

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем і автоматики

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління

(повна назва кафедри)

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації»**

Виконала: студентка 2-го курсу,  
групи 2АКІТ-20м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології  
(шифр і назва спеціальності)

\_\_\_\_\_ Ірина Майданевич

(ім'я та прізвище)

Керівник: д.т.н., професор каф. КСУ

\_\_\_\_\_ В'ячеслав Ковтун

(ім'я та прізвище)

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

Опонент: к.т.н., доцент каф. АІТ \_\_\_\_\_

\_\_\_\_\_ Володимир Гармаш

(ім'я та прізвище)

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

**Допущено до захисту**  
Завідувач кафедри КСУ  
д.т.н., проф.

\_\_\_\_\_ Володимир Дубовой

(ім'я та прізвище)

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

Вінницький національний технічний університет  
 Факультет комп'ютерних систем і автоматики  
 Кафедра комп'ютерних систем управління  
 Рівень вищої освіти II-й (магістерський)  
 Галузь знань – 15 Автоматизація та приладобудування  
 Спеціальність – 151 Автоматизація та комп'ютерно-інтегровані технології  
 Освітньо-професійна програма Інтелектуальні комп'ютерні системи

### **ЗАТВЕРДЖУЮ**

Завідувач кафедри КСУ

д.т.н., проф.

\_\_\_\_\_ Володимир Дубовой

«01» 10 2021 року

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

\_\_\_\_\_ Майданевич Ірині Олександрівні \_\_\_\_\_

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації»

керівник магістерської кваліфікаційної роботи професор кафедри КСУ

Ковтун В'ячеслав Васильович, д. т. н., професор

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “24” 09 2021 року  
 № 277

2. Строк подання студентом магістерської кваліфікаційної роботи 10.12. 2021 року

3. Вихідні дані до магістерської кваліфікаційної роботи експлуатаційні дані з об'єкту дослідження, сучасна програмна архітектура, мова програмування Python.

4. Зміст розрахунково-пояснювальної записки огляд предметної області; розробка математичного апарату; розробка програмного забезпечення та експериментальні дослідження.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) архітектура системи; робочий процес системи; приклад вихідних даних; інтерфейс користувача розробленої системи; лістинг програмного забезпечення.

## 6. Консультанти розділів роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	доцент кафедри ЕПВМ, доцент, к.е.н. Кавецький В.В.		

7. Дата видачі завдання “ 01 ” 10 2021 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Огляд предметної області	04.09.21р.	
2	Розробка математичного апарату	22.09.21р.	
3	Розробка програмного забезпечення та експериментальні дослідження	30.10.21р.	
4	Підготовка економічної частини	12.11.21р.	
5	Оформлення пояснювальної записки і графічного матеріалу	08.12.21р.	
6	Попередній захист роботи	16.12.21р.	
7	Остаточний захист роботи	22.12.21р.	

Студент

\_\_\_\_\_  
(підпис)

Керівник роботи

\_\_\_\_\_  
(підпис)Ірина МайданевичВ'ячеслав Ковтун

## АНОТАЦІЯ

У магістерській роботі ми проаналізували велику кількість матеріалів про історію розвитку технологій розумного дому, робототехніки та Інтернету речей, розпізнавання голосу та відтворення існуючих технологій та реалізацій, що дозволило створити стабільну, просту та легку у використанні -удосконалити підпроцес однокристалльної структури програмної системи, інтерфейс для взаємодії людини та комп'ютера через голос, роботу через прослуховування, обробку, інтерпретацію та відповіді, роботу в циклах, що дозволяє безперервно виконувати розпізнавання та інтерпретацію в наступних командах, а також моніторинг роботи статус і розпізнавання за допомогою процесу веб-інтерфейсу.

Інструменти та бібліотеки, які ми використовуємо для розробки, дозволяють нам значно спростити її, полегшити підтримку в майбутньому, додати нові функції та інтегрувати в готові технологічні комплекси. Також ми провели комплексну оцінку всіх параметрів безпеки та охорони праці та отримали всі необхідні параметри для комфортного та безпечного робочого місця для розробників систем. Надалі планується подальше розширення системи та удосконалення наявних недоліків для досягнення максимальної ефективності та мінімальних помилок.

## ABSTRACT

In the master's thesis, we analyzed a large number of materials on the history of smart home technology, robotics and the Internet of Things, voice recognition and reproduction of existing technologies and implementations, which allowed to create a stable, simple and easy to use -improve single-process structure human and computer through voice, Work through listening, processing, interpretation and response, Work in cycles, which allows you to continuously perform recognition and interpretation in subsequent commands, as well as monitoring Work status and recognition through the web interface process. The tools and libraries we use for development allow us to greatly simplify it, facilitate future support, add new features and integrate into ready-made technological complexes. We also conducted a comprehensive assessment of all safety and health parameters and obtained all the necessary parameters for a comfortable and safe workplace for system developers. In the future, it is planned to further expand the system and improve the existing shortcomings to achieve maximum efficiency and minimum errors.

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>1 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ ЗАСТОСУНКІВ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ ТА ІСНУЮЧІ АНАЛОГІЇ</b> .....	10
1.1 Пошук перспективного сегменту застосування застосунків для голосового управління.....	10
1.2 Досвід створення голосових інтерфейсів.....	13
<b>2 АНАЛІЗ ТЕХНОЛОГІЇ ЗАСТОСУНКУ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ</b> .....	25
2.1 Структурна і функціонал застосунку для голосового управління...25	
2.2 Класифікація систем голосового інтерфейсу.....	31
2.3 Характеристика технологій для розпізнавання усного мовлення...32	
<b>3 ОПИС РОЗРОБКИ ВЛАСНОГО ЗАСТОСУНКУ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ</b> .....	44
3.1 Вибір середовища та методу розроблення.....	44
3.2 Вибір системи управління базою даних для власного проекту.....	47
3.3 Вибір інструментарію голосового управління для власного проекту.....	48
3.4 Компіляція і тестування власного застосунку для голосового управління.....	49
<b>4 ЕКОНОМІЧНА ЧАСТИНА</b> .....	59
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	59
4.2 Розрахунок узагальненого коефіцієнта якості розробки.....	64
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	66
4.3.1 Витрати на оплату праці.....	66
4.3.2 Відрахування на соціальні заходи.....	69
4.3.3 Сировина та матеріали.....	69

4.3.4 Розрахунок витрат на комплектуючі.....	70
4.3.5 Спецстаткування для наукових (експериментальних) робіт.....	71
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт.....	71
4.3.7 Амортизація обладнання, програмних засобів та приміщень.....	72
4.3.8 Паливо та енергія для науково-виробничих цілей.....	74
4.3.9 Службові відрядження.....	75
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	75
4.3.11 Інші витрати.....	76
4.3.12 Накладні (загальновиробничі) витрати.....	76
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	77
<b>ВИСНОВКИ.....</b>	<b>83</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>85</b>
<b>ДОДАТКИ.....</b>	<b>91</b>
Додаток А. ТЕХНІЧНЕ ЗАВДАННЯ.....	92
Додаток Б. Фрагменти програмного коду.....	95
Додаток В. ІЛЮСТРАТИВНА ЧАСТИНА.....	103

## ВСТУП

Мова є природним способом спілкування людини. Інтерфейс з голосовим керуванням може покращити існуючий інтерфейс користувача - він забезпечує більш зручний і менш обмежувальний спосіб взаємодії людей із технологіями. Якісний голосовий інтерфейс допомагає подолати відмову користувача від технології, оскільки її використання не вимагає нових навичок. Важливим і практичним аспектом цих проблем є те, що голосовий інтерфейс є невід'ємною частиною створення комфортних умов для проживання пацієнтів із захворюваннями опорно-рухового апарату та фахівців, які не можуть користуватися стандартними інструментами. Через професійне захворювання, травму або каліцтво. У процесі реалізації так званої концепції такі системи з часом стануть частиною повсякденного життя «Розумний будинок». Сучасність тематики досліджень залежить від стрімкого розвитку ринку програмного забезпечення для розпізнавання мовлення, який охоплює практично всі сфери нашого життя. Згідно з останнім звітом, світовий ринок цих технологій сьогодні оцінюється приблизно в 900 мільйонів доларів США і зростає приблизно на 28% щорічно. У зв'язку з вищевикладеним, можливість створення голосового інтерфейсу та системи для професіоналів надає такі інструменти. Враховуючи неможливість миттєвого переходу та необхідність адаптації до нових інструментів і засобів, стає очевидною необхідність інтеграції з існуючими системами. На практиці вирішити такі проблеми і створити необхідні системи, які можна інтегрувати, непросто. На даний момент не всі завдання розробки голосового інтерфейсу можна вважати вирішеними. Розробка програмного забезпечення голосового інтерфейсу складна і складна. У багатьох випадках вона володіє знаннями у великій кількості предметних областей. До них зазвичай входять інформатика, лінгвістика і навіть психологія. Значення магістерської роботи полягає в тому, щоб з'ясувати недоліки існуючих рішень інтелектуальної системи аналізу голосових



команд, створити систему, яка забезпечує більш інтуїтивно зрозумілу та просту взаємодію між користувачами розумного будинку та роботами-компаньйонами, а також використати інструменти розпізнавання голосу для перетворення аудіо в текст і текст обробляти та інтерпретувати повідомлення до команд завдання. Система буде простою у використанні, за розумною ціною, простою в обслуговуванні та розширенні.

Метою роботи є підвищення ефективності та спрощення взаємодії між користувачами та розумними пристроями.

Об'єктом дослідження є безпосередньо архітектура системи розпізнавання голосових команд.

Предметом дослідження є технологія, яка буде використана для реалізації системи розпізнавання голосових команд.

Практичне значення отриманих у роботі результатів полягає в розробленні вдосконаленої технології для реалізації архітектури системи розпізнавання та керування голосовими командами для комп'ютерних систем.

Новизна роботи полягає в тому, що запропонована система може бути використана в різних сферах для полегшення роботи за комп'ютером. Програмна реалізація системи розпізнавання голосових команд, розроблена в цій роботі, може бути використана для створення розумних будинків або автомобілів.

Апробація. Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2022)».

Публікації: Ірина Олександрівна Майданевич, Людмила Дмитрівна Дихніч «Розробка еgr-застосунку для голосового управління типовими операціями», ВВПК САМН «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2021. URL:

<https://conferences.vntu.edu.ua/index.php/mn/mn2022/schedConf/presentations>

# 1 АНАЛІЗ ПРОБЛЕМИ СТВОРЕННЯ ЗАСТОСУНКІВ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ ТА ІСНУЮЧІ АНАЛОГІЇ

## 1.1 Пошук перспективного сегменту застосування застосунків для голосового управління

Як говорить Вікіпедія: «Планування ресурсів підприємства (ERP-система) (англ. Enterprise Resource Planning System — Система планування ресурсів підприємства) — корпоративна інформаційна система (КІС), призначена для автоматизації обліку й керування. Зазвичай ERP-системи будуються за модульним принципом і в тому або іншому ступені охоплюють всі ключові процеси діяльності компанії». Відштовхуючись від цього визначення далі досліджуватимемо спрощені версії таких систем, орієнтовані на приватне використання (наприклад, системи класу «розумний дім»). Система «розумний дім», зокрема, дозволяє планувати і управляти ресурсами приватної оселі (опалення, витрати води, електроенергії тощо), а також здійснювати автоматизовану оплату цих послуг. В реаліях сучасної України це ледь не найактуальніша проблема для переважної більшості громадян. Втім, сучасні системи класу «розумний дім» керуються переважно з сенсорних панелей, що є не завжди зручним (наприклад, коли «руки зайняті»), отже, розвинений голосовий інтерфейс в таких системах є не тільки доцільним, а й необхідним. Таким чином, розробка елемента голосового інтересу для ERP-подібної системи «розумний дім» є очевидно актуальною.

Історія розумних будинків почалася в дев'ятнадцятому столітті, коли з'явилася перша система життєзабезпечення. Усі існуючі сьогодні технології базуються на розробках ХХ століття. У ході наукового прогресу ці технології також повільно розвиваються. Оснащення будинку необхідним технічним обладнанням для додання будинку сучасного вигляду: кухня з вбудованою технікою, у тому числі газова або електрична плита та холодильник, ванна кімната та туалет з сучасною сантехнікою тощо. Подальше вдосконалення

домашнього обладнання призвело до впровадження телефону, радіо і телебачення. Цей процес відбувався одночасно зі створенням майбутніх проектів житла, і уявлення про нього зазнало принципових змін. Поняття «розумний будинок» походить від англійського терміну smartbuilding [1]. Через неправильний переклад німецькою це поняття називають «розумним будинком», що часто вводить в оману. Це питання раціонального використання «будиночка» агрегату, і не впливає на поняття «автомобіль». Основний акцент робиться на його інтерактивності.

Систему «розумний будинок» можна трактувати як ERP-подібну систему. Ця інтерактивна система складається з кількох різних мереж, таких як електромережі, телефонні та телевізійні мережі. В даний час відбувається процес об'єднання мережі для створення найкращого інтерфейсу між мережею та кінцевим пристроєм, а головне – створення людино-машинного інтерфейсу у вигляді графічного інтерфейсу користувача, сенсорного екрану або лише голосового або розпізнавання жестів. Основні функції розумного будинку включають управління наступними системами:

- Інфраструктура життєзабезпечення (управління освітленням, управління опаленням, управління мікрокліматом);
- Система безпеки (датчики руху, сигналізація, блокування дверей і вікон;
- Побутова техніка (управління холодильником, чайником/кавоваркою, підтримка функцій контролю та управління через Інтернет);
- Енергетика (контроль альтернативної енергетики);
- Технічне обслуговування (включаючи відправку повідомлень і листів), моніторинг та управління (включаючи взаємодію через Інтернет).

Фахівці зазначають, що більшість користувачів систем «розумного дому» віддають перевагу такому способу регулювання своїх параметрів, як голосове управління. У більшості випадків причиною вибору є практичність власника, адже в функціональну сферу системи може входити будь-який елемент «розумного будинку»:

- Клімат-контроль;
- Охорона та відеоспостереження;
- Постачання води;
- Освітлення;
- Зрошення ділянки;
- Басейн;
- Жалюзі;
- Телебачення.



Рисунок 1.1 - Приклад розвиненої системи «Розумний будинок»[5]

Установка системи надає користувачам багато переваг:

- Використовуючи тільки запрограмовані або саморобні голосові команди, можна керувати практично всіма сферами життя;
- Використовуйте інноваційні технології для створення «розумного будинку», щоб зробити його надійним;
- Система голосового управління «Розумний будинок» може розпізнавати голоси та забезпечувати додаткову безпеку вашого дому;

-Прийнятна ціна.

Наприклад, ви хочете мати можливість керувати світлом у кімнаті, коли ваші руки дуже зайняті або брудні або ваші діти не можуть дістатися до вимикача. По-перше, потрібно точно знати силу світильників, які освітлюють кімнату. Для реалізації системи голосового управління «розумний будинок» знадобиться:

-Запобіжник;

-Потужні датчики для управління різними зонами освітлення (у приватних будинках деякі приміщення використовуються менше, особливо підвал);

-Обмежувач-актуальний при установці системи в дитячій (так дитина не зможе балуватися світлом);

- Мікрофон.

Їх можна встановити за допомогою вимикача. Найбільш підходящими є пристрої, менш чутливі до фонового шуму, щоб обробляти вхідні голосові команди. Також важливо визначити найкращий спосіб керування всім домашнім звуком. Будь то голосова команда чи хлопок, все залежить від середовища. Наприклад, якщо вдома є діти, вони будуть голосно і радісно спілкуватися один з одним, а краще всього вибрати керування плесканням у долоні.

## 1.2 Досвід створення голосових інтерфейсів

В даний час через широку популярність мобільних пристроїв все більше уваги привертають системи розпізнавання голосу. З огляду на обмежений інтерфейс для ручного введення, застосування голосового керування до цих пристроїв розширить їх функції та спростить використання. Через обмеження одного або кількох параметрів існуюча технологія розпізнавання зазвичай не користується популярністю. Наприклад, що дозволяє ідентифікувати більшість систем Запити користувача вимагають підключення до Інтернету (до віддаленого сервера, що обробляє запит),

оскільки вони не мають достатньої обчислювальної потужності та обмеженої пам'яті, виділеної для словника. Станом на 2012 рік ринок систем розпізнавання мовлення оцінюється в 1 мільярд доларів США, і планується його зростання. Голосова біометрична технологія користується великим попитом у криміналістичних та військових застосуваннях і є основною «рушійною силою» ринку [3].

Однією з найперших комерційних програм була Voice Navigator, Dragon NaturallySpeaking тощо. Початок використання таких процедур можна простежити з початку 1990-х років. В основному ними користуються люди з обмеженими можливостями, для яких введення великої кількості тексту є проблематичним. Перетворюючи голосову інформацію мовця в текст, ці програми дозволяють вводити текст, не використовуючи руки. Якість перекладу цих програм досить низька, але з часом вона значно покращилася. Завдяки покращеній продуктивності мобільних пристроїв є можливість використовувати технологію розпізнавання голосу в його додатках. Прикладом такого програмного забезпечення є голосові команди Microsoft. Програма надає безліч функцій голосового управління. Доступні функції: створення нових документів, введення повідомлень та відтворення аудіофайлів. У повсякденному житті ви можете знайти наступні програми розпізнавання голосу. Демонстрація програми для вирішення можливих проблем: для лікарів у поліклініках не потрібно вручну заповнювати медичну карту пацієнта, достатньо лише повідомити діагноз та дані пацієнта.

Іншим прикладом є система «розумний будинок», де використовуються голосові команди для управління електрикою в будинку, наприклад, включення або вимкнення світла, кондиціонера тощо. Так, все більше і більше в багатьох додатках на сучасних мобільних пристроях Використовуйте автоматичну систему розпізнавання мовлення. Важливо те, що дані, які використовує система, не залежать від мовця і можуть розпізнавати акустичні сигнали зі спектром людської мови. Наступним кроком у реалізації технології розпізнавання мовлення є використання

інтерфейсу SSI. Ці системи засновані на процесі обробки мови на ранніх стадіях вимови. Однак можна підкреслити істотний недолік сучасних систем розпізнавання мовлення, а саме: занадто висока чутливість до шуму, тобто необхідно оперативно подавати до системи розпізнавання досить чітку і зрозумілу голосову команду. Слід зазначити, що SSI заснований на методі, який передбачає використання датчиків, на які не впливає шум, як доповнення до обробленого акустичного сигналу.



Рисунок 1.2 – Шум каналу під час розпізнавання мовлення [5]

Характеристики мовлення конкретної людини (за віком, статтю), діалект, швидкість мовлення, мовленнєвий чи просодичний контекст, фоновий шум, характеристики мікрофона, напрямок мовлення тощо. Система повинна бути в змозі протистояти цим показникам. Мобільне середовище оточене шумом навколишнього середовища. На малюнку показані причини акустичних змін у мовленні.

Таблиця 1.1 - Категорії інтерференції при розпізнаванні мовлення

Категорія завад	Потенційні завади
Динамік	<ol style="list-style-type: none"> <li>1. Стать</li> <li>2. Якість голосу</li> <li>3. Діалект</li> <li>4. Стрес/емоційність</li> <li>5. Швидкість мовлення</li> </ol>
Мікрофон	<ol style="list-style-type: none"> <li>1. Електричний шум</li> <li>2. Спотворення</li> <li>3. Напрямок мовлення</li> </ol>
Шум	<ol style="list-style-type: none"> <li>1. Інші динаміки</li> <li>2. Фоновий шум</li> <li>3. Ревербації</li> </ol>
Канал зв'язку	<ol style="list-style-type: none"> <li>1. Спотворення</li> <li>2. Шум</li> <li>3. Ехо</li> <li>4. Випадіння</li> </ol>

Існує п'ять основних методів розпізнавання мов:

1. Розпізнайте одну команду, тому вам потрібно вимовити слова та фрази, перш ніж їх можна буде розпізнати. Якість розпізнавання цього методу обмежена.

2. Розпізнавання граматики - таким чином, розпізнавання здійснюється за допомогою фраз, які відповідають певному набору правил. Для визначення граматики тут використовується стандартна мова XML, а обмін даними між системою розпізнавання та програмою здійснюється відповідно до протоколу MRCP.

3. Пошук ключових слів у комбінованому голосовому потоці – у цьому способі розпізнавання голосу він виконується окремою частиною мови. Мова тут, швидше за все, відповідає набору деяких правил, і вона також відповідає певному правилу. У цьому методі немає необхідності перекладати весь текст це включає лише ті області, які містять вказане слово чи фразу.



4. Розпізнавання комбінованого мовлення у великих словниках – у цьому методі фрази перетворюються буквально в текст. Надійність цього методу досить висока.

5. Використання нервової системи для розпізнавання мовлення - досить складний метод, але є можливість створення учнів і систем самонавчання на основі нейронних мереж, що є важливою передумовою їх використання при розпізнаванні (і синтезі) мовлення.

Ідея використання голосового керування для управління системою «розумного дому» не нова. У великій кількості комерційних систем він певним чином використовується. Наприклад, Siri - це зручний голосовий інтерфейс, інтегрований в iPhone. Він знає багато мов, і ця кількість з кожним роком зростає. Він має широкий спектр функцій, але через закритий протокол його неможливо безпосередньо використовувати як контролер для голосового керування розумними пристроями. Усвідомлюючи необхідність таких функцій, компанія самостійно розробила додаток HomeKit, який є «тунелем» між Siri та розумним середовищем [4].



Рисунок 1.3 - Інтерфейс Apple HomeKit та Asus Zenbo [7]

Системи зі схожими функціями та завданнями, які вони вирішують, включають:

-LG CLOi,

-серія роботів-помічників з різними функціями.

За допомогою голосового керування ви можете керувати іншими розумними пристроями вдома.

-Допоміжний кінцевий пристрій ASUS Zenbo може пересуватися по кімнаті, розпізнавати обличчя, шукати інформацію в Інтернеті, а найголовніше для нас – це голосове керування.

-Omote TrueSmart Robot, робот-помічник на базі голосового помічника Amazon Alexa, дозволяє знаходити інформацію в Інтернеті та спростити управління системою «розумний будинок». Оснащений 5 мікрофонами.

Таблиця 1.2 - Аналіз аналогів застосунку для голосового управління розумним будинком

Назва	Переваги	Недоліки
Asus Zenbo	<ol style="list-style-type: none"> <li>1. Самостійно пересувається по приміщенню.</li> <li>2. Розпізнавання обличчя.</li> <li>3. Має здатність пошуку інформації у мережі.</li> <li>4. Має голосове управління.</li> </ol>	<ol style="list-style-type: none"> <li>1. Відсутність маніпуляторів.</li> <li>2. Має обмежену можливість у пересуванні.</li> </ol>
LG CLOi	<ol style="list-style-type: none"> <li>1. Має голосове управління.</li> <li>2. Керування іншими розумними пристроями.</li> <li>3. Має функцію нотаток.</li> </ol>	<ol style="list-style-type: none"> <li>1. Не пересувається самостійно.</li> </ol>
Omote TrueSmart Robot	<ol style="list-style-type: none"> <li>1. Має функцію пошуку інформації за допомогою «Алекси».</li> <li>2. Керування системою «розумний дім».</li> <li>3. Оснащений 5ма мікрофонами.</li> </ol>	<ol style="list-style-type: none"> <li>1. Пропріетарне програмне забезпечення «Алекса» від Amazon.</li> </ol>
Cortana	Схожий функціонал. Різниця лише в алгоритмах, методиках навчання нейронних мереж, тощо.	
Siri		
Google Asistant		
Amazon Alexxa		

Основним недоліком цих рішень є фірмовий характер і те, що вони пов'язані з реалізацією тієї чи іншої технології. Перевагою нашої системи буде відкритість і можливість розширення та подальшого вдосконалення.

Давайте зосередимося на готових рішеннях для інструментів розпізнавання мовлення, які можна інтегрувати в більш складні системи. Концепція SpeechRecognition базується на технології DeepSpeech з інтегрованою архітектурою від Baidu і платформою TensorFlow від Google

для машинного навчання. Крім моделей розпізнавання мовлення, колекція також включає модулі Python і NodeJS, системи, що спрощують інтеграцію програмних продуктів, і інструменти командного рядка. Основними особливостями DeepSpeech є низькі вимоги до системних ресурсів (навіть запущених на Raspberry Pi) і високоякісне розпізнавання під зовнішнім шумом. За словами Mozilla, ідея проекту полягає у створенні джерела мовлення з відкритим вихідним кодом паралельно з новою моделлю розпізнавання мовлення, яка є достатньо великою для використання у великих проектах.

Проаналізуємо відомі інтерфейси для розпізнавання мовлення.

Програма розпізнавання мовлення BitVoicer. Дозвольте простим пристроям із низькою потужністю обробки стати голосом. З цією метою BitVoicer використовує обчислювальну потужність ПК для аналізу звукових потоків, визначення пропозицій, які існують у цих потоках, і надсилання команд на підключений мікроконтролер.

Cloud Speech-to-Text Open API для обробки аудіофайлів розпізнавання мовлення від Google. Кількість безкоштовних запитів обмежена. Поверніть об'єкт json з інформацією про точність розпізнавання та сам текст відповідно.

Розпізнавання мовлення веб-інтерфейс Mozilla для розпізнавання мовлення. Перевага в тому, що система безкоштовна і може бути інтегрована в розширення браузера.

Відкритий API SpeechKit Яндекс. Однією з переваг Cloud Speech-to-Text є більше можливостей, а кількість безкоштовних запитів необмежена.

CMU Sphinx Sphinx CMU (скорочено Sphinx) - це загальний термін, що описує групу систем розпізнавання мовлення, розроблених Університетом Карнегі-Меллона. Він включає в себе серію програм розпізнавання мовлення (Sphinx 2-4) і тренажер акустичної моделі (SphinxTrain).

Інструмент розпізнавання мовлення від Microsoft. Використовується в Windows.

Snowboy Hotword Detection - це механізм виявлення «гарячих слів», який можна запуснути локально. В основному він слухає ключову фразу (наприклад, «Привіт, Олександр»), а потім встановлює прапорець, що фраза була виявлена. Ця функція дуже корисна. Оскільки це позбавляє пристрій від необхідності постійно надсилати все, що він чує, на сервер Amazon LEX, щоб перевірити, чи має він відповідати. Замість цього він буде слухати свої ключові фрази і вважати, що все, що буде сказано після цього, для неї, поки не настане період очікування або не буде виконаний поточний намір.

Набір інструментів ідентифікації мови Kaldi написаний на C++ і ліцензований за ліцензією Apache v2.0. Kaldi призначений для використання дослідниками розпізнавання мовлення. Механізм розпізнавання мовлення Julius, включаючи високопродуктивний двопрхідний великий безперервний словник розпізнавання мовлення (LVCSR), програмне забезпечення для декодування мовлення.

Wit Speech Wit Speech дозволяє розробникам легко створювати програми та пристрої, за допомогою яких можна розмовляти або надсилати текст.

Відкрита й розширювана платформа природної мови Wit.ai Wit.ai вивчає людську мову під час кожної взаємодії та використовує спільноту: вивчений вміст ділиться між розробниками.

VaxForge VaxForge - це вільний інструмент для мовлення та модель акустичного сховища системи розпізнавання мовлення з відкритим вихідним кодом. Він був створений для збору транскрибованого мовлення, щоб створити безкоштовний мовний інструмент GPL для механізму розпізнавання мови з відкритим вихідним кодом. Аудіофайли трансляції будуть скомпільовані в акустичні моделі для використання з відкритими механізмами розпізнавання мовлення, такими як Julius, ISIP, Sphinx і НТК.

Усі розглянуті інструменти та програми мають різні функції, і я використовую різні алгоритми та методи для обробки інформації та розпізнавання мовлення. Більшість з них засновані на дуже складних

концепціях і вимагають від розробників глибокого розуміння предметної області. Майже всі рішення призначені для великих словників, що не дуже важливий аспект для нашої системи, оскільки функції помічника в системі «розумний будинок» дуже обмежені, а кількість типових завдань не дуже велика.

Також варта уваги зворотна до розпізнавання мови задача - Text to speech. Текст в мовлення є артефактом мови. Комп'ютеризована система, яка використовується для цієї мети, називається мовним комп'ютером або синтезатором мовлення, який може бути реалізований в програмних або апаратних продуктах. Проаналізуємо відомі інтерфейси Text to speech.

Speech Synthesis. Синтез мовлення, безкоштовне програмне забезпечення для багатомовного синтезу мов, що працює на кількох платформах, забезпечує відкриту архітектуру для дослідження синтезу мовлення.

Інструмент для відтворення голосу та тексту від Яндекс SpeechKit. Gusspeech Toolkit може легко створювати високоякісні комп'ютерні мови, розробляти нові мовні бази даних і створювати керовані голосові стимули для психофізичних експериментів.

eSpeenkNG - компактний синтезатор з відкритим кодом для Windows та довільних інших операційних середовищ. Він використовує формантний синтез, який робить багато мов меншими. Багато програм підтримки мов мають власні файли правил зворотного зв'язку.

IBM TTS використовує Watson Text-to Speech, ви можете генерувати людський голос із написаного тексту. Покращуйте досвід і взаємодію з клієнтами, взаємодіючи з користувачами різними мовами та тонами. Збільште доступність вмісту для користувачів із різними функціями, надайте опції аудіо, щоб уникнути відволікань, або автоматизуйте взаємодію з обслуговуванням клієнтів для підвищення ефективності.

Інструмент Microsoft Speech для розробників програм Microsoft.

Інструмент Apple TTS для розробників додатків операційної системи iOS.

Бібліотека gTTS від Google для перетворення тексту в мовлення

Бібліотека Pytsx для мови програмування Python для textToSpeech.

Система текстового мовлення перетворює звичайну текстову мову в аудіоформат, інші системи відтворюють мовний символ у формі фонетичної транскрипції. Синтетичну мову можна створити шляхом поєднання записаних фрагментів мовлення, що зберігаються в базі даних. Система відрізняється розміром збереженої мовної одиниці. Система зберігає програвачі записів і підсилювачі, що дозволяє отримати більш широкий діапазон звуків, але без чіткості. Для конкретних цілей розмір повних слів або речень може забезпечити кращу якість звуку. Крім того, синтезатор може включати моделі вокального тракту або інші особливості людського голосу для повного відтворення «синтезованої» мови.

Сьогодні ринок «розумного дому» розвивається дуже стрімко. Голосове управління більше не є несподіванкою. Кожен сучасний смартфон має своє рішення, яке може забезпечити розпізнавання та аналіз голосу, а також копіювання тексту. Усі компанії також розуміють найбільш актуальні області застосування технології. Apple вже має HomeKit, програмне забезпечення, яке діє як канал між розумними пристроями та Siri. Як ми бачили, вже існує велика кількість готових рішень у вигляді допоміжних роботів з інтегрованими системами розпізнавання мовлення, а також відповідних підсистем, модулів чи наборів інструментів, які можна використовувати. Звичайно, основним недоліком є те, що всі рішення платні, а безкоштовні мають обмежені функції або закриті системи. Після аналізу аналогів стає зрозуміло, що запропонована нами система має право на життя і може бути новою. Нам також потрібен модуль для перетворення тексту назад у аудіофайл. Система текстового мовлення перетворює мову звичайного тексту в аудіоформат. Інші системи символічно відтворюють мовлення у вигляді фонетичних символів у мовленні. Система захисту фону

дозволяє отримати більш широкий діапазон звуків, але недостатньо чіткого. Для конкретних цілей розмір повних слів або речень може забезпечити краще звучання.



## 2 АНАЛІЗ ТЕХНОЛОГІЇ ЗАСТОСУНКУ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ

### 2.1 Структурна і функціонал застосунку для голосового управління

Алгоритм і інтерпретація системи інтелектуального аналізу голосових команд для подальшого планування дій робота-помічника повинні описувати повний цикл взаємодії з користувачем. Починаючи зі статичного режиму робота, у нашому прикладі ми будемо чекати кодового слова як сигналу для початку обробки мовлення та порівняти результат з базою типових команд. Алгоритм поведінки закінчується в системі, де слово неможливо розшифрувати чи неможливо знайти належним чином. У разі замовлення. Алгоритм повинен бути циклічним, тому що Кінцевий пристрій повинен повернутися в режим очікування після виконання завдання. Алгоритм також повинен бути стійким до помилок, оскільки в складній і великій системі, як «розумний будинок», існує багато сценаріїв, за яких нерозуміння голосових команд може призвести до небажаних наслідків або тупикових ситуацій та невизначених ситуацій. Спілкування також має бути інтуїтивно зрозумілим для користувачів. Необхідно описати сценарії поведінки робота в усіх можливих ситуаціях. Можливо, допоміжний кінцевий пристрій вже виконує завдання. У цьому випадку він повинен сповістити користувача і повернутися до завдання.

Якщо після інтерпретації та порівняння голосової команди з існуючою командою не знайдено відповідності - кінцевий пристрій повинен повідомити про це користувача і не виходити з режиму прослуховування команди, поки кодове слово не чекає роз'яснення команди. Якщо користувач з будь-якої причини не повторює команду - поверніться в режим очікування. Також буде використовуватися система Text to Speesh, оскільки кінцевий пристрій повинен багато спілкуватися з користувачем і інформувати його про поточний статус, виконання команди або інтерпретацію помилок.

Основними компонентами та поясненнями системи інтелектуального аналізу голосових команд будуть:

1. Сам інструмент для прослуховування та обробки голосових команд.
2. База даних команд.
3. Веб-інтерфейс.

Потрібно вибрати метод розпізнавання мовлення, мову, реалізовану технологією інтерфейсу, і базу даних словника робота.

1. Система чекає на ключові слова.
2. Система розпізнає ключове слово.
3. Система нагадує користувачеві про зміну статусу звуковим сигналом під час очікування команди.
4. Система чекає команди.
5. Система розпізнає слово чи фразу.
6. Система обробляє та отримує текст команди.
7. Команда знайдена.
8. Кінцевий пристрій продовжує виконувати завдання, відповідні команді.
9. Система повідомляє користувача про початок виконання завдання та відповідну зміну поточного стану.
10. Кінцевий пристрій виконує команду.
11. Система повідомляє про стан очікування завдання та відповідне ключове слово return.

Сценарії для неправильного пошуку ключових слів. Ключове слово має бути унікальним, але якщо система знаходить систему в звичайній розмові, система повинна сповістити користувача.

У цьому випадку, якщо після розпізнавання ключового слова більше не буде виявлено звукових сигналів, воно має автоматично перейти в режим очікування.

1. Система розпізнає ключове слово.

2. Система пропонує користувачеві перейти в режим очікування команди.

3. Через 10 секунд, якщо не буде отримано жодних інструкцій, система автоматично повернеться в режим очікування.

4. Система повідомляє користувача про перехід у сплячий режим.

Сценарій, коли відповідна команда не знайдена в базі даних. Можливо, система не зможе знайти команду з відповідною назвою. У цьому випадку система повинна сповістити користувача. Основна причина полягає в тому, що система потім повертається до режиму моніторингу команд, а не ключових слів. У другому випадку інтерфейс стає менш інтуїтивно зрозумілим і складним у використанні.

1. Система чекає команд.

2. Система отримує команду.

3. Система не знайшла відповідну команду в базі даних.

4. Повідомити користувачів.

5. Повернення до стану групи слухачів.

6. Повідомляти користувачів про зміни статусу.

Очевидно, що при заповненні бази даних слід уникати використання команд зі схожими іменами. Як ми вже зрозуміли, система розпізнавання мовлення дуже складна і складна, будь-які шуми або проблеми зі словом користувача призведуть до небажаних наслідків. Немає необхідності ще більше ускладнювати цей процес. Якщо система не може розпізнати слова, це одна справа, але якщо система може неправильно їх розпізнати і виконати дії, які користувач не хоче виконувати, це зовсім інша справа. Друге набагато гірше. Цю проблему можна вирішити багатьма способами. Якщо отриманий системою результат недостатньо точний - вона може сповістити користувача і запитати його точне значення. Навіть такі подібні варіанти можна надати.

Але це ускладнює і може дратувати користувачів. Більш ефективний варіант - просто уникнути цієї ситуації в базі даних. Після того, як система розпізнає свій власний виклик, вона видасть користувачеві голосове

повідомлення, яке вказує, що він слухає його команду. Потім система почне контролювати команди хоста. Якщо команда, надана хостом, несумісна з будь-якою відомою робочою командою після розпізнавання, чи то через помилку хоста чи через багато шуму, кінцевий пристрій повідомить, що не може зрозуміти запит хоста. Якщо команда була розпізнана, Кінцевий пристрій повідомляє власника, що зв'язок пройшла успішно, і продовжує виконувати операцію, а потім повертається в стан прослуховування. Цей набір команд буде визначено, але його можна легко розширити, якщо це необхідно або виникає потреба. Клієнт (робот) і сервер (система) будуть спілкуватися через API. Нижче наведена спрощена схема цієї конструкції.

З функціональної точки зору користувачі хочуть бачити певний набір інтерактивних функцій і доступних команд. Звичайно, до внутрішніх можливостей взаємодії системи відноситься голосова взаємодія та інтеракція через web-інтерфейс. Той має більше налаштувань і опцій, а на додачу дозволяє стежити за статистикою і статусом робота. Серед доступних команд ви можете знайти наступний список найбільш використовуваних і найпопулярніших:

1. Знайдіть інформацію про X в Інтернеті.
2. Дізнайтеся погоду X.
3. Повідомте про свій статус, ситуацію та поточні дії.
4. Почніть вживати заходів щодо розумного дому (почніть прибирати, готувати тощо).
5. Взаємодія з іншими розумними пристроями.

Розглянемо структуру кожного окремого компонента. Компонент Voice Listening and Interpretation буде модульною програмою, яка матиме інтерфейс REST з точками доступу у вигляді URL-посилань, які відповідатимуть на запити (з параметрами чи без), інформацію про процес або форму необхідної інформації. Повідомлення повертається. Позаду буде модуль, який буде реалізовувати основну логіку розпізнавання мовлення. Створіть екземпляр класу розпізнавання голосу, який містить екземпляри аудіодрайвера,

розпізнавання голосу, мікрофона та класу голосового програвача, а також статус прослуховування, поточну мову та результат остаточного розпізнавання.

Основною перевагою цієї системи є її модульність і цілісність: веб-інтерфейс реалізований окремим модулем, який отримує інформацію від окремих, незалежних циклових процесів, що працюють паралельно. Незалежність від модулів покращить підтримку системи. Він має метод ініціалізації вищезгаданих параметрів, а також метод для запуску циклічного процесу ідентифікації та повернення результату ідентифікації. Модуль запускається окремим процесом, записує результати кожної ітерації та відтворює голосове повідомлення користувачеві на основі результатів.

Модуль веб-інтерфейса забезпечує за взаємодію з системою на щаблі веб-браузера, отримання інформації, записаної розпізнавачем голосу, а також відображення процесу інтерпретації та результатів, статусу розпізнавача та іншої інформації в режимі реального часу. Він також працює як окремий процес, має адресу доступу і не залежить від процесу розпізнавання. Подавати команди можна не тільки у вигляді голосу, але і через веб-інтерфейс. Це може вирішити потенційну невизначеність і покращити доступність системи.

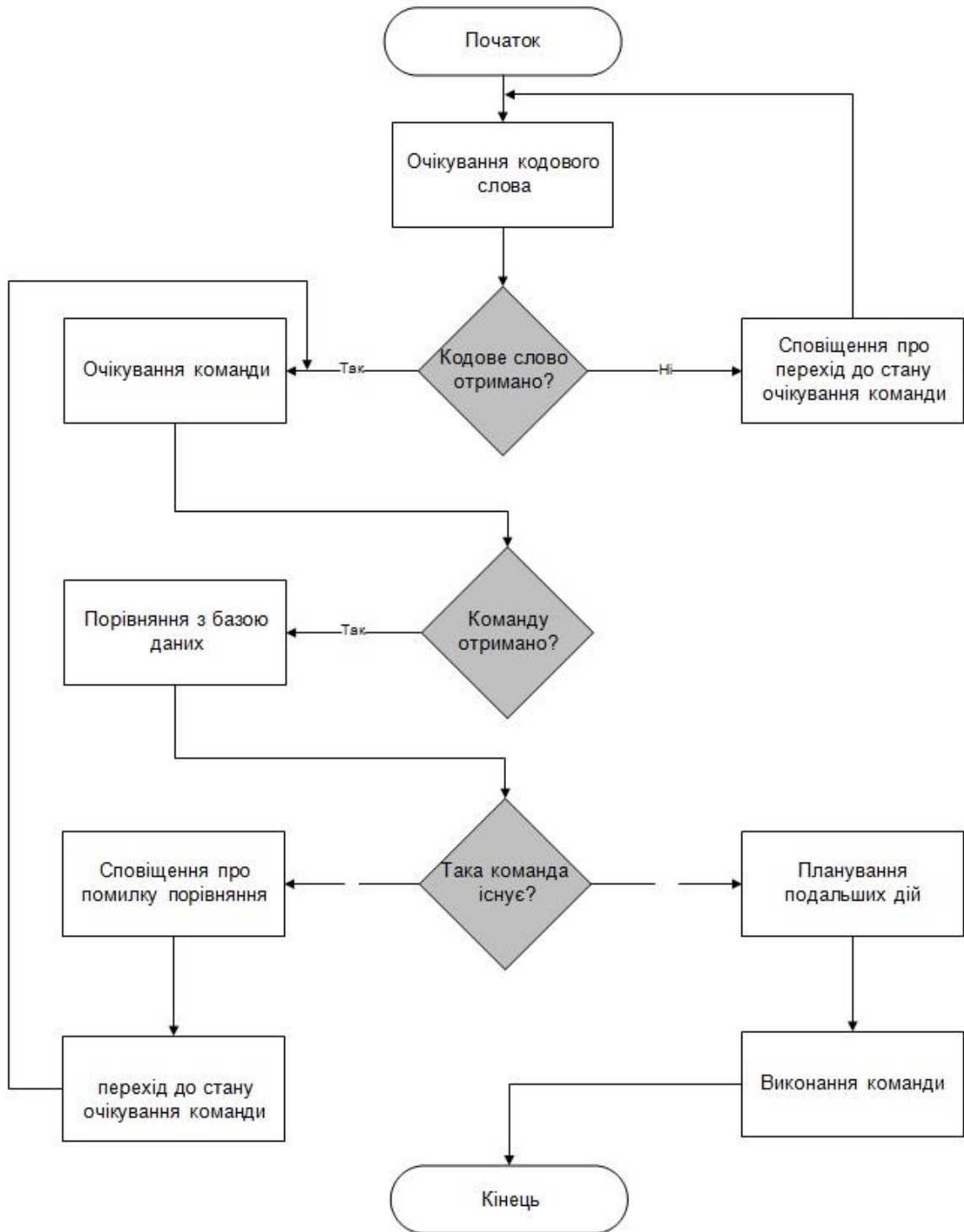


Рисунок 2.1 - Алгоритм ERP-подібного застосунку для голосового управління

## 2.2 Класифікація систем голосового інтерфейсу

Розглядаючи класифікацію системи розпізнавання мовлення, слід зазначити, що її можна класифікувати за різними параметрами. Відповідно до літератури, системи розпізнавання мови можна розділити на такі категорії:

- Залежить від розміру словника: система розпізнавання мови з обмеженим набором слів; велика система словників;

- Залежить від прив'язки до динаміка: системно-залежний і незалежний від динаміка;

- Залежить від типу розпізнавання мовлення: використовувати систему, яка поєднує або розділяє мовлення;

- За призначенням системи можуть бути призначені система диктанту та система команд;

- За алгоритмами, що використовуються в системі розпізнавання, розрізняють: нейронну мережу, приховану марківську модель, динамічне програмування;

- Залежить від типу використовуваної в системі структурної одиниці (можуть використовуватися слова, фонемі, ектопії, словосполучення);

- За принципом розрізнення структурних одиниць системи розпізнавання її можна поділити на систему розпізнавання образів і систему відбору елементів словника.

Розглядаючи систему розпізнавання мови, слід зазначити, що існуючі системи зазвичай використовують два принципово різних методи:

- Аналіз голосових позначок;

- Проаналізувати елементи словникового значення.

Розпізнавання голосових позначок – це розпізнавання фрагментів голосу з попередньо записаних зразків. Цей метод можна використовувати лише в простих системах розпізнавання мовлення для виконання оригінальних записаних голосових команд [5]. Розпізнавання лексичних елементів є більш складним методом. Ці методи передбачають виділення голосового потоку окремих елементів словника (тобто фонем і алофонів).

Надалі виділені елементи лексики будуть об'єднані в склади та морфеми. Тому цей метод використовується в складних системах розпізнавання мовлення.

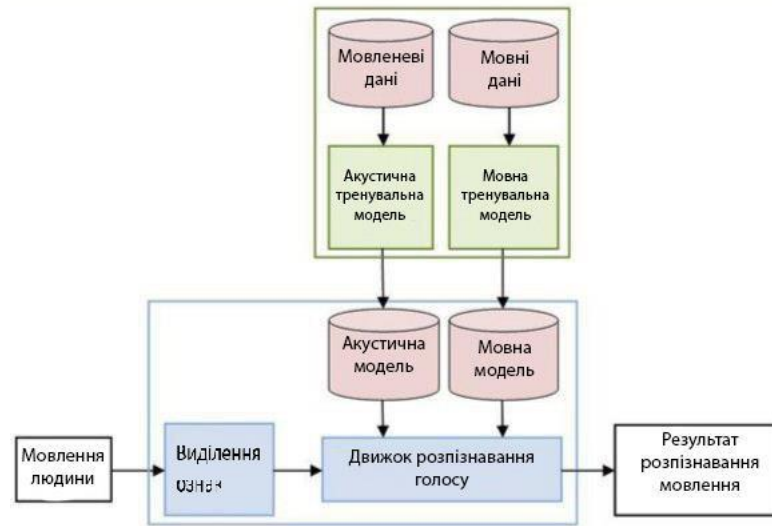


Рисунок 2.2 - Архітектура ERP- подібного застосунку для голосового управління

### 2.3 Характеристика технологій для розпізнавання усного мовлення

Категорія системи, яка не залежить від диктора, включає системи, які працюють незалежно від того, хто виступає в ролі мовця. системні дані здатність розпізнавати мову будь-якого мовця без попередньої підготовки. Категорії систем, що залежать від динаміка, включають системи, які вимагають попередньої підготовки і в процесі налаштовуються на конкретного мовця [6]. При заміні динаміків в системі такого типу необхідна повна переналаштування. Тому для створення будь-якого типу системи розпізнавання мовлення, що використовується в промислових масштабах, необхідний багаторічний досвід практичного застосування різноманітних мовленнєвих технологій. Для нашої конкретної ситуації для нас важливо мати незалежну від динаміка систему, оскільки вона повинна бути легко інтегрована в підсистему та розпізнавати голоси різних користувачів.



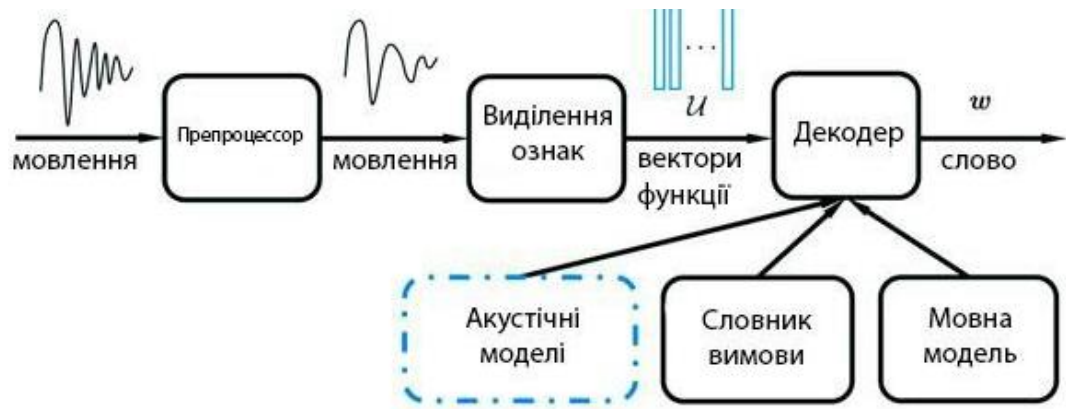


Рисунок 2.3 - Схема технології для розпізнавання усного мовлення

Методи розпізнавання мовлення мають такі відмінності:

1) Розпізнавання образів. Цей метод використовується для стандартних мобільних телефонів. У цьому випадку система розпізнавання голосу використовується з метою прискореного набору контактів абонента, що викликається, та мобільного телефону. Ця система працює так. Якщо вам потрібно ввести новий контакт у свій блокнот, система надасть вам голосову позначку, щоб ідентифікувати цей контакт. Зазвичай це прізвище та ім'я абонента, Треба чітко зазначити, що, можливо, для того, щоб правильно запровадити цю операцію, її потрібно зробити двічі-тричі. Потім, якщо вам потрібно зателефонувати користувачеві, активуйте команду голосового набору, натиснувши клавішу, і вимовте її за допомогою голосової позначки. Голосова позначка вибирає користувача і встановлює з ним зв'язок [7]. Слід відзначити, що голосове управління впроваджено не тільки в стільникових телефонах, але й в інших пристроях, наприклад, на комп'ютерних клавіатурах. При організації голосового керування клавіатурою в останню вбудовується мікрофон, а голосові позначки призначаються різним клавішам або комбінаціям клавіш. Використання такої клавіатури дуже зручно для людей з обмеженими можливостями, а за допомогою такої клавіатури можна прискорити введення інформації. Однак, якщо в кімнаті присутні шумові компоненти, якість розпізнавання голосових позначок буде значно знижена.

2) Попередній процесор акустичного сигналу. Шум і зовнішні сигнали можуть бути великою проблемою. Це вимагає попереднього аналізу та обробки акустичних сигналів. У цьому процесі буде видалено спектр за межами людського мовного спектра. Цей процес здійснюється за допомогою аналогової та цифрової фільтрації. Потім перетворіть сигнал у цифровий формат [8]. Оскільки всі ці процеси можуть виконуватися центральним процесором, система розпізнавання мовлення не потребує спеціального обладнання або апаратних засобів. Нормалізація вхідного акустичного сигналу є важливим етапом роботи препроцесора. Таким чином можна підвищити точність розпізнавання мовлення, а також уникнути проблем із гучністю. Хоча нормалізація вхідного акустичного сигналу може вирішити проблему, в деяких випадках може з'являтися шум. Недостатній частотний спектр або недостатня гучність призведе до неповної інформації в сигналі.

3) Вибір символу звукового сигналу. Завдання розпізнавання мовлення може бути складним. Щоб вирішити цю проблему, потрібно розпізнати основні ознаки людської мови. Цей метод використовує спектральну мову. Першим кроком є смуговий фільтр і отримання частотного спектру акустичного сигналу. Виконайте дискретне перетворення Фур'є. Другим кроком є перетворення отриманого спектру: логарифмічне масштабування частоти та амплітуди, згладжування спектру акустичного сигналу, зворотне перетворення Фур'є. Окрім цих характеристик, слід враховувати і їх динамічні варіанти. Для цього використовують похідну за часом від основного параметра. У цьому випадку ми можемо не тільки відстежити зміну мовних параметрів, а й відстежити їх швидкість зміни [9].

4) Вибір фонем і різних фонем: Для виділення фонем і алофонів використовуються нейронні мережі та методи формування нейронних наборів. У цьому випадку навчання розрізняти мовні примітиви (фонем та алофони) може полягати в утворенні нейронних збірок, ядро яких відповідає найпоширенішій формі кожного примітиву. Формування нейронної інтеграції – це процес навчання нейронної мережі без вчителя, при якому всі

сигнали, що надходять на вхід нейронної мережі, обробляються статистично. При цьому формується ансамбль, що відповідає найбільш частому сигналу. Рідкісні сигнали запам'ятовуються пізніше, і їх потрібно підключити до механізмів уваги або інших елементів управління вищого рівня.

5) Рівень розпізнавання комбінованого мовлення. Включення розпізнавання мовлення є багаторівневим процесом. Після попередньої обробки акустичного сигналу та виділення з нього інформаційних ознак здійснюється відбір елементів мовленнєвої лексики. Це перший рівень визнання.



Рисунок 2.4 – Ієрархія рівнів розпізнавання комбінованого мовлення

Другий шар містить склади та морфеми, а третій шар містить слова, речення та інформацію. На кожному рівні сигнал кодується представником попереднього рівня. Іншими словами, склади і морфеми складаються з фонем і алофонів, слів-складів і морфем, речень та інформаційних слів [10]. При перетворенні з одного рівня на інший, окрім представлення сигналу, також

передаються деякі додаткові символи, тимчасова кореляція та зв'язок між сигналами. Збирайте сигнали попереднього рівня, більш високий рівень має багато інформації (або інших уявлень) і може переносити. Наприклад, за участю механізму уваги, управління процесами нижчого рівня. Механізм уваги використовується для навчання нейронної мережі. У разі використання цього механізму з появою в нейронній мережі невідомих зразків швидкість навчання збільшиться в рази.

б) Використовуйте нейронну мережу для розпізнавання мовлення. Одним зі стандартних підходів для голосової конверсії є використання трансформацій, що ґрунтується на GMM (модель-суміш Гаусівських розподілів) [2, 3]. Прикладом реалізації такої техніки може слугувати програмне забезпечення “sprocket” [4]. Автори статті Kazuhiro Kobayashi, Tomoki Toda описують принцип роботи такої моделі. Ідея полягає в моделюванні функції густини ймовірнісного розподілу, що складається з суміші багатьох Гаусівських розподілів (Gaussian Mixture Model). Невідомі параметри для шуканого розподілу оцінюються за допомогою методу максимізації функції правдоподібності [5]. Процес моделювання складається з декількох етапів. Під час навчання мережі з викладачем ви можете навчити мережу розпізнавати об'єкти, які належать до заздалегідь визначеного набору класів. Якщо мережа навчається без вчителя, вона може групувати об'єкти за категоріями за їх числовими параметрами [11]. Тому на основі нейронної мережі можна створити систему самонавчання. До системи самонавчання також пред'являються деякі вимоги:

- Розробка включає побудову архітектури самої системи;
- Контролюйте їхню поведінку та надалі коригуйте їхні здібності.

Розробник проектує лише функціональну частину. Він не наповнює систему інформацією або наповнює систему мінімальною інформацією. На пізніх етапах процесу навчання система отримуватиме всю інформацію самостійно. Цей метод описує необхідність зворотного зв'язку в системі. Процес навчання штучних нейронних мереж розглядається як коригування

структури та ваги зв'язків між нейронами (параметрів) з метою ефективного виконання завдань, які покладаються на мережу. Існує два основних типи досліджень: детерміновані методи та стохастичні методи.

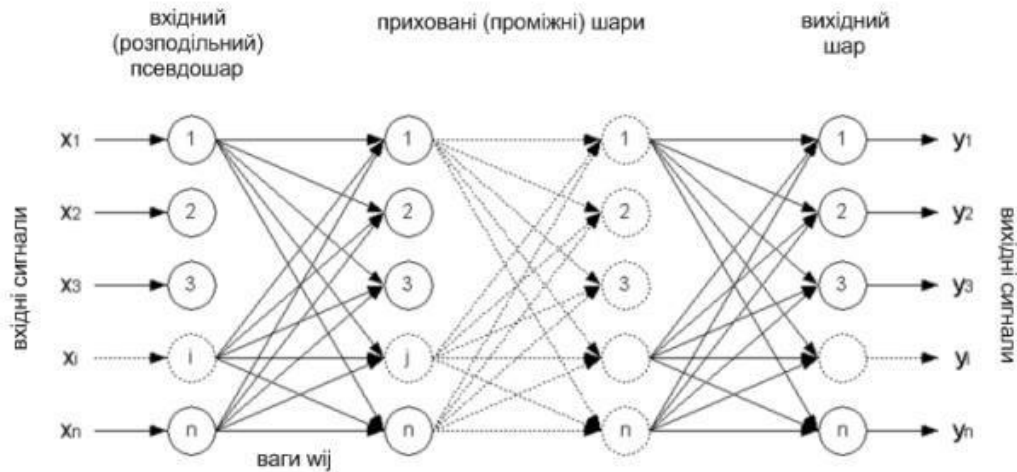


Рисунок 2.5 – Тренування нейромережі

7) Нейромережа. Слід зазначити, що як модель нейронної мережі, яка підходить для розпізнавання мовлення і не вимагає підготовки вчителя, можна вибрати самоорганізуючу карту з логотипом Кохонена. Нервова система створена для вхідного акустичного сигналу. Інша проблема - варіативність мовлення, яка вирішується шляхом статистичного усереднення. У порівнянні з класичним програмуванням, коли алгоритм вирішення задачі жорсткий, нейронна мережа дозволяє динамічно змінювати алгоритм, просто змінюючи архітектуру мережі.

8) Генетичний алгоритм. Можливість змінювати алгоритм нейронної мережі шляхом простої зміни її архітектури дозволяє вирішувати проблеми абсолютно по-новому, використовуючи так званий генетичний алгоритм [12]. При використанні генетичних алгоритмів створюються правила відбору, щоб визначити, краще чи гірше нова нейронна мережа справляється з проблемою. Крім того, визначаються правила модифікації нейронної мережі. Довго змінюючи архітектуру нейронної мережі та вибираючи архітектуру, яка

дозволяє найкращим чином вирішити проблему, ви рано чи пізно отримаєте правильне рішення. Поява генетичних алгоритмів приписують теорії еволюції. Тому можна створити таку нейронну мережу, яка раніше не досліджувалася [13].

9) Реалізація системи SAS та I/O. Система створена за технологією нейронних мереж не тільки для розпізнавання, а й для синтезу мовлення [14]. При розпізнаванні мовлення система SAS виконує введення аудіоінформації, попередню обробку, отримання енергетичного спектру та вибір мовних примітивів. Синтез мовлення витягує примітиви пам'яті з нейронної мережі, синтезує спектр (синтез частотних параметрів) і перетворює спектр у звуковий сигнал. При навчанні безперервного повторення двох вищезазначених процесів запам'ятовується пам'ять мовних примітивів у нейронній мережі. Використовуйте звукову карту або отримуйте звукові сигнали в режимі реального часу у вигляді аудіофайлів у необхідному форматі та певному кодуванні. Попередня обробка аудіо: аудіосигнал, отриманий в реальному часі, або вхідний аудіосигнал із файлу WAV попередньо обробляється в системі SAS [15].

10) Виконайте дискретне перетворення Фур'є. Результат згладжування Хеммінга піддається дискретному перетворенню Фур'є в системі SAS за алгоритмом швидкого перетворення Фур'є. В результаті отримуємо амплітудний спектр та інформацію про фазу сигналу (дійсні та уявні коефіцієнти) [16].

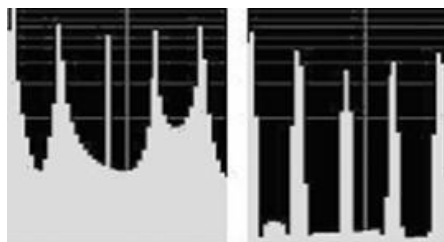


Рисунок 2.6 – Вікно згладжування Хеммінга в дії

11) Нормалізація спектру. Оскільки всі обчислення в нейронній мережі виконуються на числах з плаваючою комою, значення параметрів об'єктів, класифікованих нейронною мережею, обмежені діапазоном  $[0,0, 1,0]$ .

12) Логарифмічне стиснення спектру. Дослідження показали, що кількість інформації в різних частинах спектра різна: низькочастотна область спектра містить більше інформації, ніж високочастотна область спектра. Тому для більш економічного використання та збільшення вхідних даних нейронних мереж необхідно зменшити кількість елементів високочастотного спектру. Це означає стиснення високочастотних областей частотного спектру в частотному просторі. У системі SAS використовується найпоширеніший і найпростіший метод – логарифмічне стиснення або мел-стикс [17].

13) Використання вейвлет-перетворення. У щойно описаній системі SAS швидке перетворення Фур'є використовується для виділення синтаксичних елементів із мовлення. Однак у Фур'є є свої недоліки і недоліки, тому можна втратити інформацію про аналізований акустичний сигнал. Термін вейвлет можна перекласти як «вейвлет». Вейвлет – це новий інструмент для вирішення різноманітних задач з прикладної математики. Вейвлет-аналіз вимагає детального розуміння деяких математичних знань і не має перерахованих вище недоліків. Це дозволяє досягти хороших результатів при використанні в системі розпізнавання мовлення.

Щоб підкреслити цей факт, аналіз Фур'є в зарубіжній літературі називають одиничним спектром, тобто спектром, отриманим на основі вейвлет-перетворення (спектр шкали часу або вейвлет-спектр). Основна функція вейвлет-перетворення заснована на похідній функції Гаусса. Сам розробник відіграє важливу роль у застосуванні вейвлет-перетворення в розпізнаванні мови. Адже він повинен вибрати оптимальну функцію. Від цього залежить точність системи розпізнавання. Існує два методи розпізнавання мовлення. Перший метод реалізує розпізнавання голосових елементів через шаблони і використовується в різних типах систем голосового керування. Другий спосіб заснований на відборі Фонетика

лексичних елементів-фонем, ектопічних фонем, морфем тощо. Цей метод підходить для створення системи текстового диктанта [18]. Існують системи аналізу акустичного сигналу, які вимагають попередньої підготовки та залежать від динаміки, а є системи, які можуть працювати без попередньої підготовки, тому вони не залежать від динаміки.

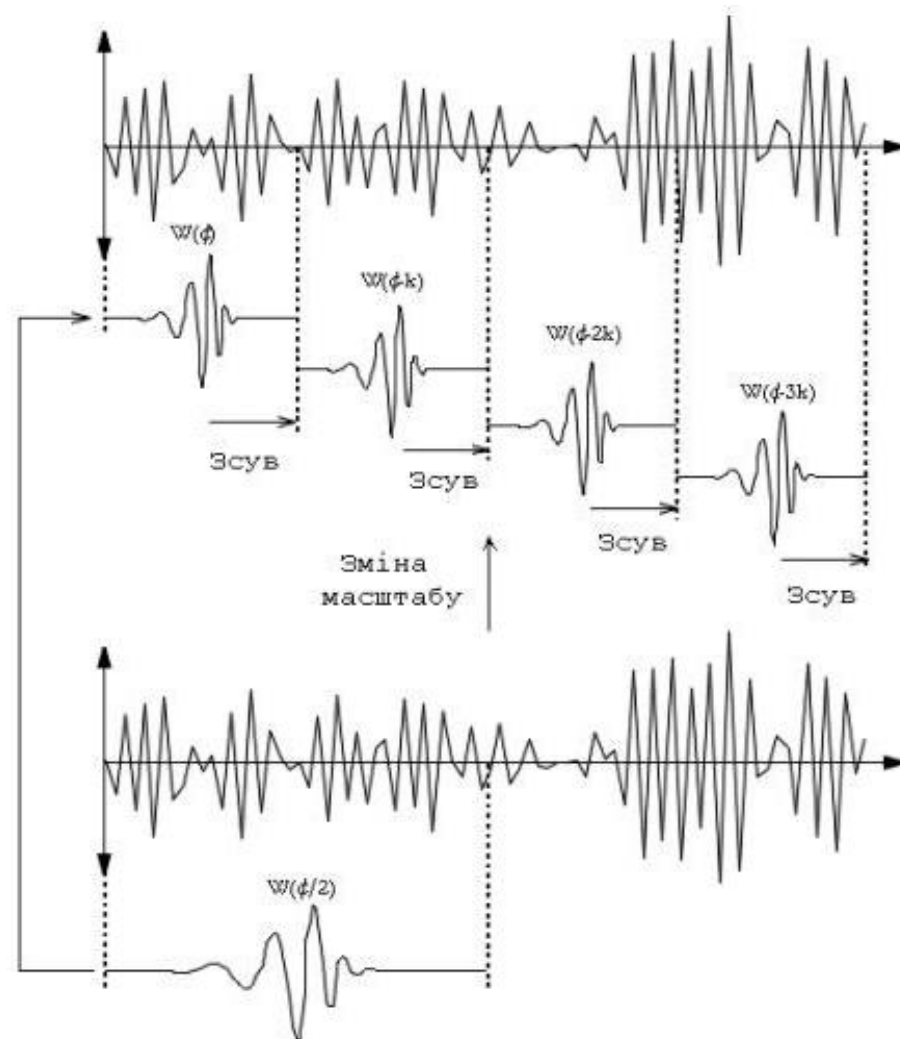


Рисунок 2.7 - Опис сигнала у спосіб зміщення вейвлету

Перш ніж приступити до виділення елементів лексики з промови, необхідно попередньо обробити акустичний сигнал. У цьому процесі з сигналу видаляються шуми, виконується частотна фільтрація та інтерпретація, а також виконується нормалізація рівня сигналу. Розглянуто два способи вилучення елементів лексики з мовлення. Перший прийом



передбачає використання дискретного перетворення Фур'є. У цьому випадку мова представляється як набір числових параметрів, оскільки нейронна мережа використовує набір таких параметрів. Другий метод виділення елементів мовного словника заснований на використанні вейвлет-перетворення [19]. У порівнянні з дискретним перетворенням Фур'є цей метод усуває втрату інформації про тимчасові характеристики обробленого сигналу. Ми помітили, що при використанні вейвлет-перетворення вхідний сигнал розкладається не на основі періодичних функцій (як у дискретному перетворенні Фур'є), а на основі функцій, що характеризують частоту та час.

Мета розпізнавання мовлення - дозволити машині «почути», «зрозуміти» і «діяти» на основі отриманої вербальної інформації. Метою автоматичної ідентифікації мовця є аналіз, виділення функцій та ідентифікація ідентифікаційної інформації мовця. Систему розпізнавання мовлення мовця можна розглядати як роботу в чотири етапи. Як бачимо, існують різні способи визначення функцій, яким властиві певні відмінності в імплементації і які використовуються для вирішення різних завдань. Всі вони знаходяться в процесі вдосконалення, а деякі проявляються як удосконалення або комбінація кількох інших методів. Після завершення етапу виділення ознак можна переходити до етапу моделювання [20].

Голосова інформація містить різні типи інформації, що відображає особистість мовця. Це інформація про мовця, пов'язана з мовним шляхом, джерелом пробудження та поведінкою. Остання інформація також вбудована в сигнал і може бути використана для ідентифікації мовця. Етап аналізу мовлення використовує рівні з відповідними кадрами для сегментації сигналу для подальшого аналізу та вибору. Метод аналізу мовлення виконується за трьома методиками: аналіз сегментації, аналіз субсегментів та аналіз суперсегментів. При аналізі сегментації розмір кадру і зміщення в діапазоні 10-30 мілісекунд використовуються для виявлення інформації про мовця, тобто для вилучення інформації про шлях голосу. Підсегментний аналіз використовує розмір стегової кістки та зміщення в діапазоні 3-5 мс. В

основному використовується для аналізу та ізоляції інформації на етапі стимулювання. Аналіз суперсегментації заснований на розпізнаванні особливостей, пов'язаних із особливостями відносин між кожним голосом у мовця.

Проблема розрізнення ознак у задачі класифікації полягає в зменшенні розмірності вхідного вектора при збереженні потужності сигналу розрізнення. З базової структури системи перевірки розпізнавання мовця ми знаємо, що кількість навчальних векторів і тестових векторів, необхідних для класифікації, збільшується з розміром цього входу, тому нам потрібна функція виділення мовного сигналу.

Метою методу моделювання є створення моделі динаміка з використанням векторів конкретних характеристик мовця. Технологія моделювання динаміка поділяється на дві категорії: розпізнавання мовця та розпізнавання мовця. Метод розпізнавання мовця автоматично визначає, хто Говоріння засноване на інтеграції окремої інформації в мовний сигнал. Розпізнавання мовця також поділяється на дві частини, одна залежить від мовця, а інша не залежить від нього. У режимі, що не залежить від мовця, затосунок має ігнорувати унікальні характеристики мовного сигналу та отримувати очікувану інформацію. З іншого боку, в режимі, пов'язаному з динаміком, машина повинна витягувати характеристики динаміка в звуковий сигнал.

Основним завданням розпізнавання особи мовця є порівняння мовленнєвого сигналу невідомої персони з базою еталонів відомих осіб. Розпізнавання мовця також можна розділити на два методи, текстовий метод і незалежний від тексту метод. Використовуючи текстовий метод, оратор промовляє ключові слова або речення з однаковим текстом для навчання та розпізнавання. Коли інший метод не покладається на конкретний текст на мові.

Алгоритм ERP- подібного застосунку має описувати весь цикл інтеракції з користувачем. Починаючи з режиму, коли Кінцевий пристрій

нерухомий, у нашому прикладі він буде чекати кодового слова як сигналу для початку обробки мовлення та порівнюватиме його з базою даних типових команд. Нарешті, алгоритм повинен бути не в змозі розшифрувати слово або знайти відповідну команду в системі. Алгоритм повинен бути циклічним, тому що Кінцевий пристрій повинен повернутися в режим очікування після виконання завдання. Алгоритм також повинен бути стійким до помилок, оскільки в складній системі, наприклад, «розумний будинок», існує багато сценаріїв, де нерозуміння голосових команд може призвести до небажаних наслідків або тупикових ситуацій та невизначених ситуацій. Спілкування також має бути інтуїтивно зрозумілим для користувача. Система повинна врахувати всі можливі помилки і правильно реагувати. Повідомляйте користувачів при переході з різних станів, щоб уникнути помилок. Крім того, команди, введені в базу даних, повинні значно відрізнятися одна від одної, щоб уникнути непотрібних тригерів.

## 3 ОПИС РОЗРОБКИ ВЛАСНОГО ЗАСТОСУНКУ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ

### 3.1 Вибір середовища та методу розроблення

Система розроблена в програмному середовищі PyCharm для розробки системи JetBrains JetBrains є розробником такого ж популярного середовища розробки Java IntelliJ IDEA. До основних методів розробки відносяться: 1. Мова програмування Python 3 2. Мікро-фреймворк сервера Flask 3. Реляційна база даних PostgreSQL 4. Мови програмування та інструменти JavaScript, HTML, CSS 5. Структура розпізнавання мовлення 6.pyttsx фреймворк 7. Фреймворк PyAudio

З різних причин мова програмування Python була обрана як найкращий вибір для реалізації цієї системи. Найважливіше те, що Python є інтерпретованою мовою, тобто він не компілюється, а виконується в одному порядку, що робить його незалежним від платформи, на якій працюють програми, створені його інструментами. Це також дозволяє рефлексувати, тобто структуру та поведінку компонентів програми під час виконання можна змінювати довільно, що вигідно нам як розробникам. Нарешті, через свою інтерпретацію програми на Python набагато легші, ніж програми на скомпільованих мовах. Python - це інтерпретована мова програмування загального призначення високого рівня. Він був створений Гвідо ван Россумом в 1991 році на основі філософії, спрямованої на покращення читабельності коду, особливо за рахунок широкого використання пробілів. Вона пропонує Розробники можуть отримати доступ до дизайнів, які забезпечують прозоре програмування малого та великого масштабу. Він підтримує велику кількість парадигм програмування, включаючи об'єктно-орієнтовану, імперативну, функціональну та процедурну, і має потужну та повну стандартну бібліотеку.

Інтерпретатор Python доступний для багатьох операційних систем. Зокрема, Python є відкритою системою з загальнодоступною моделлю

розробки, як і більшість її змінних реалізацій. Його керує Python Software Foundation. Основні концепції мови були офіційно створені наприкінці 1980-х років і реалізовані в грудні 1989 року за ініціативою Гвідо ван Россума з Національного інституту математики та інформатики Нідерландів. Він є спадкоємцем мови ABC (яка, у свою чергу, є наступником SETL), має здатність обробляти винятки та може легко взаємодіяти з операційною системою Amoeba. Гвідо ван Россум досі залишається головним автором мови. 16 жовтня 2000 р. був випущений Python 2.0, який має багато нових функцій, включаючи збірник сміття з визначенням циклу та підтримкою символів Unicode. Після виходу цієї версії розробка стала більш прозорою та підтримуваною спільнотою. Після тривалого періоду тестування Python 3 був випущений 3 грудня 2008 року. Ця версія є основною редакцією всієї мови і не повністю підтримує попередню версію. Проте багато основних функцій було перенесено на старіші версії 2.6.x та 2.7.x, а також була випущена утиліта 2to3, яка автоматично перетворює код Python 2 на Python 3.

Python - це багатопарадигмальна мова програмування. Він повністю підтримує об'єктно-орієнтоване та структуроване програмування, а багато мовних функцій підтримують функціональні та аспектно-орієнтовані стилі. (Включаючи метапрограмування та метаоб'єктно-магічні методи). Розширення підтримує багато інших парадигм, включаючи логічне програмування. Python підтримує динамічний введення і поєднує підрахунок посилань і збирач сміття з визначенням циклу керування пам'яттю. Він також підтримує динамічне розширення імен, яке пов'язує імена методів та змінних лише під час виконання програми. Архітектура мови підтримує деякі особливості функціонального програмування в стилі Lisp. Доступні методи: `filter()`, `map()`, `reduce()`, вирази списку, словники та набори, а також генератори виразів.

Python також дуже зручний як платформа для запуску легких серверів і сервісів, створення мікросервісних архітектур тощо. Це дозволяє нам легко та швидко розгорнути нашу систему та забезпечити її модульність. Крім

того, Python має велику кількість бібліотек, доступних для використання будь-кому, що дозволяє виконувати різні операції, використовувати велику кількість сервісів без доступу до API служби, а також виконувати складні обчислення або операції, пов'язані з людьми. Інтелект. Все це робить Python мовою вибору для системи.

Зокрема, ми використовуємо фреймворк сервера Flask, написаний для мови Python. Він дуже легкий і простий у використанні, з великою кількістю завантажуваних бібліотек для реалізації необхідних елементів веб-сервісу, таких як авторизація або використання бази даних. На її основі працюватимуть основні модулі всієї системи. Flask - це web-мікрофреймворк, написаний на мові Python. Його називають мікрофреймворком, оскільки для його запуску не потрібні спеціальні інструменти чи бібліотеки. Він не має рівня абстракції для баз даних, перевірки форм або будь-яких інших потужних компонентів. Зазвичай надається готовою сторонньою бібліотекою. Однак Flask підтримує велику кількість розширень, які додають функціональність додатку, як ніби вони були реалізовані безпосередньо в самому мікрокадрі. Існують також об'єктно-реляційне відображення, перевірка форми, обробка завантажень, розширення технології відкритої аутентифікації та деякі інструменти, які зазвичай пов'язані з фреймворком. Розширення оновлюються частіше, ніж саме ядро Flask. У проектах, які використовують Flask, можна згадати Pinterest, LinkedIn і саму сторінку фрейму.

Двома основними компонентами мікрофреймворка Flask є Werkzeug і Jinja2. Werkzeug - бібліотека мови програмування Python, що являє собою набір інструментів для реалізації додатків WSGI, що поширюються під ліцензією BSD. Werkzeug дозволяє створювати скрипти для оброблення запитів та ін. функцій. Його можна використовувати для створення власного програмного забезпечення як плагін і підтримує Python 2 і Python 3. Jinja2 - це шаблонний механізм мови програмування Python, подібний до веб-фреймворку Django, шаблон розраховується в окремому середовищі. Flask

підтримує такі функції, як сервер розробки та налагоджувач, вбудовані модулі для модульного тестування, обробку запитів RESTful, безпечні файли cookie та вичерпну документацію.

```
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

Рисунок 3.1 – Код підключення технології Flask до власного проекту

### 3.2 Вибір системи управління базою даних для власного проекту

Можливі варіанти використання бази даних:

1. MySQL.
2. SQLite.
3. PostgreSQL.

MySQL - це перевірена і завжди стабільна реляційна база даних, яка полегшує зберігання великих обсягів даних, взаємопов'язаних через певні типи з'єднань. PostgreSQL і SQLite одностипні, але перший з них має велику кількість додаткових типів даних, функцій і доповнень, а другий - полегшена і мобільна версія MySQL. Останній варіант є прикладом так званої бази даних NoSQL, яка заснована на методі зберігання даних {key: value}. Вони зручні для зберігання великих обсягів даних, які можуть бути слабо пов'язані між собою, але підвищують швидкість кожного.

PostgreSQL - це об'єктно-реляційна база даних, яка зосереджена на масштабованості та відповідності. Як сервер баз даних, його основна функція - безпечне збереження інформації та її відновлення на потребу програмних програм. Він може обробляти навантаження від невеликих програм на одній машині до великих програм з доступом до Інтернету та великою кількістю

одночасних користувачів. Це база даних за замовчуванням в операційних системах Apple, але її також можна використовувати в операційних системах Microsoft Windows і Linux. PostgreSQL є транзакційним і підтримує основні принципи транзакцій бази даних, такі як атомарність, константність, ізоляція та гнучкість. Він містить функції для оновлення та матеріалізації переглядів даних, тригерів, зовнішніх ключів, а також підтримує функції та збережені процедури. Це безкоштовна система з відкритим вихідним кодом, розроблена багатьма компаніями та індивідуальними інвесторами та поширюється під ліцензією PostgreSQL. Він підтримує функції зберігання та маніпулювання даними, такі як копіювання, індексування та створення шаблонів, із розширеною кількістю типів даних, визначених користувачем об'єктів, імітацій, інтерфейсів, тригерів тощо.

### 3.3 Вибір інструментарію голосового управління для власного проекту

SpeechRecognition - це бібліотека, написана на Python, яка надає інструменти розпізнавання мовлення та підтримує декілька механізмів та API. Він підтримує версії Python 2 і 3 і вимагає для запуску бібліотеки PyAudio. Це бібліотека, яка надає інструменти для використання мікрофонних пристроїв, а саме запису та зберігання аудіоінформації та представлення її у вигляді потоку. Pytsx - це кросплатформна оболонка для перетворення тексту на мову, написану на Python. Він має дуже простий інтероперабельний інтерфейс, а також підтримує кілька механізмів відтворення голосу на різних платформах. Також варто зазначити, що для розробки ми використовуємо середовище PyCharm. Середовище розроблено для використання цієї мови програмування. Воно інтегрує підтримку більшості фреймворків, вбудований термінал, консоль мови Python і дуже зручний налагоджувач. Процес налагодження та підтримка системи контролю версій (git тощо).



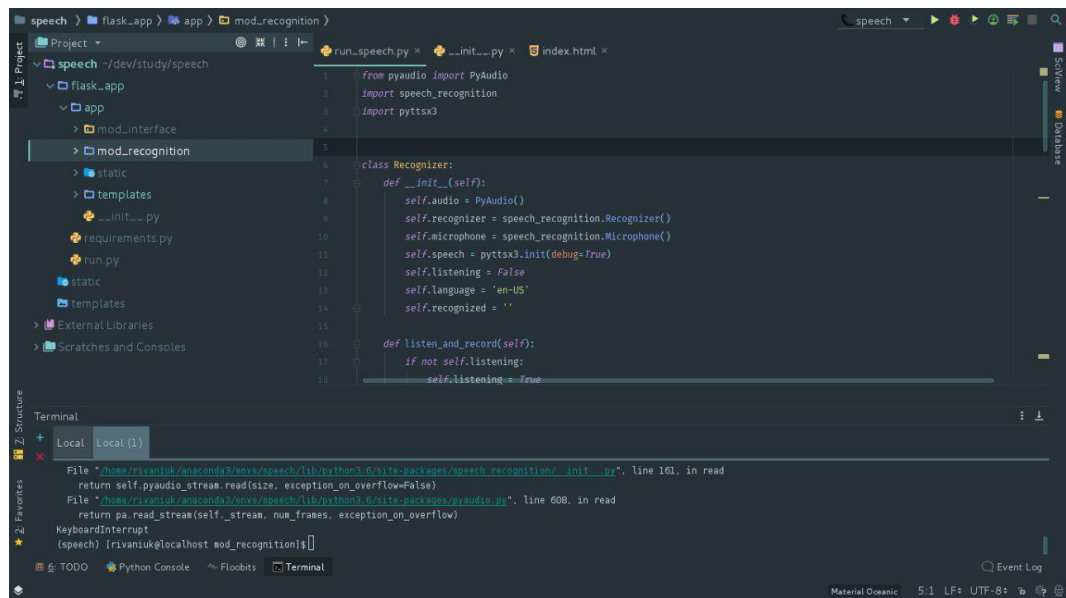
### 3.4 Компіляція і тестування власного застосунку для голосового управління

Загалом всю систему можна розділити на такі модулі:

1. Модуль веб-сервера інтерактивного інтерфейсу.
2. Модулі для запису, обробки, інтерпретації та синтезу мови.
3. Прикріпіть сценарій.

Модуль обробки мовного запису має такі характеристики:

1. Використовуйте мікрофон для запису інформації в циклі.
2. Записати обробку інформації та перетворити її в текст.
3. Зберігайте результати обробки в системних файлах.
4. Відтворіть голос у відповідь на команду.



```

speech > flask_app > app > mod_recognition >
Project
├── speech - /dev/study/speech
│   ├── flask_app
│   │   ├── mod_interface
│   │   ├── mod_recognition
│   │   ├── static
│   │   ├── templates
│   │   ├── ...init...py
│   │   ├── requirements.py
│   │   ├── run.py
│   │   ├── static
│   │   └── templates
│   ├── External Libraries
│   └── Scratches and Consoles
├── Structure
├── Favorites
├── TODO
├── Python Console
├── Floobits
└── Terminal

run_speech.py
1 from pyaudio import PyAudio
2 import speech_recognition
3 import pyttsx3
4
5 class Recognizer:
6     def __init__(self):
7         self.audio = PyAudio()
8         self.recognizer = speech_recognition.Recognizer()
9         self.microphone = speech_recognition.Microphone()
10        self.speech = pyttsx3.init(debug=True)
11        self.listening = False
12        self.language = 'en-US'
13        self.recognized = ''
14
15    def listen_and_record(self):
16        if not self.listening:
17            self.listening = True
18
Terminal
Local Local (1)
File ~/home/irvanisuk/anaconda2/envs/speech/lib/python3.6/site-packages/speech_recognition/_init_.py, line 161, in read
return self.pyaudio_stream.read(size, exception_on_overflow=False)
File ~/home/irvanisuk/anaconda2/envs/speech/lib/python3.6/site-packages/pyaudio.py, line 608, in read
return pa_read_stream(self_stream, num_frames, exception_on_overflow)
KeyboardInterrupt
(speech) [irvanisuk@localhost mod_recognition]$

```

Рисунок 3.2 – Інтерфейс обраного середовища для розробки

Основною перевагою системи є її модульність і цілісність: веб-інтерфейс реалізований окремим модулем, який отримує інформацію від окремих незалежних циклових процесів, що працюють паралельно. Все це буде розгорнуто на локальному хості, де розташована сама система. Адреса, що відповідає службі, - 0.0.0.0:5000. Якщо технічна частина розташована безпосередньо в розумному домі, і має достатню потужність, і оснащена

досить потужною системою прийому і передачі Wi-Fi, немає необхідності розгортати додаткові сервіси. В іншому випадку ви можете використовувати хмарне розгортання, наприклад систему Microsoft Azure, AWS Amazon тощо. Це видно з малюнка. 3.3 наша структура виглядає наступним чином. У головному каталозі проекту знаходяться модулі програми.

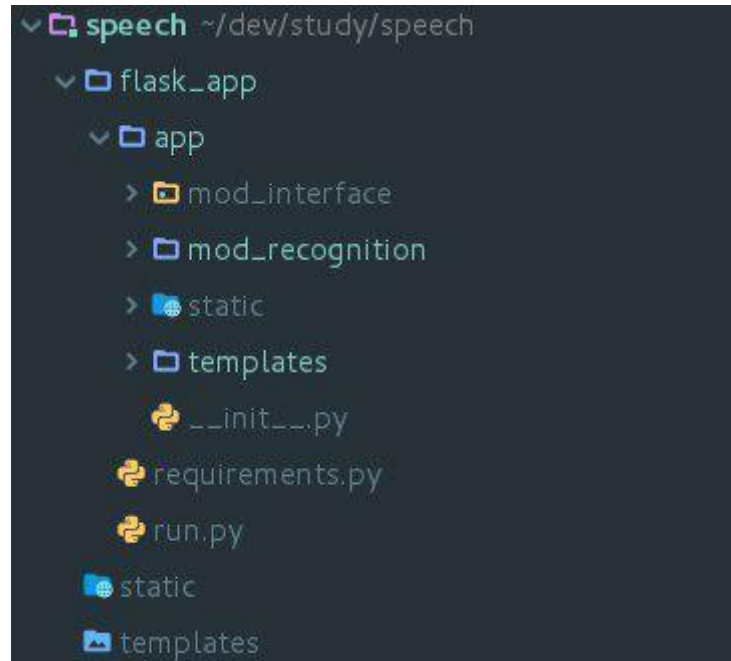


Рисунок 3.3 – Структура власного ERP- подібного застосунку для голосового управління

Усередині прикладного модуля є підмодуль для веб-інтерфейсу та розпізнавання голосу. Перший реалізував логіку контролера відображення веб-сторінки, а також взаємодію з клієнтом і файл результату розпізнавання, а також каталог зі статичними файлами CSS і JS і каталог з шаблонами HTML. Другий - це реалізація підмодуля розпізнавання та відтворення голосу, а також каталог файлу результату. Ви також можете використовувати інші скрипти для встановлення необхідних залежностей і компонентів. PyAudio є основним об'єктом модуля, що представляє здатність обробляти звук. Створено екземпляр класу розпізнавання мовлення, який міститиме екземпляри класів аудіодрайвера, розпізнавання мовлення, мікрофона та

програвача мовлення, а також статус прослуховування, поточну мову та остаточний результат розпізнавання.

```
class Recognizer:
    def __init__(self):
        self.audio = PyAudio()
        self.recognizer = speech_recognition.Recognizer()
        self.microphone = speech_recognition.Microphone()
        self.speech = pyttsx3.init(debug=True)
        self.listening = False
        self.language = 'en-US'
        self.recognized = ''

    def listen_and_record(self):
        if not self.listening:
            self.listening = True
            with self.microphone as mic:
                while True:
                    try:
                        print('Adjusting')
                        self.recognizer.adjust_for_ambient_noise(mic)
                        print('Listening')
                        res = self.recognizer.listen(mic, timeout=3, phrase_time_limit=3)
                        if res:
                            print('Recognizing')
                            recognized = self.recognizer.recognize_google(res,
                                                                           language=self.language)
                            if recognized:
                                self.recognized = recognized
```

Рисунок 3.4 Фрагмент коду з реалізацією голосового управління

Тепер реалізуємо клієнтську частину ERP- подібного застосунку для голосового управління . Функції модуля графічного інтерфейсу користувача включають:

1. Спостерігайте за поточним станом роботи.
  2. Дайте йому команди за допомогою інших засобів, крім голосового введення.
  3. Перезавантажте систему і працюйте.
  4. Спостерігайте за процесом розпізнавання в режимі реального часу.
- Виберіть бібліотеку React JavaScript, щоб розробити модулі інтерфейсу користувача.

React - відкрита бібліотека JavaScript, яка використовується для створення користувацьких інтерфейсів і спрямована на вирішення проблеми часткового оновлення веб-контенту, що виникає при розробці односторінкових додатків. React дозволяє розробникам створювати великі веб-додатки, які використовують дані, які змінюються з часом, не перезавантажуючи сторінку. Його мета - бути швидким, простим і масштабованим. React має справу лише з інтерфейсом користувача в додатку. Це відповідає подання в шаблоні модель-представлення-контролер (MVC), який можна об'єднати з іншими бібліотеками JavaScript або великими фреймворками MVC, такими як AngularJS. Його також можна використовувати з плагінами на основі React для обробки частин без інтерфейсу користувача для створення веб-додатків.

Його (React) переваги:

- Декларативний. React спрощує створення інтерактивних інтерфейсів. Все, що вам потрібно зробити, це описати, як виглядають різні частини інтерфейсу в кожному стані вашої програми. Коли ваші дані змінюються, React ефективно оновить і відобразить потрібні вам компоненти. Декларативний інтерфейс робить ваш код більш передбачуваним і легшим для налагодження;

- На основі компонентів. Дозволяє створювати інкапсульовані компоненти, які керують вашим власним станом, і створювати з них складні інтерфейси. Оскільки логіка компонента написана на JavaScript, а не на шаблоні, ми можемо легко передавати складні дані в програму та зберігати стан окремо від DOM;

- Це бібліотека. Дозволяють гнучко розвиватися.

Visual Studio Code було обрано як середовище розробки користувацького модуля. Це інструмент для створення та редагування сучасних веб-додатків і програм для хмарних систем. Visual Studio Code поширюється безкоштовно, і є версії для Windows, Linux і OS X.

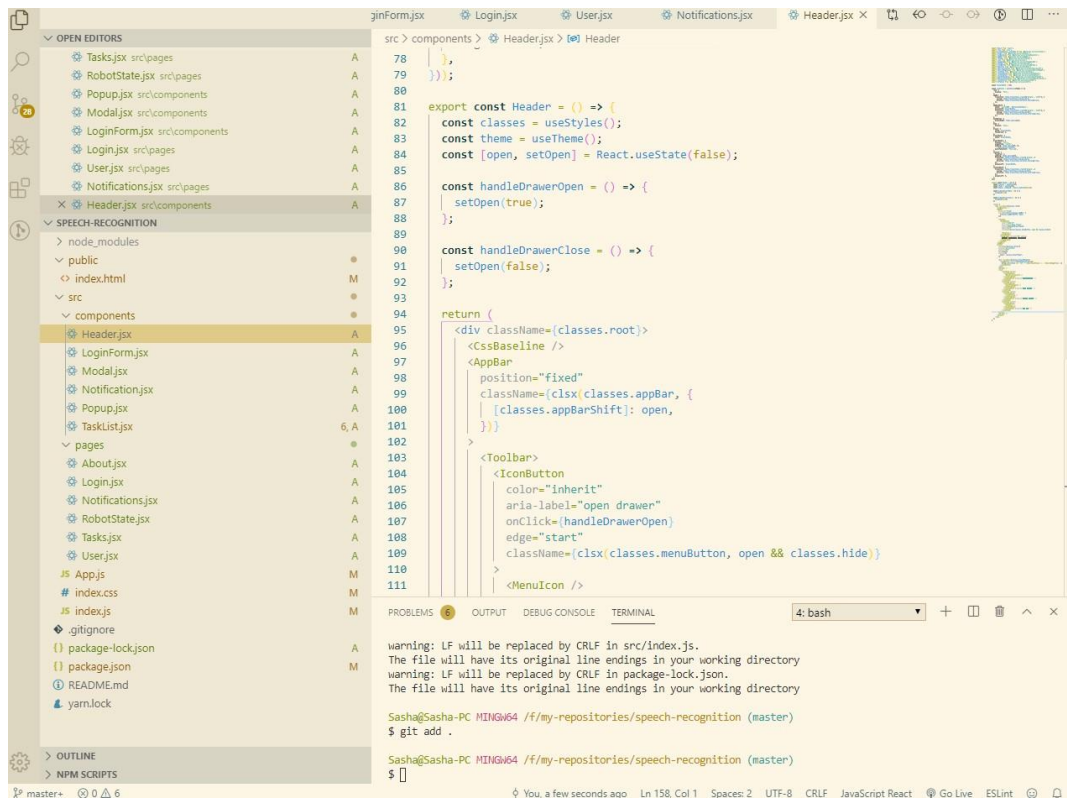


Рисунок 3.5 - Середовище для розробки React-інтерфейсу ERP-подібного застосунку для голосового управління

Завдання, засновані на функціональності програми, тобто можливі Функція створює навігаційне меню. Він включатиме такі маршрути:

- Повідомлення користувача про зміни статусу роботи та виконання завдань;
- Поточний стан роботи;
- Список доступних команд
- Інформаційний розділ про програму.

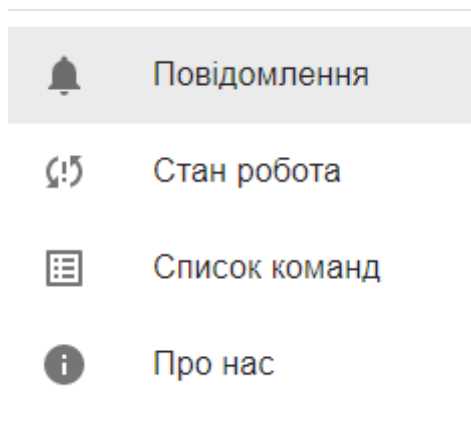


Рисунок 3.6 – Меню-навігатор

Структура Material UI використовується для стилів компонентів. Структура проекту складатиметься з компонентів та сторінок.

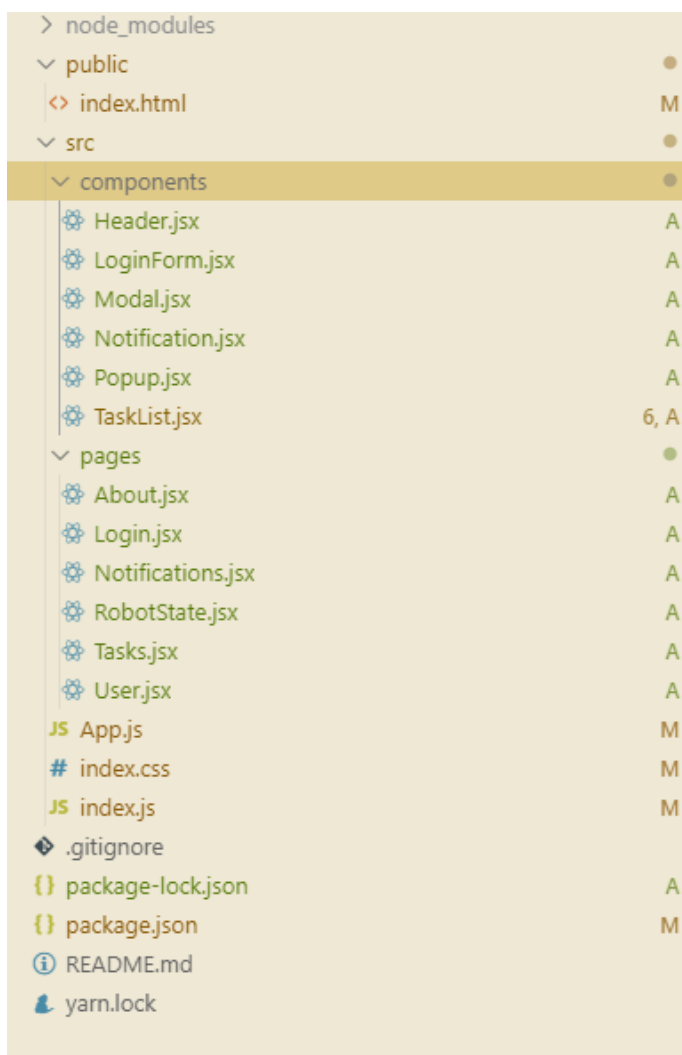


Рисунок 3.7 – Реакт-інтерфейс в вихідному представленні

Material UI - це бібліотека стилізованих компонентів React від Google, що дозволяє значно збільшити швидкість розробки. У той же час бібліотека надає можливість налаштувати всі компоненти відповідно до завдань розробки інтерфейсу користувача. Виберіть бібліотеку Redux, щоб відстежувати стан програми. Це значно покращить розвиток компонентів і моніторинг.

```
const reducer = (state = initialState, action) => {
  switch (action.type) {
    case 'USER_LOGGED_IN':
      const user = state.users.find(user => user.email === action.email);

      if (user && user.password !== action.password) {
        return {
          ...state,
          invalidEmail: false,
          invalidPassword: true
        };
      };

      if (user && user.password === action.password) {
        return {
          ...state,
          loggedIn: true,
          currentUser: user,
          invalidEmail: false,
          invalidPassword: false
        };
      };

      return {
        ...state,
        invalidEmail: true
      };
  }
};
```

Рисунок 3.8 – Redux-технологія авторизації користувача ERP-подібного застосунку для голосового управління

Redux - відкрита бібліотека JavaScript, призначена для керування станом програми. Найчастіше він використовується в поєднанні з React або Angular для створення інтерфейсів користувача. Redux був створений у 2015

році Деном Абрамовим та Ендрю Кларком. Redux був створений під впливом Flux, розробленого під впливом функціональної мови програмування Elm. Redux - це контейнер стану для програм JavaScript. Це допомагає розробникам оптимізувати програмний код. Крім того, це також покращує роботу розробника, наприклад редагування коду в реальному часі разом із налагоджувачем на роботі. Redux можна використовувати з React або іншими бібліотеками. Файл Redux дуже маленький, 2 КБ, включаючи залежності.

Redux зберігає стан всієї програми в дереві об'єктів у сховищі. Дерево станів полегшує налагодження або тестування програми; воно також дозволяє підтримувати стан програми під час розробки, щоб прискорити цикл розробки. Єдиний спосіб змінити стан - вибрати дію, об'єкт, який описує те, що сталося. Це гарантує, що ні вигляд, ні зворотний виклик мережі не змінять стан. Натомість вони лише висловили намір це зробити. Усі зміни централізовані і відбуваються одна за одною в чіткому порядку. Оскільки дії є простими об'єктами, їх можна зареєструвати, серіалізувати, зберегти, а потім відтворити для налагодження чи тестування.

Редуктори - це просто чисті функції, вони приймають попередній стан і дію і повертаються до наступного стану. У процесі розробки редуктор можна розділити на менші редуктори для керування певними частинами дерева станів. Оскільки редуктори - це лише функції, ви можете контролювати їх порядок надсилання, передавати додаткові дані і навіть створювати повторювані редуктори для звичайних завдань, таких як пейджинг. Програма повинна мати можливість отримувати дані з бази даних, щоб отримати список можливих команд. Команди можна викликати з пристрою, а не лише голосом. Цей модуль взаємодіє з іншими модулями для отримання Інформація про стан робота.



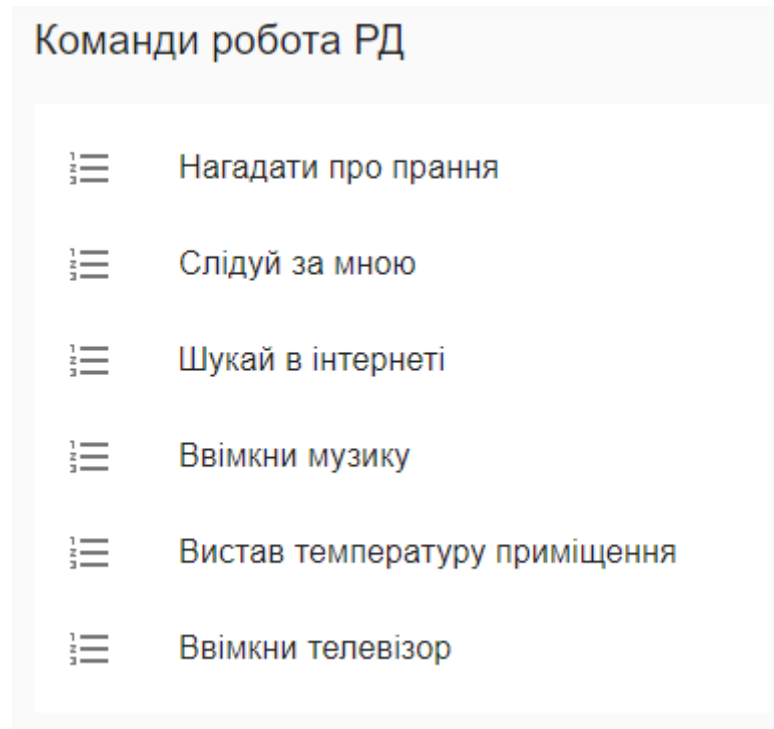


Рисунок 3.9 – Перелік сервісів ERP- подібного застосунку для голосового управління

Це лише кілька команд, деякі з яких мають підзавдання. Наприклад, окрім «пошуку в Інтернеті», система також хоче почути пошукові запити. У цьому розділі ми представили та пояснили технологію, використану в проекті, та безпосередню реалізацію нашої системи. Нарешті, ми отримали програмну реалізацію системи, яка може прослуховувати та розпізнавати голосову інформацію, надану користувачем хоста в режимі реального часу, сповіщати хост про відповідь і передавати відповідні сигнали для виконання інших можливих модулів або пристроїв. Інструменти та бібліотеки, які ми використовуємо для розробки, дозволяють нам значно спростити його, полегшити майбутню підтримку, додати нові функції та інтегрувати в готові технологічні комплекси. Використовуючи веб-інтерфейс, ми можемо переглядати інформацію про стан роботи та системи в режимі реального часу, Це, в свою чергу, забезпечується архітектурою, яку ми використовуємо під час розробки. Усі використовувані технології сучасні та підтримуються великими компаніями. Це змушує людей вірити, що технології

продовжуватимуть існувати й бути актуальними. Це забезпечить роботу наших програмних модулів тривалий час.

## 4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### 4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка застосунку для голосового управління типовими

операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [51].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

	0	1	2	3	4
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Продовження таблиці 4.1

	0	1	2	3	4
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовують ся у військово промисловом у комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовують ся у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	4	3
2. Ринкові переваги (наявність аналогів)	3	4	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	2	3	2
5. Ринкові переваги (експлуатаційні витрати)	3	4	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	3	3	4
8. Практична здійсненність (наявність фахівців)	4	3	3
9. Практична здійсненність (наявність фінансів)	3	3	4
10. Практична здійсненність (необхідність нових матеріалів)	3	3	3
11. Практична здійсненність (термін реалізації)	3	3	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	36	39	38
Середньоарифметична сума балів $СБ_c$	37,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [51].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ_c$ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема

класифікації» становить 37,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

#### 4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення розрахуємо за формулою [52]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де  $k$  – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

$\alpha_i$  – коефіцієнт, який враховує питому вагу  $i$ -го технічного показника в загальній якості розробки. Коефіцієнт  $\alpha_i$  визначається експертним шляхом і

при цьому має виконуватись умова  $\sum_{i=1}^k \alpha_i = 1$ ;

$\beta_i$  – відносне значення  $i$ -го технічного показника якості нової розробки.

Відносні значення  $\beta_i$  для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де  $I_{ni}$  та  $I_{ai}$  – чисельні значення конкретного  $i$ -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:



$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуюючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Швидкість пошуку слів	мс	10	6	1,67	0,1
Кількість рівнів класифікації	шт.	3	6	2	0,3
Правильність розпізнавання слів	%	85	85	1	0,15
Знаходження потрібної інформації	%	80	90	1,13	0,25
Доступність для користувача	бал	6	8	1,33	0,2

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,67 \cdot 0,15 + 2 \cdot 0,1 + 1 \cdot 0,3 + 1,13 \cdot 0,25 + 1,33 \cdot 0,2 = 1,47.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,47 рази.

### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

#### 4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

#### Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [51]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=22$  дні.

$$Z_o = 12500,00 \cdot 22 / 22 = 12500,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12500,00	568,18	22	12500,00
Інженер-програміст	11650,00	529,55	22	11650,00
Аналітик забезпечення голосового управління	11900,00	540,91	8	4327,27
Інженер-проектувальник автоматизованих систем управління	11450,00	520,45	22	11450,00
Всього				39927,27

#### Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства.

Ч. 2. Підсистема класифікації» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=2379,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [51];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 22$  дн;

$t_{зм}$  – тривалість зміни, год.

$$C_l = 2379,00 \cdot 1,10 \cdot 1,65 / (22 \cdot 8) = 24,53 \text{ грн.}$$

$$З_{pl} = 24,53 \cdot 6,00 = 147,20 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка офісного обладнання	6,00	2	1,10	24,53	147,20
Підготовка робочого місця розробника ERP-систем	8,00	3	1,35	30,11	240,87
Інсталяція програмного забезпечення розробки програмного модуля підсистеми класифікації	4,50	5	1,70	37,92	170,62
Формування інформаційної бази досліджень голосових команд	10,00	4	1,50	33,45	334,55
Тренування системи розпізнавання мовлення	8,00	4	1,50	33,45	267,64
Тестування підсистеми класифікації	8,00	3	1,35	30,11	240,87
Всього					1401,75

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{доп}} = (Z_o + Z_p) \cdot \frac{H_{\text{доп}}}{100\%}, \quad (4.7)$$

де  $H_{\text{доп}}$  – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{доп}} = (39927,27 + 1401,75) \cdot 11 / 100\% = 4546,19 \text{ грн.}$$

#### 4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{доп}}) \cdot \frac{H_{zn}}{100\%} \quad (4.8)$$

де  $H_{zn}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (39927,27 + 1401,75 + 4546,19) \cdot 22 / 100\% = 10092,55 \text{ грн.}$$

#### 4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації».

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (4.9)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$C_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 2,00 \cdot 128,00 \cdot 1,11 - 0,000 \cdot 0,00 = 284,16 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний SOLOMON-500 (A4)	128,00	2,00	-	-	284,16
Папір для заміток SOLOMON-100 (A5)/70	88,00	2,00	-	-	195,36
Начиння канцелярське SOLOMON	220,00	4,00	-	-	976,80
Органайзер офісний SOLOMON-DCZ	180,00	4,00	-	-	799,20
Картридж для принтера HP-A2142	780,00	2,00	-	-	1731,60
Диск оптичний SOLO-CD	12,20	3,00	-	-	40,63
FLASH-пам'ять Kingdom-GT 32GB	299,00	1,00	-	-	331,89
Всього					4359,64

#### 4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_e$ ), які використовують при проведенні НДР на тему «Розробка застосунку для голосового управління типовими

операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» відсутні.

#### 4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (4.10)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.і}}$  – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  – кількість найменувань устаткування.

$$B_{\text{спец}} = 12450,00 \cdot 1 \cdot 1,1 = 13695,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Синтезатор мовлення цифровий програмований	1	12450,00	13695,00
Блок інтерфейсний Rapid-ZX2020	1	1670,00	1837,00
Серверний блок бази даних	1	18650,00	20515,00
Всього			36047,00

#### 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних

засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{нрз}} = \sum_{i=1}^k C_{\text{нрз}} \cdot C_{\text{нрз.}i} \cdot K_i, \quad (4.11)$$

де  $C_{\text{нрз}}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{нрз.}i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{\text{нрз}} = 5530,00 \cdot 1 \cdot 1,11 = 6138,30 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows 11	1	5530,00	6138,30
Прикладний пакет Microsoft Office 2019	1	4370,00	4850,70
Прикладний пакет MATLAB Project	1	6800,00	7548,00
Всього			18537,00

#### 4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{C_{\text{б}}}{T_{\text{е}}} \cdot \frac{t_{\text{вук}}}{12}, \quad (4.12)$$



де  $C_6$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_6$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (25890,00 \cdot 1) / (2 \cdot 12) = 1078,75 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс	25890,00	2	1	1078,75
Графічно-обчислювальний комплекс обробки даних	21460,00	2	1	894,17
Програмне забезпечення розробки підсистеми параметризації	11300,00	2	1	470,83
Обладнання виводу інформації	7800,00	4	1	162,50
Обладнання вводу голосової інформації	6700,00	4	1	139,58
Офісна оргтехніка	9460,00	5	1	157,67
Приміщення дослідницької лабораторії	322000,00	20	1	1341,67
Всього				4245,17

## 4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.13)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 4,25$  грн;

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,20 \cdot 240,0 \cdot 4,25 \cdot 0,95 / 0,97 = 204,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс	0,20	160,0	204,00
Графічно-обчислювальний комплекс обробки даних	0,65	140,0	552,50
Синтезатор мовлення цифровий програмований	0,01	10,0	0,43
Обладнання виводу інформації	0,40	25,0	42,50
Обладнання вводу голосової інформації	0,02	40,0	3,40
Офісна оргтехніка	0,60	40,0	102,00
Серверний блок бази даних	0,20	140,0	170,00
Всього			1074,83

#### 4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.14)$$

де  $H_{cv}$  – норма нарахування за статтею «Службові відрядження», приймемо  $H_{cv} = 20\%$ .

$$B_{cv} = (39927,27 + 1401,75) \cdot 20 / 100\% = 8265,80 \text{ грн.}$$

#### 4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.15)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо  $H_{cn} = 40\%$ .

$$B_{cn} = (39927,27 + 1401,75) \cdot 40 / 100\% = 16531,61 \text{ грн.}$$

#### 4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.16)$$

де  $H_{ie}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{ie} = 50\%$ .

$$I_e = (39927,27 + 1401,75) \cdot 50 / 100\% = 20664,51 \text{ грн.}$$

#### 4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.17)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{нзв} = 100\%$ .

$$B_{нзв} = (39927,27 + 1401,75) \cdot 100 / 100\% = 41329,02 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{доо} + Z_n + M + K_e + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сн} + I_e + B_{нзв}. \quad (4.18)$$

$$B_{заг} = 39927,27 + 1401,75 + 4546,19 + 10092,54769 + 4359,64 + 0,00 + 36047,00 + 18537,00 + 4245,17 + 1074,83 + 8265,80 + 16531,61 + 20664,51 + 41329,02 = 207022,34 \text{ грн.}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.19)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta=0,9$ .

$$ZB = 207022,34 / 0,9 = 230024,83 \text{ грн.}$$

#### 4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$\Delta N$  – збільшення кількості споживачів програмного продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	500	750	1500	2000

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 6500 осіб;

$C_o$  – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 2700,00 грн;

$\pm\Delta C_o$  – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 300,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta\Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [51]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\mathcal{G}}{100}\right), \quad (4.20)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).  
Прийmemo  $\rho = 25\%$ ;

$\mathcal{G}$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році  $\mathcal{G} = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (300,00 \cdot 6500,00 + 3000,00 \cdot 500) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 587017,50 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (300,00 \cdot 6500,00 + 3000,00 \cdot 1250) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 969855,00 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (300,00 \cdot 6500,00 + 3000,00 \cdot 2750) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 1735530,00 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (300,00 \cdot 6500,00 + 3000,00 \cdot 4750) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 2756430,00 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків  $III$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$III = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.21)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,12$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 587017,50/(1+0,12)^1 + 969855,00/(1+0,12)^2 + 1735530,00/(1+0,12)^3 + \\ &+ 2756430,00/(1+0,12)^4 = 524122,77 + 773162,47 + 1235315,97 + 1751761,10 = \\ &= 4284362,31 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.22)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв} = 2$ ;

$ZB$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 230024,83 грн.

$$PV = k_{инв} \cdot ZB = 2 \cdot 230024,83 = 460049,65 \text{ грн.}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (4.23)$$

де  $III$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 4284362,31 грн;

$PV$  – теперішня вартість початкових інвестицій, 460049,65 грн.

$$E_{абс} = III - PV = 4284362,31 - 460049,65 = 3824312,66 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_6$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_6 = T_{жс} \sqrt[1 + \frac{E_{абс}}{PV}] - 1, \quad (4.24)$$

де  $E_{абс}$  – абсолютний економічний ефект вкладених інвестицій, 3824312,66 грн;

$PV$  – теперішня вартість початкових інвестицій, 460049,65 грн;

$T_{жс}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.



$$E_g = \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 3824312,66/460049,65)^{1/4} = 0,75.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій

$\tau_{\min}$ :

$$\tau_{\min} = d + f, \quad (4.25)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні  $d = 0,1$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,15.

$\tau_{\min} = 0,1 + 0,15 = 0,25 < 0,75$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.26)$$

де  $E_g$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,75 = 1,34 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації» становить 37,7 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,47 рази.

Також термін окупності становить 1,34 р., що менше 3-х років, і свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації».

## ВИСНОВКИ

У магістерській роботі ми проаналізували велику кількість матеріалів про історію розвитку технологій розумного дому, робототехніки та Інтернету речей, розпізнавання голосу та відтворення існуючих технологій та реалізацій, що дозволило створити стабільну, просту та легку у використанні -удосконалити підпроцес однокристалъну структуру програмної системи, інтерфейс для взаємодії людини та комп'ютера через голос, роботу через прослуховування, обробку, інтерпретацію та відповіді, роботу в циклах, що дозволяє безперервно виконувати розпізнавання та інтерпретацію в наступних командах, а також моніторинг роботи статус і розпізнавання за допомогою процесу веб-інтерфейсу.

Описує вимоги та пояснення щодо функцій системи розпізнавання голосових команд. Алгоритм системи допомоги робота описує повний цикл взаємодії з користувачем. Починаючи зі стаціонарного режиму робота, у нашому прикладі це режим очікування кодового слова як сигнал для початку обробки мови та порівняння отриманого з базою типових команд. У наступних випадках система не може закінчитися заданим алгоритмом поведінки розшифруйте слово, або не знайшли відповідну команду.

Стійкість до помилок є важливим фактором у розробці таких систем, оскільки в складній системі, наприклад «розумний будинок», існує багато сценаріїв, а нерозуміння голосових команд може призвести до небажаних наслідків або тупиків і невизначеності. Спілкування також має бути інтуїтивно зрозумілим для користувача. Система повинна врахувати всі можливі помилки і правильно реагувати. Повідомляйте користувачів при переході з різних станів, щоб уникнути помилок. Крім того, команди, введені в базу даних, повинні значно відрізнятися одна від одної, щоб уникнути непотрібних тригерів.

Інструменти та бібліотеки, які ми використовуємо для розробки, дозволяють нам значно спростити її, полегшити підтримку в майбутньому,

додати нові функції та інтегрувати в готові технологічні комплекси. Також ми провели комплексну оцінку всіх параметрів безпеки та охорони праці та отримали всі необхідні параметри для комфортного та безпечного робочого місця для розробників систем. Надалі планується подальше розширення системи та удосконалення наявних недоліків для досягнення максимальної ефективності та мінімальних помилок.

Усі технології, які ми використовуємо для впровадження модулів, сучасні та підтримуються великими компаніями. Це змушує людей вірити, що технології продовжуватимуть існувати й бути актуальними. Це забезпечить роботу наших програмних модулів тривалий час.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аграновский, А.В. Теоретические аспекты алгоритмов обработки и классификации речевых сигналов / А.В. Аграновский, Д.А. Леднов. М.: Радио и связь, 2004. 164 с.
2. Винцюк, Т.К., Анализ, распознавание и интерпретация речевых сигналов / Т.К. Винцюк. Киев: Наук.думка, 1987. 264с.
3. Голд, Б. Цифровая обработка сигналов / Б. Голд, Ч. Рейдер. М.: Сов. Радио, 1973. 368 с.
4. Сорокин В.Н., Цыплихин А.И. Сегментация и распознавание гласных// Информационные процессы, т. 4 , № 2, с. 202-220, 2004 г.
5. Жиялков, Е.Г. Вариационные методы анализа и построения функций по эмпирическим данным /Е.Г. Жиялков. Белгород: Изд-во БелГУ, 2007. 160 с.
6. Жиялков, Е.Г. Методы обработки речевых данных в информационно- телекоммуникационных системах на основе частотных представлений / Е.Г. Жиялков, С.П. Белов, Е.И. Прохоренко. Белгород: Изд-во БелГУ, 2007. 136 с.
7. Жиялков, Е.Г. Сегментация речевых сигналов на основе анализа распределения энергии по частотным интервалам / Е.Г. Жиялков, Е.И. Прохоренко, А.В. Болдышев, А.А. Фирсова, М.В. Фатова // Научные ведомости Белгородского государственного университета. Серия: История. Политология. Экономика. Информатика, Том 18. 2011. №7-1 (102). С. 187-196
8. Кипяткова, И.С. Автоматическая обработка разговорной русской речи: монография / И.С. Кипяткова, А.Л. Ронжин, А.А. Карпов. СПИИРАН. СПб.: ГУАП, 2013. 314 с.
9. Куприянов, М.С. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования / М.С. Куприянов. СПб.: Политехника, 1999. 592 с.

10. Лайонс, Р. Цифровая обработка сигналов / Лайонс Р; 2-е изд. ; Пер. с англ. М.: ООО "Бином-Пресс", 2006. 656 с.: ил.
11. Марпл-мл, С.Л. Цифровой спектральный анализ и его приложения / Марпл-мл. С.Л.; Пер. с англ. М.: Мир, 1990.
12. Михайлов В.Г., Златоусов Л.В. Измерение параметров речи / В.Г. Михайлов, Л.В. Златоусова; Под.ред. М.А. Сапожникова. Москва: Радио и связь, 1987. 168с.: ил.
13. Рабинер Л. Р., Шафер Р.В. Цифровая обработка речевых сигналов = Digitalprocessingofspeechsignals / Л.Р. Рабинер, Р.В. Шафер.;Пер. с англ.М.В. Назарова, Ю.Н. Прохорова; Под ред. М.В.Назарова, Ю.Н. Прохорова. Москва: Радио и связь, 1981. 496с.:ил.
15. Сергиенко А.Б. Цифровая обработка сигналов: учебное пособие. 3-е изд. М.: БХВ-Петербург, 2011. 768 с.
16. Солонина, А.И. Основы цифровой обработки сигналов / А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б.Соловьева. СПб.: БХВ-Петербург, 2005. 768с.: ил.
17. Ю.Лабунец В. Г. Алгебраическая теория сигналов и систем. Красноярск: Изд-во Краснояр. ун-та, 1984.
18. Цемель Г. И. Оpoznание речевых сигналов. М., Наука, 1971.
19. Шелепов В.Ю. Новые алгоритмы сегментации речевого сигнала и распознавания некоторых классов фонем / В.Ю. Шелепов, А.В. Ниценко // Искусственный интеллект. 2007. № 1. С. 213-224.
20. Шелепов В.Ю. Новые алгоритмы распознавания фонем и их классов, поиск слова по его смешанной транскрипции при распознавании слов большого словаря / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук // Искусственный интеллект. 2007. № 2. С. 139-147.
21. Шелепов В.Ю. О распознавании фонем с помощью анализа речевого сигнала в частотной и временной областях. Приложение к распознаванию синтаксически связанных фраз / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук, Д.С. Азаренко // Речевые технологии. 2008. № 2. С. 43-52

22. Жилияков Е.Г., Фирсова А.А.. Сегментация речевых сигналов на основе субполосного анализа // Вестник НТУ "ХПИ", № 39 (1012). 2013г. С.73-81 .
23. Фирсова А.А. Разработка и исследование субполосных методов и алгоритмов сегментации речевых сигналов / Фирсова А.А.//Автореферат диссертации на соискание ученой степени кандидата технических наук. 17 мая 2013 г. Белгород. С.15-19.
24. Музычук Д.С. Сегментация, шумоподавление и фонетический анализ в задаче распознавания речи / Музычук Д.С., Медведев М.С. // Молодой ученый. 2013. №6. С. 86-96.
25. Дремин И.М., Иванов О.В., Нечитайло В.А. Вейвлеты и их использование. //Успехи физических наук, т. 171, №5 с. 465-500, 2001 г.
26. Ермоленко Т.Н. Алгоритмы сегментации с применением быстрого вейвлет- преобразования / Т.Н. Ермоленко, В.И. Шевчук // Статьи, принятые к публикации на сайте международной конференции Диалог. 2003.
27. Шелухин О.И. Цифровая обработка и передача речи . / Лукьянцев Н.Ф. М.: Радио и связь, 2000. 454 с.
28. Жилияков, Е.Г. Вариационные методы частотного анализа звуковых сигналов . / Белов С.П., Прохоренко Е.И. // Труды учебных заведений связи. СПб, 2006. № 174. С.163-170.
29. Жилияков Е.Г. Бабаринов С.Л. Чадюк П.В. Исследование сервиса компании Google Inc. по распознаванию русской речи / Жилияков Е.Г. Бабаринов С.Л. Чадюк П.В // Научные ведомости Белгородского государственного университета. Серия: История. Политология. Экономика. Информатика № 15-1 (158). том 27. 2013
30. Жилияков Е.Г. Вариационные методы анализа и построения функций по эмпирическим данным на основе частотных представлений. Белгород: Изд-во БелГУ, 2007. с. 160
31. Цыплихин А.И. Системный анализ, управление и обработка информации. 2006.

32. Фирсова А.А. О различиях распределения энергии звуков русской речи и шума / А.В. Болдышев, А.А. Фирсова // Материалы 12-ой Международной конференции и выставки "Цифровая обработка сигналов и её применение. –"DSPА'2010". Москва. 2010. С. 204–207.
33. Сорокин В.К. Синтез речи. М. : Наука, 1992. С. 392
34. Цыплихин А.И. Анализ и автоматическая сегментация речевого сигнала: дис. канд. тех. наук / ИППИ РАН. М., 2006. 149 с.
35. А. С. Колоколов. Обработка сигнала в частотной области при распознавании речи. 2006. с. 13–18
36. Конев А. А. Параметрическое описание сегментов речевого сигнала/ В. И. Голубев, А. А. Конев // Научная сессия ТУСУР – 2005: Материалы Всероссийской научно-технической конференции студентов, аспирантов и молодых специалистов. Томск: Издательство ТУСУРа, 2005. С. 113- 116.
37. Кочаров Д.А. Автоматическая интерпретация звуков речи // Диссертационная работа. СПбГУ 2008
38. Утробин В.А., Гай В.Е. Алгоритм выделения вокализованных участков речевого сигнала // Вестник Нижегородского университета им. Н.И. Лобачевского, 2012, № 6 (1). С. 175–179
39. Мясникова Е.Н. Объективное распознавание звуков речи Л.: «Энергия», 1967. 150 с.
40. Черник, Н. Н. Сегментация спонтанной речи в языках различных типов / Н. Н. Черник // Вестник Белорусского государственного экономического университета. 2009. N 4. С. 101-107.
41. Ли У. А. Методы автоматического распознавания речи. М., Мир, 1983.
42. Огнев И.В., Огнев А.И., Парамонов П.А., Классификация речевых образов на основе анализа распределений их локальных экстремумов// труды XXI международной научно-технической



конференции "Информационные средства и технологии". - М.: МЭИ, 2013 - с. 53-57.

43. Винцюк Т.К., Анализ, распознавание и интерпретация речевых сигналов. Киев: Наукова думка, 1987. 264 стр.

44. Бондаренко Л. В., Вербицкая Л. А., Гордина М. В., Основы общей фонетики. М.: Академия, 2004. 160 с.

45. Шарий Т.В., О проблеме параметризации речевого сигнала в современных системах распознавания речи // Вісник Донецького Національного Університету, 2008, вып. 2, стр. 536-541

46. Маркел Дж. Д., Грэй А. Х., Линейное предсказание речи. М.: Связь, 1980. – 308 с.

47. Загоруйко Н. Г., Методы распознавания и их применение. М.: Советское радио, 1972. 208 с.

48. Агашин О.С., Корелин О.Н., Методы цифровой обработки речевого сигнала в задаче распознавания изолированных слов с применением сигнальных процессоров // Труды Нижегородского государственного технического университета им. Р.Е. Алексеева № 4, 2012, с. 32-44

49. Ронжин А.Л., Ли И. В. Автоматическое распознавание русской речи // Вестник Российской академии наук, 2007, том 77, № 2, с. 133-138.

50. Огнев И. В., Парамонов П.А. Исследование способов представления числа для реализации арифметических операций в ассоциативной среде с командным управлением // Информационные средства и технологии: труды Международной научно-технической конференции (19 – 21 октября 2010 г.): в 3 т. М.: МЭИ, 2010. 1 т. с. 54-60.

51. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

52. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с.

## ДОДАТКИ

## Додаток А

**ЗАТВЕРДЖУЮ**

Завідувач кафедри КСУ

д.т.н., проф.

\_\_\_\_\_ Володимир Дубовой

**ТЕХНІЧНЕ ЗАВДАННЯ**

на магістерську кваліфікаційну роботу

«Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2.

Підсистема класифікації»

08-01.МКР.004.00.000 ТЗ

Виконала: студентка 2-го курсу,  
групи 2АКІТ-20м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології  
(шифр і назва спеціальності)

\_\_\_\_\_ Ірина Майданевич  
(ім'я та прізвище)

Керівник: д.т.н., професор каф. КСУ

\_\_\_\_\_ В'ячеслав Ковтун  
(ім'я та прізвище)

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

## 1. Назва та галузь застосування

Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2. Підсистема класифікації.

Розроблена архітектура, модель і система для підвищення ефективності класифікації мовного сигналу.

Розробка призначена для використання в галузях інформаційних технологій.

## 2. Підстави для розробки

Розробку системи здійснювати на підставі наказу по університету №\_\_\_\_\_ від \_\_\_\_\_ та завдання до магістерської кваліфікаційної роботи складеного та затвердженого кафедрою КСУ.

## 2. Мета та призначення розробки

Метою дипломної дисертації є підвищення ефективності класифікації мовного сигналу та спрощення взаємодії між користувачами та розумними пристроями. Для досягнення поставленої мети необхідно розробити архітектуру, модель і саму відповідну систему.

## 4. Джерела розробки

1. CO-ResNet: Optimized ResNet model for COVID-19 diagnosis from X-ray images / S. Bharati et al. International Journal of Hybrid Intelligent Systems. 2021. Vol. 17. P. 71–85.

2. Nibali A., He Z., Wollersheim D. Pulmonary Nodule Classification with Deep Residual Networks // International Journal of Computer Assisted Radiology and Surgery. 2017. Vol. 12. P. 1799–1808.

3. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent // Proceedings of COMPSTAT2010. 2010. P. 177–186.

4. He S., Wu Q.H., Saunders J.R. A Group Search Optimizer for Neural Network Training // Computational Science and Its Applications - ICCSA 2006 Lecture Notes in Computer Science. 2006. P. 934–943.

5. Rozložník M. Solution Approaches for Saddle-Point Problems // Nečas Center Series Saddle-Point Problems and Their Iterative Solution. 2018. P. 33–39.

### 5. Показники призначення

Вихідними даними для розробки є результати аналізу об'єкта дослідження.

Результатом роботи методу є підвищення ефективності класифікації мовного сигналу та спрощення взаємодії між користувачами та розумними пристроями.

### 6. Стадії розробки

1. Стадія розробки «Огляд предметної області» має бути виконана до 04.09.2021р.
2. Стадія розробки «Розробка математичного апарату» має бути виконана до 22.09.2021р.
3. Стадія розробки «Розробка програмного забезпечення та експериментальні дослідження» має бути виконана до 30.10.2021р.
4. Стадія розробки «Підготовка економічної частини» має бути виконана до 12.11.2021р.
5. Стадія розробки «Оформлення пояснювальної записки і графічного матеріалу» має бути виконана до 08.12.2021р.

### 7. Порядок контролю та приймання

1. Рубіжний контроль. Провести до «\_\_» \_\_\_\_\_ 202\_ р.
2. Попередній захист магістерської кваліфікаційної роботи. Провести до «\_\_» \_\_\_\_\_ 202\_ р.
3. Захист магістерської кваліфікаційної роботи. Провести «\_\_» \_\_\_\_\_ 202\_ р.

## Додаток Б

### Фрагменты программного коду

```

import speech_recognition as sr from gtts import
gTTS

import pygame
from pygame import mixer mixer.init()
import os import sys import time import datetime
import logging
import webbrowser import subprocess

class Speech_AI: def  init  (self):
self._recognizer = sr.Recognizer() self._microphone
= sr.Microphone()

now_time = datetime.datetime.now()
self._mp3_name =
now_time.strftime("%d%m%Y%H%M%S") +
".mp3" self._mp3_nameold = '111'

def work(self):
print("Минутку тишины, пожалуйста...") with
self._microphone as source:
self._recognizer.adjust_for_ambient_noise(source)

try:
while True:
print("Скажи что -нибудь!") with self._microphone
as source:
audio = self._recognizer.listen(source) print("Понял,
идет распознавание...") try:
statement =
self._recognizer.recognize_google(audio,
language="ru_RU")
statement = statement.lower()

# Команды для открытия различных внешних
приложений

if ((statement.find("калькулятор") != -1) or
(statement.find("calculator") != -1)):
self.osrun('calc')

!= -1)):

```

1)):

!= -1)):

!= -1) or (

-1)) and (

!= -1) or (

```
if ((statement.find("блокнот") != -1) or
(statement.find("notepad") self.osrun('notepad')
if ((statement.find("paint") != -1) or
(statement.find("паинт") != - self.osrun('mspaint')
if ((statement.find("browser") != -1) or
(statement.find("браузер")
self.openurl('http://google.ru', 'Открываю браузер')
# Команды для открытия URL в браузере
if (((statement.find("youtube") != -1) or
(statement.find("youtub") statement.find("ютуб") !=
-1) or (statement.find("you tube") !=
statement.find("смотреть") == -1)):
self.openurl('http://youtube.com', 'Открываю ютуб')
```

```
if (((statement.find("новости") != -1) or
(statement.find("новость")
```

```
statement.find("на усть") != -1)) and (
(statement.find("youtube") == -1) and
```

```
(statement.find("youtub") != -1) and (
statement.find("ютуб") == -1) and
(statement.find("you tube")
```

== -1))):

```
self.openurl('https://www.youtube.com/user/rtrussian
/videos',
```

```
'Открываю новости')
```



почту')

```
if ((statement.find("mail") != -1) or
(statement.find("майл") != -1)):
self.openurl('https://e.mail.ru/messages/inbox/',
'Открываю
```

```
if ((statement.find("вконтакте") != -1) or
(statement.find("в контакте") != -1)):
self.openurl('http://vk.com', 'Открываю Вконтакте')
```

# Команды для поиска в сети интернет

1) or (

1) or (

```
if ((statement.find("найти") != -1) or
(statement.find("поиск") != -
statement.find("найди") != -1) or
(statement.find("дайте") != -
statement.find("mighty") != -1)): statement =
statement.replace('найди', ") statement =
statement.replace('найти', ") statement =
statement.strip()
self.openurl('https://yandex.ru/yandsearch?text=' +
statement, "Я
```

нашла следующие результаты")

1))):

```
if ((statement.find("смотреть") != -1) and (
(statement.find("фильм") != -1) or
(statement.find("film") != -
```

```
statement = statement.replace('посмотреть', ")
statement = statement.replace('смотреть', ")
statement = statement.replace('хочу', ") statement =
statement.replace('фильм', ") statement =
statement.replace('film', ") statement =
statement.strip()
```

```
self.openurl('https://yandex.ru/yandsearch?text=Смо
треть+онлайн+фильм+' + statement,
"Выберите сайт где смотреть фильм")
```

```
-1) or (
```

```
if (((statement.find("youtube") != -1) or
(statement.find("ютуб") !=
```

```
statement.find("you tube") != -1)) and
```

```
(statement.find("смотреть") != -1)):
```

```
statement = statement.replace('хочу', ") statement =
statement.replace('на ютубе', ") statement =
statement.replace('на ютуб', ") statement =
statement.replace('на youtube', ") statement =
statement.replace('на you tube', ") statement =
statement.replace('на youtub', ") statement =
statement.replace('youtube', ") statement =
statement.replace('ютуб', ") statement =
statement.replace('ютубе', ") statement =
statement.replace('посмотреть', ") statement =
statement.replace('смотреть', ") statement =
statement.strip()
```

```
self.openurl('http://www.youtube.com/results?search
_query=' + statement, 'Ищу в ютуб')
```

```
-1)):
```

```
if ((statement.find("слушать") != -1) and
(statement.find("песн") !=
```

```
statement = statement.replace('песню', ") statement
= statement.replace('песни', ") statement =
statement.replace('песня', ") statement =
statement.replace('песней', ") statement =
statement.replace('послушать', ") statement =
statement.replace('слушать', ") statement =
statement.replace('хочу', ") statement =
statement.strip()
```

```
self.openurl('https://my.mail.ru/music/search/' +
statement,
```

```
"Нажмите плэй")
```

```
# Поддержание диалога
```

```

if ((statement.find("до свидания") != -1) or
(statement.find("досвидания") != -1)):
answer = "Пока!" self.say(str(answer))
while pygame.mixer.music.get_busy():
time.sleep(0.1)
sys.exit()
print("Вы сказали: {}".format(statement)) except
sr.UnkownValueError:
print("Упс! Кажется, я тебя не поняла, повтори
еще раз") except sr.RequestError as e:
print("Не могу получить данные от сервиса
Google Speech Recognition; {}".format(e))
except KeyboardInterrupt: self._clean_up()
print("Пока!")

def osrun(self, cmd):
PIPE = subprocess.PIPE
p = subprocess.Popen(cmd, shell=True, stdin=PIPE,
stdout=PIPE, stderr=subprocess.STDOUT)

def openurl(self, url, ans): webbrowser.open(url)

self.say(str(ans))
while pygame.mixer.music.get_busy():
time.sleep(0.1)

def say(self, phrase):
tts = gTTS(text=phrase, lang="ru")
tts.save(self._mp3_name)

# Play answer mixer.music.load(self._mp3_name)
mixer.music.play()
if (os.path.exists(self._mp3_nameold)):
os.remove(self._mp3_nameold)

now_time = datetime.datetime.now()
self._mp3_nameold = self._mp3_name
self._mp3_name =
now_time.strftime("%d%m%Y%H%M%S") +
".mp3"

def _clean_up(self): def clean_up():
os.remove(self._mp3_name)

def main():
ai = Speech_AI() ai.work()

main()

```

```

Тестувальна частина import
os
import unittest

import speech_recognition as sr

class TestRecognition(unittest.TestCase):
    def setUp(self):

        self.AUDIO_FILE_EN =
os.path.join(os.path.dirname(os.path.realpath(
    file    )), "english.wav")
        self.AUDIO_FILE_FR =
os.path.join(os.path.dirname(os.path.realpath(
    file    )), "french.aiff")
        self.AUDIO_FILE_ZH =
os.path.join(os.path.dirname(os.path.realpath(
    file    )), "chinese.flac")

    def test_sphinx_english(self):
        r = sr.Recognizer()
        with sr.AudioFile(self.AUDIO_FILE_EN) as source:
            audio = r.record(source)
        self.assertEqual(r.recognize_sphinx(audio), "wanted
to three")

    def test_google_english(self):
        r = sr.Recognizer()
        with sr.AudioFile(self.AUDIO_FILE_EN) as source:
            audio = r.record(source)
        self.assertIn(r.recognize_google(audio), ["1 2 3",
"one two three"])

    def test_google_french(self):
        r = sr.Recognizer()
        with sr.AudioFile(self.AUDIO_FILE_FR) as source:
            audio = r.record(source)
        self.assertEqual(r.recognize_google(audio,
language="fr-FR"), u"et c'est la dictée numéro 1")

    def test_google_chinese(self):
        r = sr.Recognizer()
        with sr.AudioFile(self.AUDIO_FILE_ZH) as source:
            audio = r.record(source)
        self.assertEqual(r.recognize_google(audio,
language="zh-CN"), u"砸自己的脚")

```

```

@unittest.skipUnless("WIT_AI_KEY" in
os.environ, "requires Wit.ai key to be specified in
WIT_AI_KEY environment variable")

def test_wit_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source:
audio = r.record(source)
self.assertEqual(r.recognize_wit(audio,
key=os.environ["WIT_AI_KEY"]), "one two three")

@unittest.skipUnless("BING_KEY" in os.environ,
"requires Microsoft Bing Voice Recognition key to
be specified in BING_KEY environment variable")
def test_bing_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source:
audio = r.record(source)
self.assertEqual(r.recognize_bing(audio,
key=os.environ["BING_KEY"]), "123.")

@unittest.skipUnless("BING_KEY" in os.environ,
"requires Microsoft Bing Voice Recognition key to
be specified in BING_KEY environment variable")
def test_bing_french(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_FR) as source:
audio = r.record(source)
self.assertEqual(r.recognize_bing(audio,
key=os.environ["BING_KEY"], language="fr-FR"),
u"Essaye la dictée numéro un.")

@unittest.skipUnless("BING_KEY" in os.environ,
"requires Microsoft Bing Voice Recognition key to
be specified in BING_KEY environment variable")
def test_bing_chinese(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_ZH) as source:
audio = r.record(source)
self.assertEqual(r.recognize_bing(audio,
key=os.environ["BING_KEY"], language="zh-
CN"), u"砸自己的脚
。 ")

@unittest.skipUnless("HOUNDIFY_CLIENT_ID"
in os.environ and "HOUNDIFY_CLIENT_KEY" in
os.environ, "requires Houndify client ID and client
key to be specified in HOUNDIFY_CLIENT_ID and
HOUNDIFY_CLIENT_KEY
environment variables")
def test_houndify_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source:
audio = r.record(source)

```

```
self.assertEqual(r.recognize_houndify(audio,
client_id=os.environ["HOUNDIFY_CLIENT_ID"],
client_key=os.environ["HOUNDIFY_CLIENT_KEY"]), "one two three")
```

```
@unittest.skipUnless("IBM_USERNAME" in
os.environ and "IBM_PASSWORD" in os.environ,
"requires IBM Speech to Text username and
password to be specified in IBM_USERNAME and
IBM_PASSWORD environment variables")
def test_ibm_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source:
audio = r.record(source)
self.assertEqual(r.recognize_ibm(audio,
username=os.environ["IBM_USERNAME"],
password=os.environ["IBM_PASSWORD"]), "one
two three ")
```

```
@unittest.skipUnless("IBM_USERNAME" in
os.environ and "IBM_PASSWORD" in os.environ,
"requires IBM Speech to Text username and
password to be specified in IBM_USERNAME and
IBM_PASSWORD environment variables")
def test_ibm_french(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_FR) as source:
audio = r.record(source)
self.assertEqual(r.recognize_ibm(audio,
username=os.environ["IBM_USERNAME"],
password=os.environ["IBM_PASSWORD"],
language="fr-FR"), u"si la dictée numéro un ")
```

```
@unittest.skipUnless("IBM_USERNAME" in
os.environ and "IBM_PASSWORD" in os.environ,
"requires IBM Speech to Text
```

```
username and password to be specified in
IBM_USERNAME and IBM_PASSWORD
environment variables")
def test_ibm_chinese(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_ZH) as source:
audio = r.record(source)
self.assertEqual(r.recognize_ibm(audio,
username=os.environ["IBM_USERNAME"],
password=os.environ["IBM_PASSWORD"],
if name__ == " main ": unittest.main()
```

## Додаток В

## ІЛЮСТРАТИВНА ЧАСТИНА

**Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 2.****Підсистема класифікації**

Перелік ілюстративних матеріалів:

1. Алгоритм роботи застосунку
2. Архітектура ERP- подібного застосунку для голосового управління
3. Схема технології для розпізнавання усного мовлення
4. Архітектура нейромережі
5. Структура власного ERP-подібного застосунку для голосового управління
6. Фрагмент коду з реалізацією голосового управління
7. Меню-навігатор
8. React-інтерфейс в вихідному представленні
9. Перелік сервісів ERP- подібного застосунку для голосового управління

Виконала: студентка 2-го курсу,  
групи 2АКІТ-20м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології  
(шифр і назва спеціальності)

\_\_\_\_\_ Ірина Майданевич

(ім'я та прізвище)

Керівник: д.т.н., професор каф. КСУ

\_\_\_\_\_ В'ячеслав Ковтун

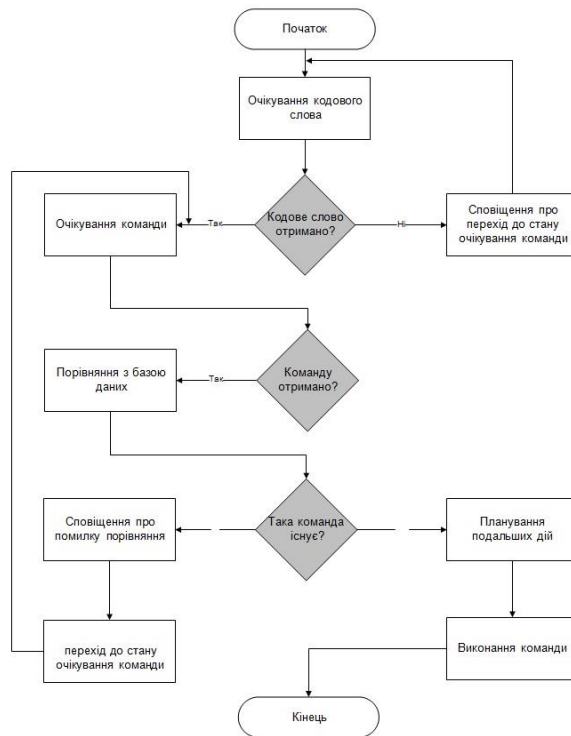
(ім'я та прізвище)

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

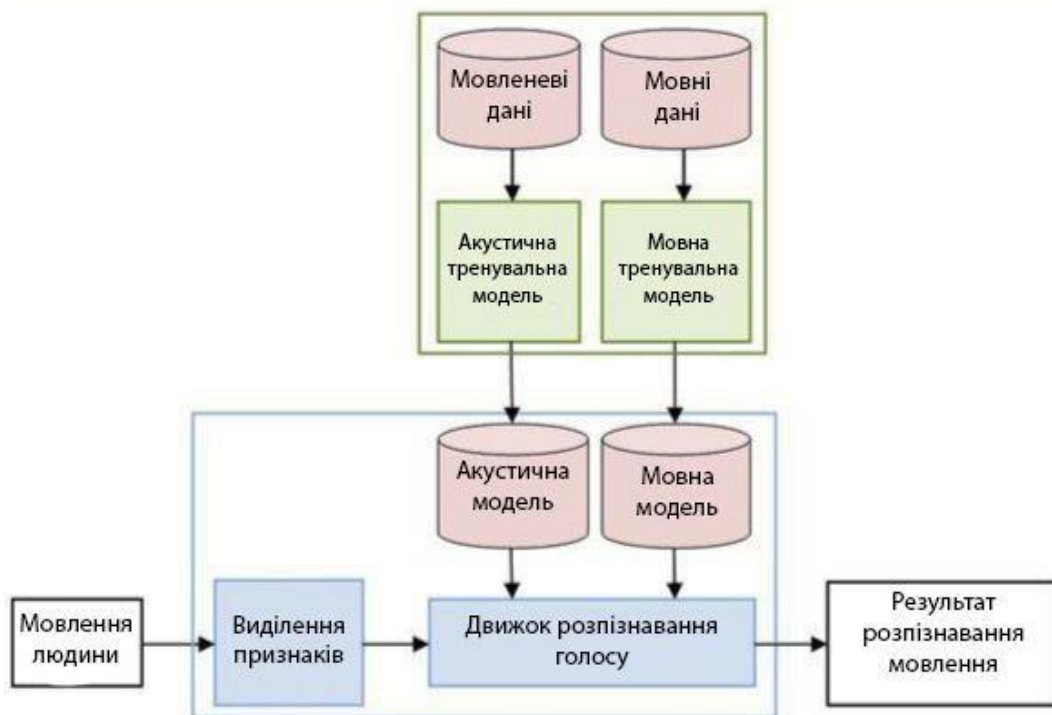
Опонент: \_\_\_\_\_

\_\_\_\_\_ (ім'я та прізвище)

« \_\_\_\_ » \_\_\_\_\_ 2021 р.



Алгоритм роботи застосунку

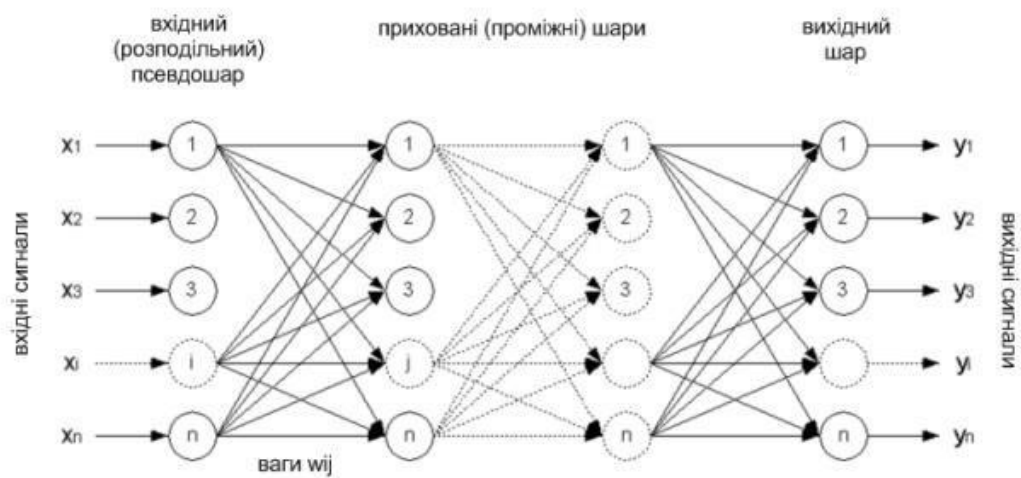


Архітектура ERP- подібного застосунку для голосового управління





Схема технології для розпізнавання усного мовлення



Архітектура нейромережі

```

speech ~/dev/study/speech
├── flask_app
│   └── app
│       ├── mod_interface
│       ├── mod_recognition
│       ├── static
│       └── templates
│       ├── __init__.py
│       ├── requirements.py
│       └── run.py
│       ├── static
│       └── templates

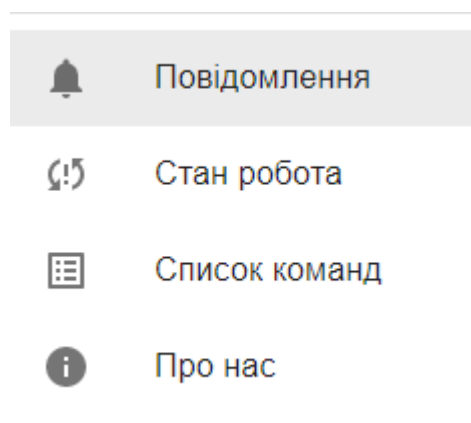
```

## Структура власного ERP- подібного застосунку для голосового управління

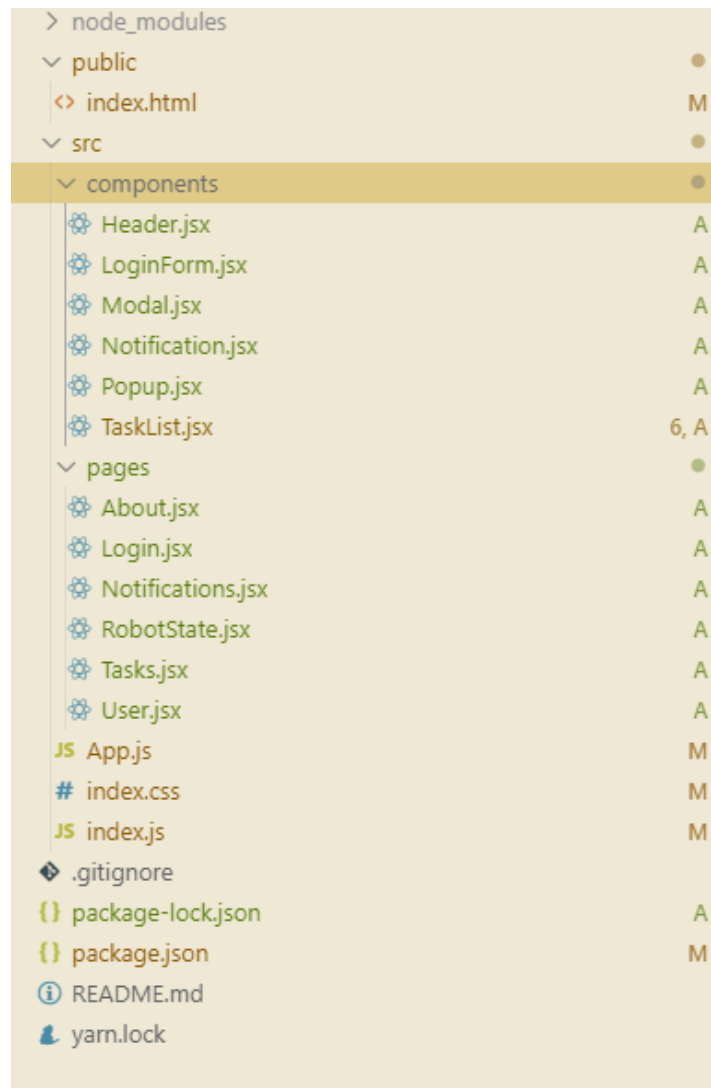
```
class Recognizer:
    def __init__(self):
        self.audio = PyAudio()
        self.recognizer = speech_recognition.Recognizer()
        self.microphone = speech_recognition.Microphone()
        self.speech = pyttsx3.init(debug=True)
        self.listening = False
        self.language = 'en-US'
        self.recognized = ''

    def listen_and_record(self):
        if not self.listening:
            self.listening = True
            with self.microphone as mic:
                while True:
                    try:
                        print('Adjusting')
                        self.recognizer.adjust_for_ambient_noise(mic)
                        print('Listening')
                        res = self.recognizer.listen(mic, timeout=3, phrase_time_limit=3)
                        if res:
                            print('Recognizing')
                            recognized = self.recognizer.recognize_google(res,
                                                                           language=self.language)
                            if recognized:
                                self.recognized = recognized
```

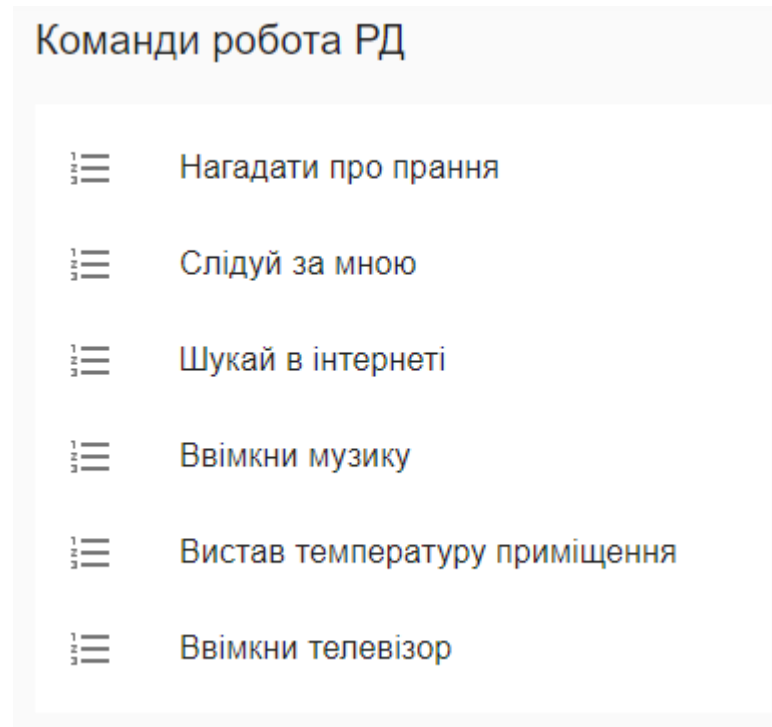
Фрагмент коду з реалізацією голосового управління



Меню-навігатор



React-інтерфейс в вихідному представленні



Перелік сервісів ERP- подібного застосунку для голосового управління