано " + str(used_ram) + " відсотків оперативної
    пам'яті.")
```

На рисунку 3.3 показано результат роботи модулів psutil та shutil.



```
[log] Розпізнано: алекс інформація про пам'ять
Загально 110 гігабайт пам'яті.
Використано з них 86 гігабайт.
Вільно: 24 гігабайт.
Використано 90.9 відсотків оперативної пам'яті.
```

Рисунок 3.3 – Результат роботи модулів psutil та shutil

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		41

3.2.6 JSON

JSON забезпечує обмін даних між пристроями у вигляді тексту. Ця нотація є структурованою та легко може прочитатись людиною.

JSON з'явився на початку 2000-х років через потребу передачі текстової інформації без допомоги різних розширень для браузерів.

JSON може бути побудований на різних структурах. Прикладом таких структур є словник, хеш-таблиця, структура, тобто ті, які містять ключ-значення. Також це може бути впорядкований набір значень, таких як вектор, масив або список.

На рисунку 3.4 зображено сайт OpenWeatherMap, який дозволяє при наявності ключа API використовувати дані про погоду у програмному коді (для десктопних додатків, мобільних, Телеграм-ботів і т.д.) у режимі реального часу.

```
{ "message": "like", "cod": "200", "count": 1, "list":  
  [ { "id": 706369, "name": "Khmelnyskyi", "coord": { "lat": 49.4216, "lon": 26.9965 }, "main":  
    { "temp": 289.04, "feels_like": 288.33, "temp_min": 289.04, "temp_max": 289.04, "pressure": 1015, "humidity": 63, "sea_level": 1015, "grnd_level": 982 }, "dt": 1621667352, "wind":  
    { "speed": 5.64, "deg": 182 }, "sys": { "country": "UA" }, "rain": null, "snow": null, "clouds":  
    { "all": 6 }, "weather": [ { "id": 800, "main": "Clear", "description": "clear sky", "icon": "01d" } ] ] }
```

Рисунок 3.4 – Вигляд сайту OpenWeatherMap при наявності API-ключа

Для звичайних користувачів OpenWeatherMap буде виглядати, як звичайний сайт. Він відображатиме усю інформацію про погоду у зрозумілому для користувача вигляді. Доступ до JSON-даних здійснюється тільки після реєстрації користувача та отримання його персонального ключа API, який буде дійсний, поки існує профіль на сайті, до якого прив'язаний цей ключ.

Звичайний вигляд сайту OpenWeatherMap зображено на рисунку 3.5.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		42


```
[log] Розпізнано: алекс де я
Ви знаходитесь в Хмельницький, Хмельницька обл
Ваш провайдер: компанія AS49223 EVEREST TV AND RADIO COMPANY LLC
```

Рисунок 3.6 – Результат використання JSON

3.2.5 OpenCV

У програмному забезпеченні буде реалізовано команду для доступу до веб-камери пристрою. Для цього буде використано OpenCV. OpenCV – це бібліотека, яка використовується для обробки зображень та відео з файлів та у режимі реального часу, використовуючи вбудовану чи зовнішню веб-камери. OpenCV використовує величезну кількість алгоритмів і може використовуватись для будь-яких цілей, наприклад розпізнавання різних об'єктів на фото, таких як люди, машини, дерева або навіть дорожня розмітка.

Початковий код OpenCV був створений компанією Intel за допомогою мови програмування C++ та може використовуватись у різних мовах програмування, наприклад, у Python, Java, Ruby і т.д.

У коді нижче показано команду, метою якої буде запуск веб-камери та збереження зображення з неї:

```
elif cmd == 'cam':
    cap = cv2.VideoCapture(0, cv2.CAP_DSHOW) #Підключення вбудованої веб-камери
    speak("Для збереження змінку натисніть на клавішу пробіл, для виходу з камер - утримуйте клавішу ESC")
    while True:
        count = 0
        ret, img = cap.read() #зчитування кадрів з веб-камери
        cv2.imshow("XYREXX", img) #виведення зображення у вікні
        if cv2.waitKey(10) == 32:
            cv2.imwrite("frame%d.jpg" % ret, img) #запис кадру
            count += 1
        if cv2.waitKey(10) == 27: #Вимкнення на ESC
            break
    cap.release() #вимкнення камери
    cv2.destroyAllWindows()
```

					ДППЗ.170115.01.14.ПЗ	Арк.
						44
Зм.	Арк	№ докум.	Підпис	Дата		

Результат виведення зображення у окремому вікні показано на рисунку 3.7.

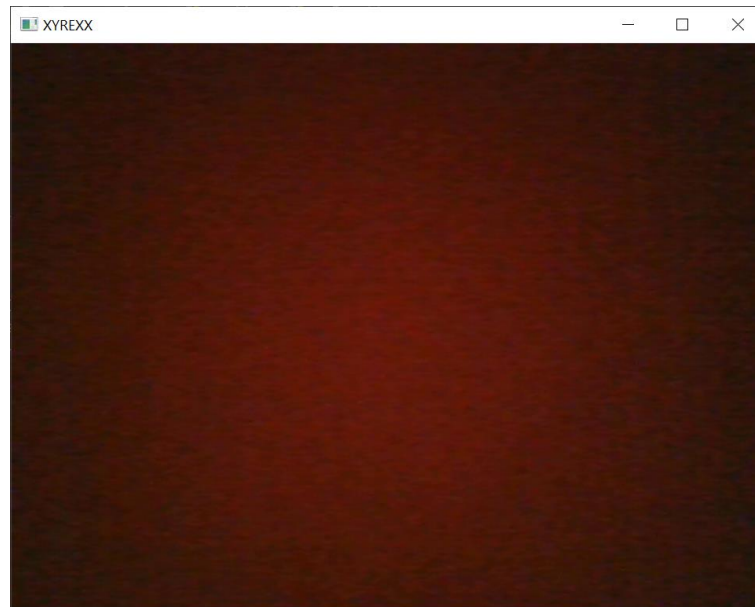


Рисунок 3.7 – Вікно із зображенням з веб-камери

3.3. Компіляція проекту

Основним недоліком проекту на даний момент є залежність від середовища програмування і неможливість автономного функціонування через неможливість використання модулів. Для вирішення цієї проблеми проект буде компільовано у виконавчий файл з можливістю імпортування модулів. Здійснити це можна за допомогою таких засобів:

- PyInstaller;
- py2exe;
- cx_Freeze.

Особливих відмінностей у використанні цих засобів немає, тому було обрано PyInstaller. У разі виникнення помилок можна буде використати аналоги.

На рисунку 3.5 показано процес установки PyInstaller та його залежностей у терміналі графічного середовища PyCharm.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		45

Після завершення компіляції у папці проекту з'явилась папка dist з папкою з готовим програмним продуктом, вміст якої зображено на рисунку 3.7.

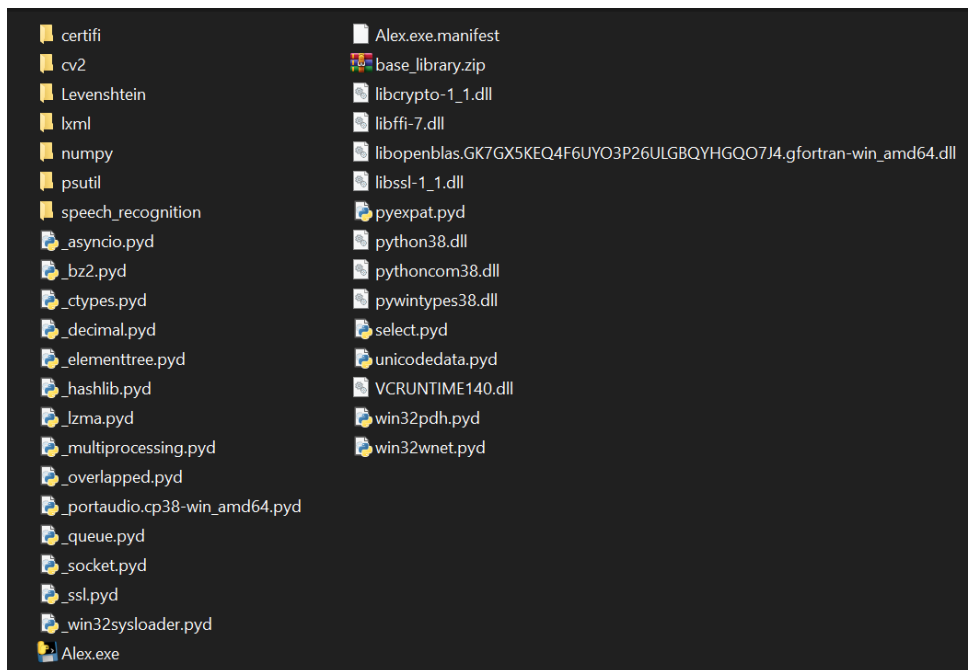


Рисунок 3.7 – Результат компіляції проекту

У разі проблем із запуском програми можна перевірити запуск через консольний рядок Windows, який допоможе вивести проблему, якщо запуск виконавчого файлу переривається без можливості показу помилки.

На рисунку 3.8 показано одну із таких проблем, яка трапилась через відсутність при компіляції аргументу для імпорту однієї з бібліотек другого рівня.

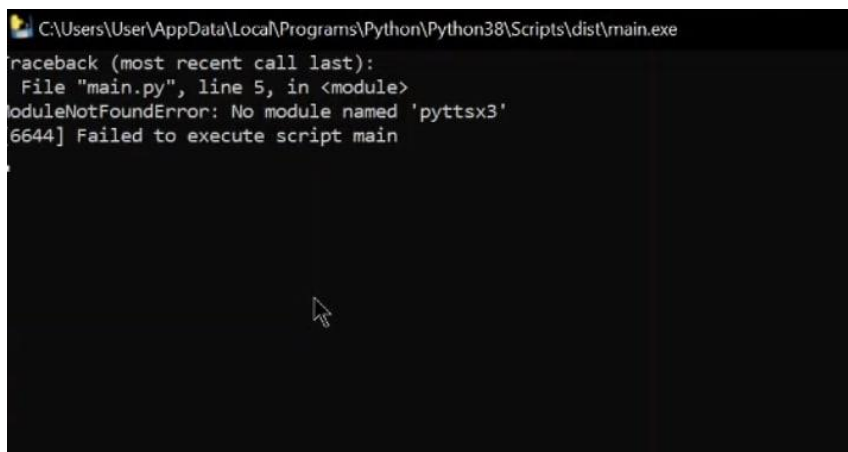


Рисунок 3.8 – Помилка після компіляції

3.3 Інтерфейс користувача

Інтерфейс користувача використовують для взаємодії звичайного користувача з програмним забезпеченням. Інтерфейси користувача використовують для зручності використання програмного продукту.

Серед можливих варіантів інтерфейсу користувача для розробки програмного продукту є GUI – графічний інтерфейс, але даний вид інтерфейсу потребує більше ресурсів комп'ютера, CLI, тобто інтерфейс командного рядка, який використовує консольне вікно, а також ZUI, в якому використовується дуже великий простір, а більшість кнопок стаються доступними при масштабуванні.

Так як актуальність програмного забезпечення полягає у створенні програмного продукту, що містить мінімалістичний дизайн, масштабований інтерфейс користувача не задовольняє умов для розробки програмного продукту, а при використанні графічного інтерфейсу програма споживатиме більше постійної та оперативної пам'яті, що може вплинути на багатозадачність при використанні інших програм у фоновому режимі, тому найбільш підходящим та найменш ресурсовитратним є інтерфейс командного рядка, тобто CLI, який використовує мінімальну кількість пам'яті у порівнянні з графічним інтерфейсом користувача. У якості введення команд було використано голосове введення, яке цілком може замінити введення з клавіатури при використанні сторонніх та вбудованих модулів і бібліотек.

Інтерфейс командного рядка забезпечує введення команди програмному забезпеченню за допомогою клавіатури. Результатом роботи програми, яка містить інтерфейс командного рядка може бути певний текст або навіть графічне чи звукове виведення інформації.

На рисунку 3.5 зображено інтерфейс командного рядка для програмного забезпечення для голосового управління процесами комп'ютера.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

```

C:\Users\User\PycharmProjects\DP_assisv2\dist\Alex\Alex.exe
[log] Розпізнано: алекс інформація про пам'ять
Загально 110 гігабайт пам'яті.
Використано з них 86 гігабайт.
Вільно: 24 гігабайт.
Використано 90.9 відсотків оперативної пам'яті.
[log] Розпізнано: алекс інформ я про систему
Тип машини: AMD64
Версія: 10.0.19041
Платформа: Windows-10-10.0.19041-SP0
Система: Windows
Процесор: Intel64 Family 6 Model 142 Stepping 9, GenuineIntel
Архітектура:
('64bit', 'WindowsPE')
Версія Python: 3.8.6
Версія Windows: 10
Ось ваша детальна інформація про систему.

```

Рис 3.5 – Інтерфейс програмного забезпечення

Використання консольного інтерфейсу допоможе зменшити використання оперативної та постійної пам'яті пристрою, що допоможе йому паралельно запускати декілька інших програмних продуктів, а також використовувати програмне забезпечення на більш застарілих пристроях.

3.4 Інструкція користувача

Запуск програми здійснюється за допомогою виконавчого файлу Alex.exe, після запуску якого перед користувачем з'явиться консольне вікно, а також значок мікрофону у системному треї, як це показано на рисунку 3.9.

					ДППЗ.170115.01.14.ПЗ	Арк.
						49
Зм.	Арк	№ докум.	Підпис	Дата		



Рисунок 3.9 – Значок мікрофону у системному треї

Даний значок означає, що програма «слухає» користувача і використовує мікрофон, який вказаний у системі за замовчуванням. Для правильної роботи програмного забезпечення рекомендується не використовувати мікрофон в інших програмах під час використання голосового помічника або використовувати сторонній мікрофон.

Без робочого мікрофону мікрофону або пристрою, який зможе емулювати його роботу функціонування програми стає неможливим.

Якщо користувач не впевнений, чи працює його мікрофон і чи налаштований він правильно, то він може відкрити «Звуки» (рисунок 3.10), перейти у вкладку «Запис» та, натиснувши на праву кнопку мишки на потрібному мікрофоні, встановити його за замовчуванням, після чого натиснути кнопку «Застосувати» та «Ок» (рисунок 3.11). У разі відсутності мікрофону у списку потрібно встановити драйвери для нього. Якщо користувач у якості мікрофону використовує смартфон і використовує стороннє ПЗ для емуляції мікрофону, то драйвери для такого мікрофону встановлюються автоматично.

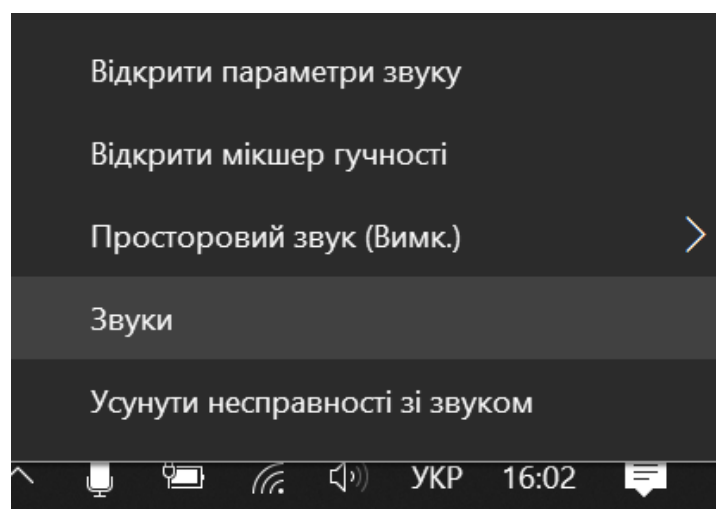


Рисунок 3.10 – Налаштування звуку

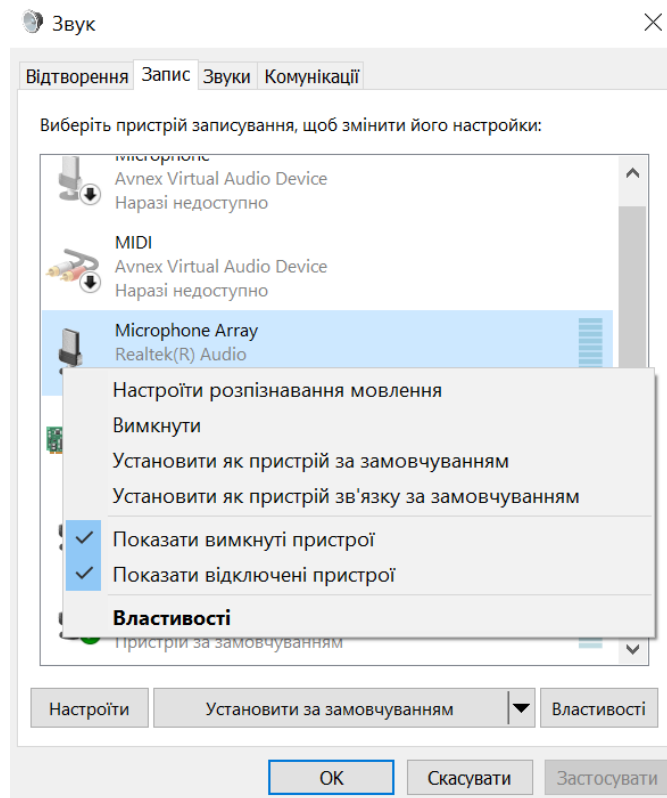


Рисунок 3.11 – Вибір мікрофону по замовчуванню

Далі користувач вводить команду, яка починається з імені «Алекс», або декількох інших його варіацій. Це допоможе боту розуміти, що звертаються саме до нього. У разі відсутності звернення помічник повідомить, що голос користувача не розпізнано і виведе відповідне повідомлення на екран.

Якщо з'єднання з мережею Інтернет відсутнє, тоді голосовий помічник виведе помилку про відсутність Інтернету.

У разі неможливості розпізнавання голосу помічник повідомить про це користувача і порекомендує налаштувати мікрофон.

Іноді може виникнути проблема з мовним шумом, тобто помічник може використовувати англійський мовний пакет. Для вирішення цієї проблеми потрібно встановити мовний рушій RHVoice та голосовий пакет Anatol та також переключити голос по замовчуванню на Anatol.

На рисунку 3.12 зображено перемикання голосового пакету для операційної системи Windows 10.

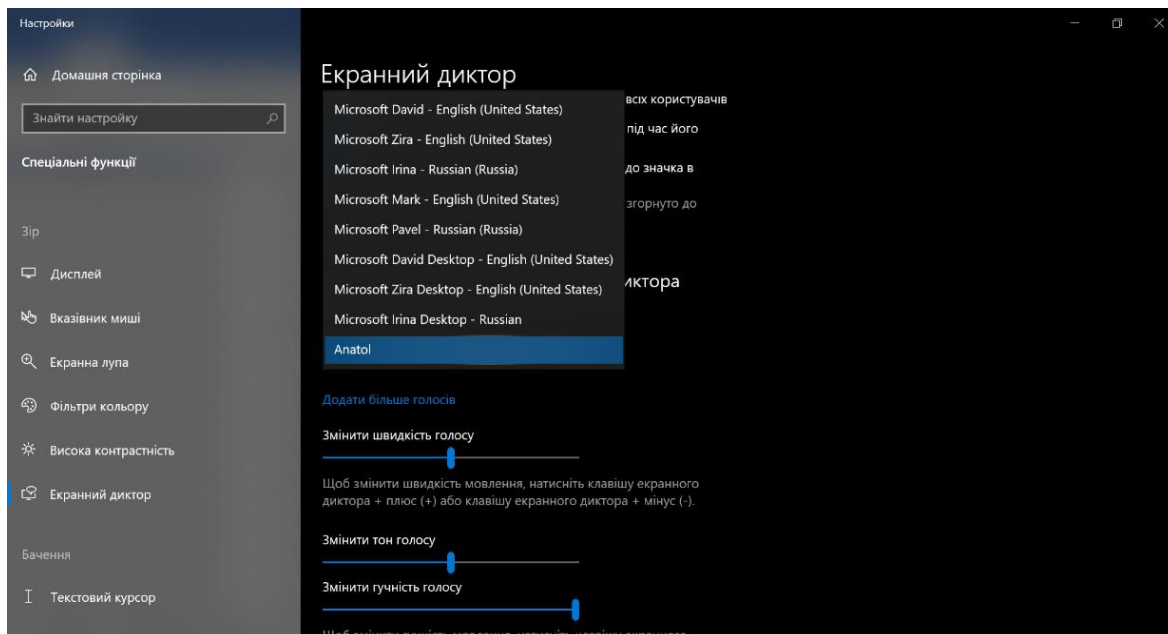


Рисунок 3.12 – Перемикання мовного пакету за замовчуванням

Після виконання помічником введеної команди на панелі системного трею знову з'явиться значок мікрофона, що свідчить про те, що голосовий помічник використовує мікрофон і готовий виконувати наступну команду.

Отже, у під час програмної реалізації було описано, який метод введення даних було вибрано для написання коду програмного продукту, а також вказані налаштування програмного забезпечення для цього методу. Після цього було детально описано модулі та бібліотеки, що використовувались при розробці програмного забезпечення та описано процес компіляції програми для її самостійного функціонування. Останнім кроком було написання інструкції для користувача, яка допоможе його для налаштування програмного забезпечення та правильного його функціонування.

Фрагменти коду програмного забезпечення наведено у додатку Б.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вибір та обґрунтування методів тестування додатку

Для мови програмування Python існує багато методів тестування додатку. Автоматичні тести використовують для автоматизації тестування програмного коду, за допомогою скрипту. Одними з найбільш відомих методів є модульне та інтеграційне тестування.

Модульний тест (unittest) дозволяє перевірити лише окремий компонент програми, що дозволить визначити неполадку та усунути, на відміну від інтеграційного, який не може перевірити цей компонент окремо, а лише їх взаємодію між собою.

Варто врахувати, що команди голосового помічника працюють незалежно одна від одної. Тому було прийнято рішення провести модульне тестування команд голосового помічника.

Тоді як команди працюють незалежно одна від одної, основні функції працюють взаємопов'язано між собою, виконуючи певне завдання і передаючи результат іншій функції. Тому було вирішено провести інтеграційне тестування для функцій.

Також існує безліч виконавців для тестів, тобто фреймворки. Найбільш популярними є pytest, unittest та nose.

Основні відмінності між цими фреймворками:

- nose використовують у більшості випадків для швидкого тестування;
- pytest може виконувати декілька тестів паралельно, тобто це дозволяє економити час на тестуванні програмного забезпечення;
- unittest використовує базовий клас TestCase для тестування, а також є найбільш відомим фреймворком для проведення тестів.

Кожен з цих фреймворків використовують при різних потребах та різних вимогах до тестування, залежно від складності програмної системи.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		53

4.2 Доведення працездатності програми

Для доведення працездатності програмного забезпечення буде використовуватись unit-тестування. Unit-тестування використовують для тестування окремих компонентів програми. У даному випадку буде доречним тестування команд голосового помічника.

У таблиці 4.1 показано сценарії проведення модульних тестів для команд.

Таблиця 4.1 – Сценарії проведення модульних тестів

Команда	Опис	Вхідні дані	Очікуваний результат
1	2	3	4
ctime	Показ поточного часу	Голосовий запит	Відображається поточний час
browser	Запуск браузеру	Голосовий запит	Відкривається браузер, що є встановленим за замовчуванням у системі.
shutdown	Вимкнення пристрою	Голосовий запит	Персональний комп'ютер вимикається на протязі 10 секунд.
restart	Перезавантаження пристрою	Голосовий запит	Пристрій перезавантажується зі затримкою у 10 секунд.

Продовження таблиці 4.1

logoff	Вихід з облікового запису	Голосовий запит	Вихід із облікового запису користувача через 10 секунд
sysinfo	Інформація про систему	Голосовий запит	Виводиться інформацію про операційну систему, процесор, тип машини, платформу і т.д.
meminfo	Інформація про пам'ять	Голосовий запит	Виводиться інформацію про загальну, використану та вільну постійну пам'ять та про використану оперативну пам'ять.
location	Інформація про місцезнаходження	Голосовий запит	Виводиться інформацію про місцезнаходження користувача та його провайдера, використовуючи інтернет-ресурси.
weather	Інформація про поточну погоду	Голосовий запит	Виводиться усю детальну інформацію про поточну погоду (температуру, тиск та

Продовження таблиці 4.1

1	2	3	4
weather	Інформація про поточну погоду	Голосовий запит	швидкість вітру), використовуючи інтернет-ресурс OpenWeatherMap
cam	Увімкнення веб-камери	Голосовий запит	<ol style="list-style-type: none"> 1. Вмикається індикатор веб-камери 2. З'являється вікно з зображенням з веб-камери. 3. Знімок зберігається при натисненні на «Пробіл» 4. Знімок зберігається у папку з проектом. 5. Вихід здійснюється за допомогою затиснення клавіші «Esc»
ping	Тестування роботи програми	Голосовий запит	Показує час роботи та використання оперативної пам'яті програмним забезпеченням та поточну швидкість з'єднання з мережею

Кінець таблиці 4.1

1	2	3	4
quit	Завершення роботи програми	Голосовий запит	Програма завершує роботу та перестає використовувати мікрофон

Наступним кроком буде проведення інтеграційних тестів. Інтеграційні тести допоможуть протестувати взаємодію компонентів програмного забезпечення. Доцільним буде проведення тестування взаємодії функцій для розпізнавання мовлення, порівнювання команд та виконання.

Таблиця 4.2 – Сценарії проведення інтеграційних тестів

Функція	Опис	Вхідні дані	Очікуваний результат
callback	Розпізнавання мовлення	Голосова команда	Перетворює введenu голосову команду у текст, виводить її та передає функції recognize_cmd
recognize_cmd	Порівняння команди	Команда у вигляді тексту.	Порівнює перетворену у текст команду з готовим набором команд у словнику. Результат передає функції execute_cmd
execute_cmd	Виконання команди	Команда у вигляді тексту.	Виконує команду з найбільшим відсотком подібності.

4.3 Результати тестування програмного забезпечення

У ході виконання тестування було проведено модульні та інтеграційні тести. За результатами тестування у таблиці 4.3 було наведено текстовий сценарій модульного та інтеграційного тестувань.

Таблиця 4.3 – Текстовий сценарій тестувань

Процес	Опис	Очікуваний результат	Результат
1	2	3	4
ctime	Показує поточний час	Відображається поточний час	Правильно
browser	Відкриває браузер	Відкривається браузер, що є встановленим за замовчуванням у системі.	Правильно
shutdown	Вимкнення пристрою	Персональний комп'ютер вимикається на протязі 10 секунд.	Правильно
restart	Перезавантаження пристрою	Персональний комп'ютер перезавантажується на протязі 10 секунд.	Правильно
logoff	Вихід із облікового запису користувача	Програма виконує вихід зі активного облікового запису користувача через 10 секунд	Правильно
sysinfo	Інформація про систему	Виводиться інформація про операційну	Правильно

Продовження таблиці 4.3

1	2	3	4
sysinfo	Інформація про систему	систему, процесор, тип машини, платформу і т.д.	Правильно
meminfo	Інформація про пам'ять	Виводиться інформація про загальну, використану та вільну	Правильно
meminfo	Інформація про пам'ять	постійну пам'ять та про використану оперативну пам'ять.	Правильно
location	Інформація про місцезнаходження	Виводиться інформація про місцезнаходження користувача та його провайдера, використовуючи інтернет-сервіси	Правильно
weather	Інформація про поточну погоду	Виводить усю детальну інформацію про поточну погоду (температуру, тиск та швидкість вітру), використовуючи ресурс OpenWeatherMap та інтернет-адресу користувача для визначення його розташування	Правильно

Продовження таблиці 4.3

1	2	3	4
ping	Тестування роботи програми	Показує час роботи, швидкість інтернету та використання оперативної пам'яті програмним забезпеченням	Правильно
cam	Увімкнення веб-камери	<ol style="list-style-type: none"> 1. Вмикається індикатор веб-камери 2. З'являється вікно з зображенням з веб-камери. 3. Знімок зберігається при натисненні на «Пробіл» 4. Знімок зберігається у папку з проектом. 5. Вихід з програми здійснюється завдяки затисненню клавіші «Esc» 	Правильно
quit	Завершення роботи програми	Програма завершує роботу та перестає використовувати мікрофон	Правильно
callback	Розпізнавання мовлення	Виводить команду у вигляді тексту та	Правильно

Кінець таблиці 4.3

1	2	3	4
callback	Розпізнавання мовлення	передає функції recognize_cmd	Правильно
recognize_cmd	Порівняння команди	Порівнює конвертовану у текст команду з готовим набором команд, що	Правильно
recognize_cmd	Порівняння команди	знаходяться у словнику. Результат передає функції execute_cmd.	Правильно
execute_cmd	Виконання команди	Виконує команду з найбільшим відсотком подібності.	Правильно

У ході проведення модульних та інтеграційних тестів було з'ясовано, що програмне забезпечення працює правильно і задовольняє усі вимоги та потреби до програмного забезпечення. Для цього було проведено модульні тести для команд, а також інтеграційні тести для основних функцій програми.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

ВИСНОВКИ

На сьогоднішній день майже кожен пристрій має голосового помічника, основним завданням якого є пошук інформації в інтернеті, автоматизація управління пристроєм та відправка повідомлень.

Темою дипломного проекту є розробка програмного забезпечення для голосового управління процесами комп'ютера. Для розробки програмного забезпечення було спершу проведено аналіз предметної області, визначено актуальність тему, а згодом було досліджено недоліки та переваги наявного програмного забезпечення.

Під час проектування було визначено та досліджено тип додатку, а також було побудовано загальну схему роботи програмного забезпечення, та діаграми, які допоможуть зрозуміти логіку програмного забезпечення та зроблять простішим написання коду. Також було визначено необхідні компоненти для розробки програмного продукту, а саме було вибрано мову програмування, яка задовільнить умови та потреби для розробки програми, було досліджено недоліки та переваги мовних рушіїв та той, який найбільше задовольняє критерії для розробки. Також було вибрано основні бібліотеки та середовище для написання коду та описано команди, які буде виконувати голосовий помічник після завершення розробки. Після цього було описано голосовий інтерфейс програмного продукту.

У ході програмної реалізації було описано реалізацію введення та обробки голосових даних, модулі та бібліотеки, що використовувались при розробці програмного забезпечення, а також їх особливості використання із наведенням прикладу у вигляді фрагментів коду.

Після цього було з'ясовано вид користувацького інтерфейсу, який повинен задовольняти критерії розробки програмного продукту. Для розробки користувацького інтерфейсу було вибрано інтерфейс консольного вікна, який допоможе зменшити використання ресурсів програмним забезпечення, у

					ДППЗ.170115.01.14.ПЗ	Арк.
						62
Зм.	Арк	№ докум.	Підпис	Дата		

результаті чого користувач може використовувати декілька програм паралельно і не закривати їх у разі відсутності великої кількості оперативної пам'яті на персональному комп'ютері чи ноутбуці.

Також було описано процес компіляції проекту з усіма його особливостями для подальшого його використання, незалежно від середовища розробки та створено детальну інструкцію для користувача, у якій було описано усі труднощі, з якими може зіштовхнутися користувач та їх подальше усунення.

У ході тестування програми було здійснено опис тест-кейсів для виявлення проблем при виконанні команд, а також при роботі основних функцій. Було проведено модульне тестування для команд та інтеграційне тестування для функцій програмного забезпечення.

Проведене тестування показало, що програмне забезпечення, його команди та функції працюють належним чином.

Отже, в ході виконання дипломного проекту було розроблено програмне забезпечення для голосового управління процесами комп'ютера, яке повністю задовольняє усі критерії та вимоги, які були описані під час дослідження предметної області.

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Л. П. Бедратюк. Дипломний проект : методичні вказівки щодо його виконання для студентів спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук, Ю. В. Форкун, О. М. Яшина. – Хмельницький: ХНУ, 2020. – 77 с.
2. Дэвид М. Бизли. Python. Подробный справочник, 4-е издание. – Перевод с английского. / Дэвид Бизли. – СПб.: Символ-Плюс, 2010. – 864 с.
3. Марк Саммерфилд. Программирование на Python 3. Подробное руководство. – Перевод с английского. / Марк Саммерфилд. – СПб.: Символ-Плюс, 2009. - 608 с.
4. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. / Гэддис Тони. – СПб.: БХВ-Петербург, 2019. – 768 с.
5. Голосовые ассистенты: что они умеют и чем отличаются друг от друга [Электронный ресурс] / AI Conference. – Режим доступа к ресурсу: <https://aicongference.com.ua/ru/news/golosovie-assistenti-chto-oni-umeyut-i-chem-otlichayutsya-drug-ot-druga-97306>
6. В Україні запрацював Google Assistant [Електронний ресурс] / Tokar.ua // Костянтин Котов. – Режим доступу до ресурсу: <https://tokar.ua/read/26613>
7. Найрозумніші голосові помічники [Електронний ресурс] / iTech.ua // Андрій Харитоненко. – Режим доступу до ресурсу: <https://itechua.com/technologies/79091>
8. 8 голосових помічників для Android [Електронний ресурс] / HTech // Федор Казаков. – Режим доступу до ресурсу: <http://headtechnology.com.ua/6-8-golosovikh-pomichnikov-dlya-android.html>
9. Голосові асистенти зламали вбудованим у музику прихованим повідомленням [Електронний ресурс] / Tehnot // Юрій Орос. – Режим доступу до ресурсу: <https://tehnot.com/ua/golosovye-assistenty-vzломали-vstroennym-v-muzyku-skryтым-soobshheniem/>

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		64

10. Умная колонка сама заказала игрушку после истерики сына пресс-секретаря Трампа [Электронный ресурс] / Life.ru // Сидорова Эмила. – Режим доступа к ресурсу: <https://life.ru/p/1079107>
11. Опубликован Python 2.7.18, последний выпуск ветки Python 2 [Электронный ресурс] / Open.NET. – Режим доступа к ресурсу: <https://www.opennet.ru/opennews/art.shtml?num=52772>
12. The Python Package Index [Electronic resource] / Python Package Index. - Available: <https://pypi.org>
13. Python functions [Electronic resource] / W3Schools. – Available: https://www.w3schools.com/python/python_functions.asp
14. Синтезатор RHVoice [Electronic resource] / NVDA.RU – Социально-информационный проект для людей с ограниченными возможностями по зрению – Доступ к ресурсу: <https://nvda.ru/sintezator-rhvoice>
15. PyCharm [Electronic resource] / JetBrains – Available: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>
16. PyCharm: IDE для Python [Электронный ресурс] / GeeksforGeeks. – Режим доступа к ресурсу: <https://timeweb.com/ru/community/articles/pycharm-ide-dlya-python-1>
17. Import module [Electronic resource] / Python Docs. – Available: <https://docs.python.org/3/reference/import.html>
18. Python Dictionary [Electronic resource] / GeeksforGeeks. – Режим доступа к ресурсу: <https://www.geeksforgeeks.org/python-dictionary/>
19. Using pytsx3 [Electronic resource] / Read The Docs. – Available: <https://pytsx3.readthedocs.io/en/latest/engine.html>
20. psutil documentation [Electronic resource] / Read The Docs. – Available: <https://psutil.readthedocs.io/en/latest/>
21. shutil — High-level file operations [Electronic resource] / Read The Docs. – Available: <https://docs.python.org/3/library/shutil.html>

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		65

22. OpenCV Python Tutorial [Электронный ресурс] / GeeksforGeeks. – Режим доступа к ресурсу: <https://www.geeksforgeeks.org/opencv-python-tutorial/>
23. Введение в тестирование в Python [Электронный ресурс] / Еще один блог веб-разработчика // Editorial Team. – Режим доступа к ресурсу: <https://webdevblog.ru/vvedenie-v-testirovanie-v-python/>

					ДППЗ.170115.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		66

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується в рамках проекту розробки програмного забезпечення для голосового управління процесами комп'ютера.

1. Підстава для розробки

Підставою для розробки є «Завдання на дипломний проект», затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: програмне забезпечення для голосового управління процесами комп'ютера.

2. Призначення розробки

Програмне забезпечення для голосового управління процесами комп'ютера призначене для управління процесами комп'ютера за допомогою голосових команд та виведення інформації про пристрій.

Користувачами системи є звичайні користувачі персонального комп'ютера чи ноутбуку.

Функціональне призначення передбачає: вимкнення та перезавантаження комп'ютера, відкриття браузера та програм, виведення інформації про місцезнаходження користувача, а також інформацію про погоду та характеристики пристрою.

Програмне забезпечення працюватиме на будь-якому комп'ютері під управлінням операційної системи Windows.

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик

Програмне забезпечення для голосового управління комп'ютером повинне виконувати функції, подані нижче:

- вимкнення та перезавантаження комп'ютера;

- відкриття браузера;
- виведення інформації про комп'ютер (його характеристики);
- виведення інформації про пам'ять;
- увімкнення веб-камери;
- виведення інформації про місцезнаходження користувача;
- виведення інформації про погоду.

3.2. Вимоги до надійності

Програмне забезпечення повинно виконувати наступні вимоги:

- обробка помилок;
- виконання команд;
- виведення результату на екран.

3.3. Умови експлуатації

В ході розробки програмного забезпечення повинні бути підготовлені: текст та опис програми, програма та керівництво користувача.

3.4. Вимоги до складу та параметрів технічних засобів

- платформа: Windows 10, 8, 8.1, 7;
- розрядність: x64 (64-bit);
- процесор: Pentium 4 з SSE2;
- відеоадаптер: nVidia, Intel, AMD / ATI;
- вінчестер: 512 Mb;
- оперативна пам'ять: 512 Mb;
- контролер: миша, клавіатура, мікрофон;
- аудіокарта: будь-яка;
- інтернет: стабільне з'єднання;
- роздільна здатність екрану: SVGA 800x600.

3.5. Вимоги до інформаційної та програмної сумісності

Для розробки програмного забезпечення будуть використовуватися технології, наведені нижче:

- Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня;
- RNVoice – синтез мовлення з відкритим програмним кодом.
- Microsoft Speech Recognizer – розпізнавання мовлення від компанії Microsoft
- PyAudio – інструмент для роботи зі звуком

3.6. Спеціальні вимоги

Програма повинна мати мінімалістичний дизайн інтерфейсу для зменшити навантаження на комп'ютер.

4. Вимоги до програмної документації

Програмна документація повинна включати такі документи наступний набір документів:

- текст програми з коментарями та поясненнями;
- відомості про функціонування програми;
- технічне завдання;
- інструкція користувача;
- керівництво програмісту.

5. Стадії та етапи розробки

Стадії та етапи розробки програмного забезпечення для голосового управління процесами комп'ютера показані у таблиці 1.4.

Таблиця 1.4 – Стадії та етапи розробки проекту

Стадія розробки	Етапи робіт	Зміст робіт
1	2	3
Технічне завдання 02.01.21 – 31.01.21	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02.21 – 14.02.21	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися
Технічний проект 15.02.21 – 28.02.21	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів
Робочий проект 01.03.21 – 10.04.21	Розробка програмного забезпечення	Реалізація програмного забезпечення; відлагодження; проведення попереднього тестування

Кінець таблиці 1.4

1	2	3
Розробка програмної документації 11.04.21 – 20.04.21	Розробка документації для програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 21.04.21 – 30.04.21	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і розгортання програмного забезпечення

б. Порядок контролю та приймання

Контроль здійснюється кінцевими користувачами системи, підключеними на етапі тестування програмного забезпечення. Прийом комплексу здійснюється після налаштування програмного забезпечення та окремих його компонентів для нормального функціонування.

ДОДАТОК Б (обов'язковий)

ФРАГМЕНТИ КОДУ ПРОГРАМНОЇ СИСТЕМИ

Код програми

```

import os
import datetime
import time
from urllib.request import urlopen
import pyttsx3
import requests
import speech_recognition as sr
import webbrowser as wb
import speedtest
from fuzzywuzzy import fuzz
import platform
import psutil
import shutil
from memory_profiler import memory_usage
import json
import cv2
from translate import Translator

start_time = time.time()
param = {
    "name": ('алекс', 'бот', 'робот'),
    "cmds": {
        "ctime": ('скільки годин', 'котра година', 'яка година'),
        "browser": ('відкрий браузер', 'запусти браузер'),
        "shutdown": ('вимкни комп', 'вимкни пк', 'виключи комп', 'включи пк'),
        "restart": ('перезавантаж комп', 'перезагрузи пк', 'перезагрузи комп', 'перезавантаж пк'),
        "logoff": ('вийди з облікового запису', 'вийди з системи'),
        "ping": ('тест', 'пінг', 'відкрий тест'),
        "sysinfo": ('інформація про систему', 'характеристики пристрою', 'характеристики'),
        "meminfo": ('інформація про пам'ять', 'кількість вільної пам'яті'),
        "location": ('де я', 'моє розташування'),
        "weather": ('погода', 'яка зараз погода', 'погода зараз'),
        "cam": ('камера', 'зроби фото', 'ввімкни камеру'),
        "quit": ('бувай', 'пока', 'дякую', 'досить', 'стоп')
    }
}

speak_engine = pyttsx3.init()

def speak(sometext):
    print(sometext)
    voices = speak_engine.getProperty('voices')
    for voice in voices:
        speak_engine.setProperty('voice', voice.id)
    speak_engine.say(sometext)
    speak_engine.runAndWait()
    speak_engine.stop()

```

```

def callback(recognizer, audio):
    try:
        voice = recognizer.recognize_google(audio, language="uk-UA").lower()
        print("[log] Розпізнано: " + voice)

        if voice.startswith(param["name"]):
            # звернення
            cmd = voice

            for x in param['name']:
                cmd = cmd.replace(x, "").strip()

            #розпізнавання та виконання
            cmd = recognize_cmd(cmd)
            execute_cmd(cmd['cmd'])

    except sr.UnknownValueError:
        print("[log] Голос не розпізнано! Можливо, ви вибрали неправильний мікрофон в налаштуваннях системи!")
        speak("Голос не розпізнано! Можливо, ви вибрали неправильний мікрофон в налаштуваннях системи!")
    except sr.RequestError as e:
        print("[log] Відсутній доступ в Інтернет! Будь ласка, підключіться до Інтернету")
        speak("Відсутній доступ в Інтернет! Будь ласка, підключіться до Інтернету")

def recognize_cmd(cmd):
    RC = {'cmd': '', 'percent': 0}
    for c, v in param['cmds'].items():

        for x in v:
            vrt = fuzz.ratio(cmd, x)
            if vrt > RC['percent']:
                RC['cmd'] = c
                RC['percent'] = vrt

    return RC

def execute_cmd(cmd):
    if cmd == 'ctime':
        now = datetime.datetime.now()
        speak("Запаз " + str(now.hour) + ":" + str(now.minute))

    elif cmd == 'browser':
        wb.open("http://google.com")

    elif cmd == 'shutdown':
        speak("Ваш комп'ютер буде вимкнено через 10 секунд. Закрийте усі зайві програми та збережіть дані")
        os.system('shutdown -s -t 10')

    elif cmd == 'restart':
        speak("Ваш комп'ютер буде перезавантажено через 10 секунд. Закрийте усі зайві програми та збережіть дані")
        os.system('shutdown -r -t 10')

    elif cmd == 'logoff':
        speak("Вихід із облікового запису відбудеться через 10 секунд.")
        os.system('shutdown -l -t 10')

```

```

elif cmd == 'ping':
    wtime = round((time.time() - start_time), 2)
    speak("Програма працює %s секунд" % wtime)
    print("Використано оперативної пам'яті програмою (МБ): ")
    a = memory_usage()
    print(a)
    stest = speedtest.Speedtest()
    download = stest.download()
    upload = stest.upload()
    speak("Будь ласка, зачекайте, вимірюю швидкість вашої мережі. Це може
зайняти певний час")
    print(f"Швидкість завантаження: {(download / 1024) / 1024} МБ/с \n
Швидкість вивантаження : {(upload / 1024) / 1024} МБ/с")
    print("ОК")

elif cmd == 'sysinfo':
    m = platform.machine()
    print("Тип машини: " + m)
    ver = platform.version()
    print("Версія: " + ver)
    plt = platform.platform()
    print("Платформа: " + plt)
    sys = platform.system()
    print("Система: " + sys)
    pr = platform.processor()
    print("Процесор: " + pr)
    arc = platform.architecture()
    print("Архітектура: ")
    print(arc)
    pv = platform.python_version()
    print("Версія Python: " + pv)
    rel = platform.release()
    print("Версія Windows: " + rel)
    speak("Ось ваша детальна інформація про систему.")

elif cmd == 'meminfo':
    psutil.cpu_percent()
    psutil.virtual_memory()
    dict(psutil.virtual_memory()._asdict())
    used_ram = psutil.virtual_memory().percent
    total, used, free = shutil.disk_usage("\\")
    speak("Загально %d гігабайт пам'яті.\n" % (total // (2 ** 30)) +
          "Використано з них %d гігабайт.\n" % (used // (2 ** 30)) +
          "Вільно: %d гігабайт.\n" % (free // (2 ** 30)) +
          "Використано " + str(used_ram) + " відсотків оперативної пам'яті.")

elif cmd == 'location':
    url = 'http://ipinfo.io/json'
    response = urlopen(url)
    data = json.load(response)
    IP = data['ip']
    org = data['org']
    city = data['city']
    country = data['country']
    region = data['region']
    translator = Translator(to_lang="Ukrainian")
    cityukr = translator.translate(city)
    regukr = translator.translate(region)
    speak("Ви знаходитесь в " + cityukr + ", " + regukr +
          "\nВаш провайдер: компанія " + org)

elif cmd == 'weather':

```

```

url = 'http://ipinfo.io/json'
response = urlopen(url)
data = json.load(response)
city = data['city']
city_id = 0
appid = "c38ac14c60eaba6b61775a83af3c5e68"
try:
    res = requests.get("http://api.openweathermap.org/data/2.5/find",
                        params={'q': city, 'type': 'like', 'units':
'metric', 'APPID': appid})
    data = res.json()
    cities = ["{} ({}).format(d['name'], d['sys']['country'])
              for d in data['list']]
    print("city:", cities)
    city_id = data['list'][0]['id']
except Exception as e:
    print("Помилка:", e)
try:
    res = requests.get("http://api.openweathermap.org/data/2.5/weather",
                        params={'id': city_id, 'units': 'metric', 'lang':
'uk', 'APPID': appid})
    print("Місто:", city)
    data = res.json()
    speak("Сьогодні очікується: " + data['weather'][0]['description'])
    speak("Температура повітря дорівнює " + str(data['main']['temp']) + "
градусів Цельсію.")
    speak("Відчувається як " + str(data['main']['feels_like']) + "
градусів Цельсію.")
    press = int(data['main']['pressure'])
    normpress = (press * 100) / 133
    normpress = int(normpress)
    speak("Тиск:" + str(data['main']['pressure']) + " гектопаскаль")
    speak("Або " + str(normpress) + " міліметрів ртутного стовпчика")
    speak("Швидкість вітру:" + str(data['wind']['speed']) + " метрів на
секунду")
except Exception as e:
    print("Помилка:", e)

elif cmd == 'cam':
    cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
    speak("Для збереження змінку натисніть на клавішу пробіл, для виходу з
камер - утримуйте клавішу ESC")
    while True:
        count = 0
        ret, img = cap.read()
        cv2.imshow("XYREXX", img)
        if cv2.waitKey(10) == 32:
            cv2.imwrite("frame%d.jpg" % ret, img)
            count += 1
        if cv2.waitKey(10) == 27: # Вимкнення на ESC
            break
    cap.release()
    cv2.destroyAllWindows()

elif cmd == 'quit':
    speak("Допобачення. Гарного вам дня!")
    os.system('cls')
    exit(0)

else:
    print('Вибачте, але я не знаю такої команди')

```

```
# запустить
r = sr.Recognizer()
#m = sr.Microphone(device_index=2)
m = sr.Microphone()
with m as source:
    while True:
        audio = r.listen(source)
        callback(r, audio)
stop_listening = r.listen_in_background(m, callback)
while True: time.sleep(0.1)
```

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ СЛАЙДИ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ ПРОЦЕСАМИ КОМП'ЮТЕРА

Виконав:

студент 4 курсу групи ІПЗ-17-1
Павлюк В.А.

Керівник:

канд. техн. наук, доцент
Форкун Ю.В.

МЕТОЮ ПРОЕКТУ Є РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ ПРОЦЕСАМИ КОМП'ЮТЕРА, ЯКЕ МАЄ ЗРУЧНИЙ ТА МІНІМАЛІСТИЧНИЙ ІНТЕРФЕЙС.

АКТУАЛЬНІСТЬ ТЕМИ ПОЛЯГАЄ У ТОМУ, ЩО НА СЬОГОДНІ ГОЛОСОВИХ ПОМІЧНИКІВ ДЛЯ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА, ЯКІ ВИКОРИСТОВУЮТЬ УКРАЇНСЬКУ МОВУ, НЕ ІСНУЄ.

ДЛЯ **РЕАЛІЗАЦІЇ ПРОЕКТУ** НЕОБХІДНО ВИКОНАТИ НАСТУПНІ ЗАВДАННЯ:

- ВИЗНАЧИТИ СПЕЦИФІКУ ПРЕДМЕТНОЇ ОБЛАСТІ;
- ПРОВЕСТИ АНАЛІЗ ПОДІБНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ВИЗНАЧИТИ ЇХ ПЕРЕВАГИ ТА НЕДОЛІКИ;
- ВСТАНОВИТИ ОСОБЛИВОСТІ ВИКОРИСТАННЯ БІБЛІОТЕК ДЛЯ РОЗПІЗНАВАННЯ ГОЛОСУ ТА ПЕРЕТВОРЕННЯ ТЕКСТУ В МОВУ.
- ПРОВЕСТИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

НАЯВНІСТЬ ПОДІБНИХ ПЗ

GOOGLE ASSISTANT

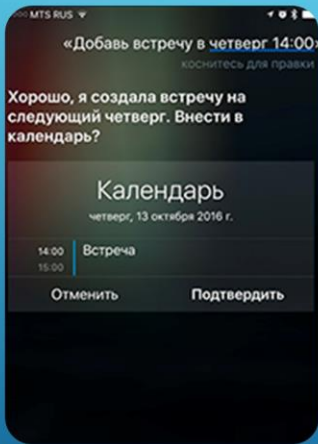
На даний момент це один з найпопулярніших голосових помічників. Він працює на пристроях, які управляються операційною системою Android та iOS. Помічник дозволяє керувати пристроєм, шукати інформацію в інтернеті, телефонувати, а також відповідати автоматично на дзвінки (ця функція недоступна в Україні і на даний момент працює лише на фірмових пристроях компанії «Google» Pixel). На даний момент помічник не вміє розпізнавати українську мову.



НАЯВНІСТЬ ПОДІБНИХ ПЗ

SIRI

Даний помічник був створений у 2011 році і працює лише на пристроях від компанії «Apple». Функціонал подібний з Google Assistant. Перевагою даного помічника є те, що він доступний («з коробки»), в той час, як для коректної роботи Google Assistant потрібно установити фірмовий додаток Google з магазину Play Store (Android) чи App Store (iOS).



АЛИСА

Окей гугл

Сири?

Слушай кортана

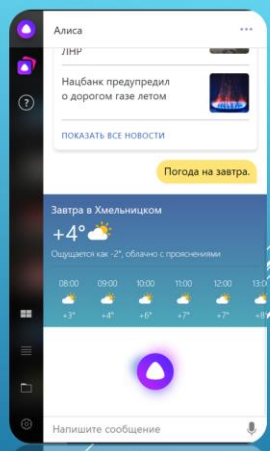
Это вот обидно было.

Какая тонкая шутка. Сарказм.

Если у вас была задача нанести голосовому помощнику психологическую травму, вы отлично справились. И нет, я не Кортана.

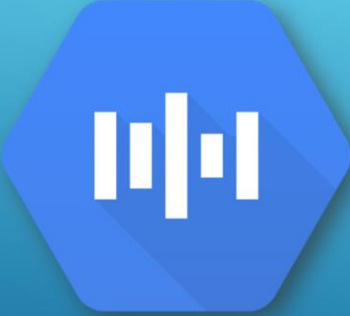
Даний помічник був створений у 2017 році. На відміну від «Google Assistant» та «Siri» даний помічник не вмiє на даний момент управляти пристроєм, але відрiзняється високоякісним синтезом мовлення, який компанія «Яндекс» створила спеціально для цього помічника. «Аліса» вмiє шукати інформацію в інтернеті, читати новини або просто спілкуватись. На території України працює лише при ввімкненому VPN чи проксі. Для роботи даного помічника потрібен браузер чи лаунчер від компанії «Яндекс».

Даний помічник також має версію для персональних комп'ютерів.



ВИБІР ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ГОЛОСУ

GOOGLE SPEECH RECOGNITION



GOOGLE SPEECH RECOGNITION – це технологія розпізнавання голосу від компанії Google. Працює онлайн, використовуючи сервери компанії Google, що дозволить економити ресурси пристрою.

Також існує **Google Cloud Speech**, який використовує більш потужну нейронну модель розпізнавання мовлення та підтримує близько 80 мов. Основним недоліком є те, що ця технологія платна та потребує використання **API**

ВИБІР ТЕХНОЛОГІЇ ПЕРЕТВОРЕННЯ ТЕКСТУ В МОВУ



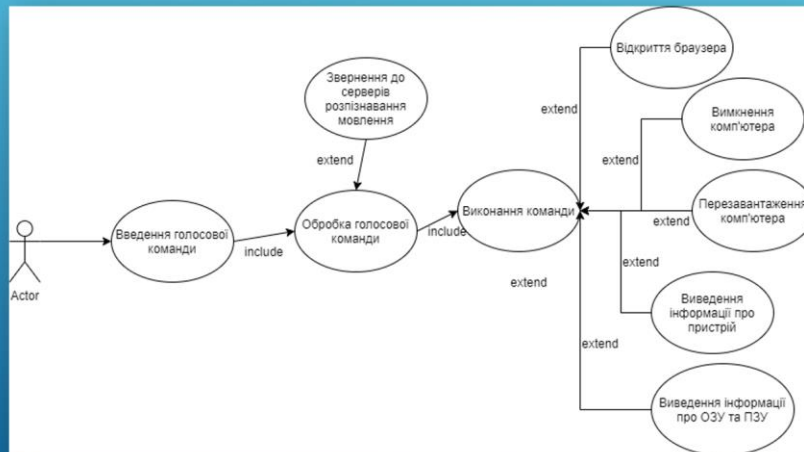
RHVOICE

Синтезатор мовлення **RHVoice** був розроблений російською програмісткою Ольгою Яковлевою та використовувався спершу, як екранний диктор для людей, які мають проблеми із зором, але на даний момент працює, як самостійна програма.

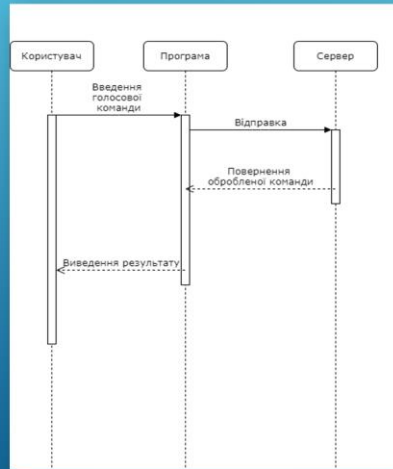
Програмний продукт має відкритий програмний код та підтримує 8 різних мов, зокрема українську.

Основною перевагою над іншими україномовними синтезаторами є те, що **RHVoice** може безпроблемно використовуватись при розробці програмного забезпечення.

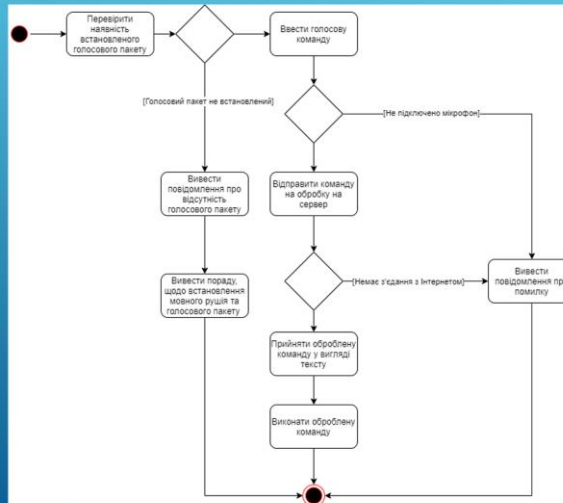
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ ДЛЯ ГОЛОСОВОГО ПОМІЧНИКА:



ДІАГРАМА ПОСЛІДОВНОСТЕЙ ДЛЯ ГОЛОСОВОГО ПОМІЧНИКА:



ДІАГРАМА ДІЯЛЬНОСТІ ДЛЯ ГОЛОСОВОГО ПОМІЧНИКА:



ІНТЕРФЕЙС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
C:\Users\User\PycharmProjects\DP_assiv2\dist\Alex\Alex.exe
[log] Розпізнано: алекс інформація про пам'ять
Загально 110 гігабайт пам'яті.
Використано з них 86 гігабайт.
Вільно: 24 гігабайт.
Використано 90.9 відсотків оперативної пам'яті.
[log] Розпізнано: алекс інформ я про систему
Тип машини: AMD64
Версія: 10.0.19041
Платформа: Windows-10-10.0.19041-SP0
Система: Windows
Процесор: Intel64 Family 6 Model 142 Stepping 9, GenuineIntel
Архітектура:
('64bit', 'WindowsPE')
Версія Python: 3.8.6
Версія Windows: 10
Ось ваша детальна інформація про систему.
```

ІНТЕРФЕЙС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
C:\Users\User\PycharmProjects\DP_assiv2\dist\Alex\Alex.exe
[log] Розпізнано: алекс погода
city: ['Khmelnytskyi (UA)']
Місто: Khmelnytskyu
Сьогодні очікується: рвані хмари
Температура повітря дорівнює 19.09 градусів Цельсію.
Відчувається як 18.96 градусів Цельсію.
Тиск:1013 гектопаскаль
Або 761 міліметрів ртутного стовпчика
Швидкість вітру:5.47 метрів на секунду
[log] Розпізнано: алекс тест
Програма працює 30.79 секунд
Використано оперативної пам'яті програмою (МБ):
[105.5625]
Будь ласка, зачекайте, вимірюю швидкість вашої мережі. Це може зайняти певний час
Швидкість завантаження: 47.126524331736874 МБ/с
Швикість вивантаження : 20.47918600852179 МБ/с
OK
```

ВИСНОВКИ

У процесі виконання дипломного проекту було проведено аналіз предметної області для програмне забезпечення для голосового управління процесами персонального комп'ютера.

Було досліджено наявне програмне забезпечення, в результаті якого було з'ясовано, що більшість з них не підтримують українську мову. Це означає, що тема є актуальною. Також було проведено аналіз та порівняння наявних мовних рушіїв та засобів розпізнавання мовлення та було визначено їх переваги та недоліки.

У ході аналізу та проектування програмного забезпечення було вибрано необхідні для функціонування бібліотеки для розробки ПЗ. У результаті роботи було реалізовано програмне забезпечення для голосового управління процесами персонального комп'ютера, яке містить мінімалістичний інтерфейс, що допоможе йому використовувати менше ресурсів пристрою.

ДЯКУЮ ЗА УВАГУ!

Завідувачу кафедри інженерії
програмного забезпечення проф.
Бедратюку Л.П.
здобувача вищої освіти
Павлюка В.А.
ФПКТС, група ІПЗ-17-1, 4 курс

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

7.06.2021

дата



підпис

Anti-Plagiarism v-15.257**Максимальне співпадіння з одним документом 11.0%****Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 11%**

ID: 92959 Назва: Програмне забезпечення для голосового управління процесами комп'ютера Додано в БД: 2021-06-09 Автора: В.А. Павлюк Керівники: Ю.В. Форкун Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	68831	663	11268 (16%)	142 (21%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
90222	Назва: Звіт з переддипломної практики Павлюк В.А. Додано в БД: 2021-05-11 Автора: Павлюк В.А. Керівники: Форкун Ю.В. Консультанти: Опоненти:	7576 (11.0%)	88 (13.0%)



Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1008255530

Дата перевірки:
10.06.2021 11:48:22 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2021 11:53:59 EEST

ID користувача:
100005589

Назва документа: Диплом_Павлюк_без_додатків

Кількість сторінок: 64 Кількість слів: 8991 Кількість символів: 68244 Розмір файлу: 3.56 MB ID файлу: 1008327081

3.63% Схожість

Найбільша схожість: 1.86% з джерелом з Бібліотеки (ID файлу: 1008272523)

1.59% Джерела з Інтернету 55 Сторінка 66

2.47% Джерела з Бібліотеки 55 Сторінка 66

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 3

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ
рівня вищої освіти «Бакалавр»Дипломник Павлюк Віталій АндрійовичТема «Програмне забезпечення для голосового управління процесами комп'ютера»Спеціальність 121 – Інженерія програмного забезпечення

Обсяг дипломного проекту:

Кількість листів креслень _____ кількість сторінок записки 64

1. Короткий зміст пояснювальної записки та прийнятих рішень У бакалаврській роботі виконано аналіз наявних голосових помічників. Розроблено логіку роботи голосового помічника. Реалізовано зчитування голосових команд та їх перетворення в текст.

2. Висновок про відповідність проекту поставленому завданню Дипломна робота рівня вищої освіти «бакалавр» в повній мірі відповідає поставленому завданню як у теоретичній, так і у практичній її частині.

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі обґрунтовується актуальність теми проекту, проводиться попередній аналіз досліджуваної проблеми, встановлюється мета та завдання на виконання проекту. У першому розділі в повній мірі досліджено роботу голосових помічників, проаналізоване наявне програмне забезпечення, визначені вимоги до проектування та реалізації системи. В другому розділі проведено аналіз існуючої архітектури та підходів до реалізації. В третьому розділі проведена реалізація та повний опис реалізованої системи. В четвертому розділі проведений аналіз методів тестування та застосовані конкретні підходи в тестування програмного забезпечення.

4. Позитивні сторони проекту Дипломний проект містить детальний аналіз роботи голосового помічника, детальний опис усіх використаних бібліотек та детальну інструкцію користувача. Програма містить достатню кількість команд.

5. Негативні сторони проекту В реалізації програмного забезпечення доцільно було застосувати мультипарадигмове програмування.

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконано відповідно до теми дипломного проекту з дотриманням вимог стандартів. Рівень графічного оформлення достатній для ілюстрування всіх методів, алгоритмів та процесів, які подані в дипломному проекті.

7. Відгук про дипломний проект в цілому Загалом дипломний проект повноцінно описує вирішення поставленої задачі. Викладений матеріал у проекті послідовний та логічний. Всі розділи логічні та повноцінно описані, що дозволяє зрозуміти весь проект. Графічні матеріали наочно показують всі особливі моменти, що спрощує розуміння вирішення задачі. В цілому проект заслуговує позитивної оцінки.

8. Інші зауваження

9. Оцінка дипломного проекту Розглянувши всі позитивні та негативні сторони представленого дипломного проекту можна зробити висновок, що він заслуговує оцінки «добре» (3.75/С).

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи)

Говорущенко Тетяна Олександрівна, доктор технічних наук, професор, завідувач кафедри комп'ютерної інженерії та системного програмування Хмельницького національного університету

« 2 » червня 2021 р.


(підпис)

5. Негативні сторони проекту Програмне забезпечення може працювати нестабільно на 32-бітних системах.

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконано відповідно до теми дипломного проекту з дотриманням вимог стандартів. Рівень графічного оформлення достатній для ілюстрування всіх принципів та процесів, що описані в дипломному проекті.

7. Відгук про дипломний проект в цілому Загалом дипломний проект повноцінно описує вирішення поставленої задачі. Весь матеріал проекту послідовний та чіткий. Всі розділи логічні та повноцінно описані, що дозволяє зрозуміти весь проект. Графічні матеріали наочно показують всі особливі моменти, що спрощує розуміння вирішення задачі. В цілому проект заслуговує позитивної оцінки.

8. Інші зауваження _____

9. Оцінка дипломного проекту Розглянувши всі позитивні та негативні сторони представленого дипломного проекту можна зробити висновок, що він заслуговує оцінки «відмінно» (3.75/С).

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи)

Говорущенко Тетяна Олександрівна, доктор технічних наук, професор, завідувач кафедри комп'ютерної інженерії та системного програмування Хмельницького національного університету

« 2 » червня 2021 р.


(Підпис)