

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього рівня бакалавр

бакалавр

(назва освітнього ступеня)

на тему: **Проектування програмного забезпечення AI агента для телефонного консультування абітурієнтів з використанням технологій Asterisk та Google Live API**

Виконав(ла): студент(ка) 4 курсу, групи СП-41
спеціальності 121

«Інженерія програмного забезпечення»

(шифр і назва спеціальності)

_____ Андрухов І. А.
(підпис) (прізвище та ініціали)

Керівник _____ Багрій-Заяць О.А.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент _____ Стадник М.А.
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

« »

2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Андрухову Ігорю Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування програмного забезпечення AI агента для телефонного консультування абітурієнтів з використанням технологій Asterisk та Google Live API

Керівник роботи Багрій-Заяць Оксана Андріївна, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20__ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Потреба замовника, вимога замовника

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

Аналіз предметної області та постановка задачі

Теоретичні основи та обґрунтування вибору технологій

Реалізації голосового асистента приймальної комісії з підключення API до телефонії

Безпека життєдіяльності та основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності та основи охорони праці	Мариненко С.Ю.		
Нормоконтроль	Стоянов Ю.М.		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з завданням кваліфікаційної роботи	13.04.2026	
2	Збір та аналіз інформації за темою дослідження (огляд існуючих голосових асистентів, IVR-систем)	13.04.2026-16.04.2026	
3	Формування структури пояснювальної записки	17.04.2026	
4	Розробка технічного завдання та вибір методів реалізації	20.04.2026	
5	Реалізація інтелектуального голосового асистента: налаштування Asterisk, розробка серверної частини, інтеграція з API Gemini та телефонією Zadarma	21.04.2026 - 30.04.2026	
6	Тестування функціоналу голосового асистента в реальних умовах та оптимізація затримок	01.05.2026 - 03.05.2026	
7	Написання розділів пояснювальної записки (1–3 розділи)	05.05.2026	
8	Написання розділу 4: «Безпека життєдіяльності та основи охорони праці»	07.05.2026 - 09.05.2026	
9	Оформлення висновків, списку використаних джерел, додатків	11.05.2026 - 14.05.2026	
10	Перевірка роботи керівником, внесення правок		
11	Нормоконтроль		
12	Перевірка кваліфікаційної роботи на плагіат		
13	Попередній захист кваліфікаційної роботи		
14	Захист кваліфікаційної роботи		

Студент

_____ (підпис)

Андрухов І.А.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Багрій-Заяць О.А.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Андрухов І.А. Проектування програмного забезпечення AI агента для телефонного консультування абітурієнтів з використанням технологій Asterisk та Google Live API, ТНТУ ім. Івана Пулюя, Тернопіль 2026.

Пояснювальна записка містить: 66 сторінок, 19 рисунків, 25 джерел та 3 додатки.

Об'єкт дослідження: процес розробки та інтеграції інтелектуального голосового асистента для обробки телефонних запитів користувачів у режимі реального часу.

Мета: розробити повнофункціональну програмно-апаратну систему, яка дозволяє автоматизувати процес телефонного консультування абітурієнтів, зменшити навантаження на працівників приймальної комісії та забезпечити високу якість обслуговування завдяки природній взаємодії.

Сучасні заклади вищої освіти стикаються з великим обсягом однотипних запитів під час вступних кампаній. Використання традиційних систем голосового меню (IVR) часто є недостатньо гнучким інструментом. Особливої актуальності набуває впровадження інтелектуальних голосових агентів на базі сучасних мультимодальних моделей штучного інтелекту, які здатні вести живий діалог, розпізнавати наміри користувача та адаптуватися до контексту розмови. У межах цієї роботи розглянуто процес створення голосового асистента з інтеграцією моделі Google Gemini (Native Audio) та відкритої IP-АТС Asterisk. Реалізовано функціонал двосторонньої потокової передачі та ресемплінгу аудіоданих через інтерфейс AudioSocket і WebSockets, налаштовано системну логіку консультування для приймальної комісії ТНТУ, а також розроблено модуль логування та транскрибації діалогів.

Ключові слова: голосовий асистент, штучний інтелект, IP-телефонія, Asterisk, Google Gemini, AudioSocket, Python, приймальна комісія, розпізнавання мовлення, обробка природної мови.

ABSTRACT

Andrukhov, I.A. Designing AI Agent Software for Telephone Counseling of College Applicants Using Asterisk and Google Live API Technologies, Ivan Pul'uj Ternopil National Technical University, Ternopil 2026.

The explanatory note contains: 66 pages, 19 figures, 25 references, and 3 appendices.

Research object: the process of developing and integrating an intelligent voice assistant for processing user telephone inquiries in real time.

Objective: to develop a fully functional software-hardware system that automates the process of telephone counseling for applicants, reduces the workload on admissions committee staff, and ensures high-quality service through natural interaction.

Modern higher education institutions face a large volume of repetitive inquiries during admissions campaigns. The use of traditional interactive voice response (IVR) systems is often not a flexible enough tool. The implementation of intelligent voice agents based on modern multimodal artificial intelligence models, which are capable of conducting a live dialogue, recognizing user intent, and adapting to the context of the conversation, is becoming particularly relevant. This work examines the process of creating a voice assistant by integrating the Google Gemini (Native Audio) model with the open-source Asterisk IP-PBX. Functionality for two-way streaming and resampling of audio data via the AudioSocket and WebSockets interfaces has been implemented, and the system logic for advising the TNTU Admissions Committee has been configured.

ПЕРЕЛІК СКОРОЧЕНЬ

AI (Artificial Intelligence) – Штучний інтелект

API (Application Programming Interface) – Прикладний програмний інтерфейс

IVR (Interactive Voice Response) – Система інтерактивної голосової відповіді

JSON (JavaScript Object Notation) – Текстовий формат обміну даними, заснований на JavaScript

LLM (Large Language Model) – Велика мовна модель

PBX (Private Branch Exchange) – Приватна телефонна станція (у роботі використовується як синонім до телефонної станції / IP-АТС)

PCM (Pulse-Code Modulation) – Імпульсно-кодова модуляція

RTP (Real-time Transport Protocol) – Протокол передачі даних у реальному часі

SIP (Session Initiation Protocol) – Протокол ініціювання сеансу

UUID (Universally Unique Identifier) – Універсальний унікальний ідентифікатор

ЗМІСТ

ВСТУП.....	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Аналіз процесу інформаційної підтримки абітурієнтів під час вступної кампанії.....	11
1.2 Огляд існуючих систем телефонного самообслуговування та їхні обмеження.....	13
1.3 Дослідження методів побудови розмовних асистентів на базі штучного інтелекту	15
1.4 Огляд технологій інтеграції IP-телефонії з програмними агентами	17
1.5 Постановка задачі на розробку програмно-апаратного комплексу .	20
2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	22
2.1 Проектування загальної архітектури програмного та апаратного комплексу.....	22
2.2 Розробка модуля взаємодії з платформою телефонії	25
2.3 Реалізація серверної частини та алгоритмів цифрової обробки звуку	29
2.4 Інтеграція хмарного сервісу штучного інтелекту для потокової обробки аудіо.....	31
2.5 Налаштування логіки віртуального асистента та розробка системного промπτу	34
2.6 Створення підсистеми моніторингу та збереження результатів телефонних консультацій.....	36
3 ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОГО ГОЛОСОВОГО АГЕНТА ТА ТЕСТУВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ.....	39

3.1 Розгортання серверної інфраструктури та налаштування мережевої взаємодії компонентів.....	39
3.2 Інтеграція хмарного сервісу штучного інтелекту Gemini Live.....	42
3.3 Програмна реалізація логіки асистента та алгоритмів обробки звуку	45
3.4 Тестування продуктивності системи та аналіз затримок обробки даних.....	48
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	52
4.1 Соціальні та психологічні фактори ризику	52
4.2 Організація безпечної роботи електроустановок.....	55
ВИСНОВКИ.....	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ.....	63
ДОДАТОК А	64
ДОДАТОК Б.....	65
ДОДАТОК В	66

ВСТУП

Сучасний етап розвитку цифрових технологій характеризується стрімким впровадженням систем штучного інтелекту в усі сфери суспільного життя, зокрема у сферу освіти та надання консультаційних послуг. Під час проведення щорічних вступних кампаній заклади вищої освіти стикаються з піковими навантаженнями на операторів приймальних комісій. Велика кількість телефонних звернень від абітурієнтів стосується типових питань: термінів подачі документів, вартості навчання, переліку спеціальностей та правил вступу.

Традиційні методи автоматизації, такі як системи інтерактивного голосового меню, мають суттєві обмеження. Вони вимагають від користувача прослуховування довгих списків опцій, не здатні розпізнавати природне мовлення та часто викликають незадоволення абонентів. З іншого боку, поява новітніх мультимодальних великих мовних моделей, здатних обробляти аудіодані в режимі реального часу, відкриває нові можливості для створення інтелектуальних голосових асистентів. Такі системи можуть вести повноцінний живий діалог, адаптуватися до контексту та миттєво надавати релевантну інформацію.

З огляду на це, розробка програмного та апаратного комплексу на базі сучасної телефонії Asterisk та генеративного штучного інтелекту для автоматизації консультацій у приймальній комісії Тернопільського національного технічного університету імені Івана Пулюя є своєчасним та актуальним завданням, що дозволить підвищити якість обслуговування вступників та оптимізувати роботу персоналу.

Метою кваліфікаційної роботи є створення повнофункціонального інтелектуального голосового асистента для автоматизації процесу обробки телефонних запитів абітурієнтів у режимі реального часу.

Для досягнення поставленої мети було сформульовано та вирішено такі завдання: проведення аналізу предметної області та існуючих рішень у сфері автоматизації телефонних консультацій; проектування загальної архітектури

системи взаємодії платформи Asterisk з хмарними сервісами штучного інтелекту; розробка серверної частини додатку для обробки двосторонніх потокових аудіоданих з використанням асинхронного програмування та протоколу WebSockets; реалізація алгоритмів цифрової обробки сигналів для забезпечення сумісності форматів телефонії та мовних моделей; налаштування системного промπτу та логіки ведення діалогу віртуального асистента згідно з інформаційною базою приймальної комісії ТНТУ; проведення тестування розробленої системи в реальних умовах експлуатації та оцінка її ефективності.

Об'єкт дослідження: процес автоматизованого розпізнавання, обробки та генерації голосових відповідей на запити користувачів у телефонних мережах.

Предмет дослідження: методи, моделі штучного інтелекту та програмні засоби розробки інтелектуального голосового асистента з використанням платформи Asterisk та Google Live API.

Методи дослідження. У процесі виконання роботи використовувались методи системного аналізу для проектування архітектури програмного комплексу. Також застосовувались методи об'єктно орієнтованого та асинхронного програмування для розробки серверної частини. Для перетворення частоти дискретизації звукового потоку залучались методи цифрової обробки сигналів. Налаштування поведінки мовної моделі здійснювалось за допомогою методів інженерії підказок Prompt Engineering.

Практичне значення одержаних результатів. Розроблений інтелектуальний голосовий асистент Емілі є готовим до впровадження програмним продуктом. Система здатна приймати вхідні телефонні дзвінки, розпізнавати українську мову, надавати актуальну інформацію про спеціальності, правила вступу та контакти ТНТУ у форматі живого діалогу. Впровадження цієї розробки у роботу приймальної комісії дозволить суттєво розвантажити операторів від рутинних консультацій, забезпечивши при цьому високу швидкість та якість обслуговування абітурієнтів.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Сфера надання освітніх послуг та взаємодії закладів вищої освіти з абітурієнтами зазнає суттєвих трансформацій завдяки стрімкому розвитку інформаційних технологій. Традиційні системи інтерактивного голосового меню які тривалий час використовувалися для автоматизації телефонних звернень поступово втрачають свою ефективність через надмірну жорсткість алгоритмів та нездатність адаптуватися до потреб співрозмовника. Сучасні вступники очікують миттєвого та персоналізованого зворотного зв'язку що вимагає переходу до якісно нових рішень на базі мультимодальних алгоритмів штучного інтелекту. У даному розділі проводиться глибокий аналіз предметної області та досліджується процес інформаційної підтримки потенційних студентів під час пікових навантажень вступної кампанії. Детально розглядаються існуючі підходи до побудови розмовних асистентів та технології інтеграції класичної телефонії з новітніми хмарними мовними моделями. На основі зібраної аналітичної інформації формулюється чітка постановка задачі на розробку інтелектуального програмно апаратного комплексу який здатен вести живий діалог у режимі реального часу.

1.1 Аналіз процесу інформаційної підтримки абітурієнтів під час вступної кампанії

Вступна кампанія є одним із найважливіших та найнапруженіших етапів у річному циклі функціонування закладів вищої освіти. У цей період відбувається масова взаємодія між університетом та потенційними студентами. Головним завданням стає забезпечення безперервної та якісної інформаційної підтримки. Вступники та їхні батьки потребують оперативних консультацій щодо умов прийому, наявності бюджетних місць, вартості навчання та переліку конкурсних предметів. Якість і швидкість надання такої інформації безпосередньо

впливають на рішення абітурієнта обрати саме цей навчальний заклад для здобуття вищої освіти.

Приймальна комісія Тернопільського національного технічного університету імені Івана Пулюя під час вступної кампанії працює в умовах екстремального навантаження. Працівники комісії виконують роль своєрідного контактного центру та змушені щоденно обробляти величезний масив вхідних звернень. Для комунікації використовуються різноманітні канали, такі як електронна пошта, месенджери та офіційні сторінки в соціальних мережах. Проте найпопулярнішим і найбільш затребуваними способом зв'язку залишається класична телефонія. Жива розмова дозволяє вступнику швидко уточнити всі деталі та отримати персоналізовану відповідь у режимі реального часу.

Велика популярність телефонних консультацій створює значні організаційні та технічні труднощі. У пікові дні вступної кампанії комунікаційні лінії університету часто виявляються перевантаженими. Абоненти змушені тривалий час очікувати на з'єднання з вільним оператором. Частина викликів залишається без відповіді через фізичну неможливість працівників обробити всі запити одночасно. Така ситуація здатна призвести до втрати потенційних студентів та зниження загального рівня задоволеності роботою приймальної комісії.

Варто зазначити, що переважна більшість телефонних звернень має типовий характер. Вступники здебільшого цікавляться стандартними питаннями, відповіді на які регламентовані правилами прийому. Необхідність безперервно повторювати ідентичну інформацію призводить до швидкого емоційного вигорання співробітників. Рутинне навантаження знижує загальну ефективність персоналу та забирає час, який фахівці могли б приділити вирішенню нестандартних ситуацій або роботі з документами.

Отже, наявна модель інформаційної підтримки потребує модернізації та залучення інноваційних підходів. Виникає об'єктивна необхідність у впровадженні методів автоматизації процесу обробки вхідних телефонних

дзвінків. Передача рутинних консультаційних завдань інтелектуальним програмним системам дозволить оптимізувати роботу приймальної комісії, підвищити пропускну здатність комунікаційних каналів та вивести рівень обслуговування абітурієнтів на якісно новий рівень.

1.2 Огляд існуючих систем телефонного самообслуговування та їхні обмеження

Історичний розвиток систем телефонного самообслуговування демонструє фундаментальний перехід від жорстко структурованих механізмів до адаптивних когнітивних інтерфейсів на базі алгоритмів штучного інтелекту. Первинні системи інтерактивної голосової відповіді базувалися на використанні попередньо записаних диктором голосових меню де абонент взаємодіяв з системою виключно через натискання клавіш. Логіка прийняття рішень у таких системах була абсолютно статичною та деревоподібною. Будь яке відхилення від запрограмованого сценарію неминуче призводило до неможливості надання сервісу. Еволюційним кроком стало впровадження розмовних систем які імплементували базові технології автоматичного розпізнавання мовлення дозволяючи використовувати природні фрази замість фізичних клавіш. Однак найбільш досконалою категорією є сучасні інтелектуальні голосові боти здатні глибоко розуміти контекст зберігати довгострокову пам'ять про попередні етапи діалогу та автономно вирішувати комплексні завдання користувачів [15, 19].

Для наочного подання архітектурних та функціональних відмінностей між цими поколіннями комунікаційних систем розроблено структурну діаграму еволюції яку наведено на рисунку 1.1.

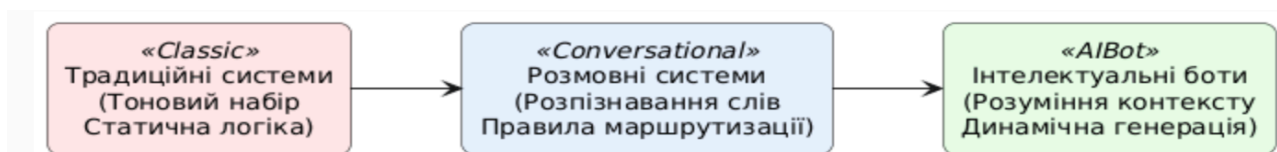


Рисунок 1.1 – Еволюція систем телефонного самообслуговування

Внутрішня архітектура сучасних інтелектуальних комунікаційних платформ є багат шаровою та складається з послідовного конвеєра високотехнологічних компонентів. Базовий життєвий цикл обробки виклику розпочинається з автоматичного розпізнавання мовлення яке перетворює акустичні хвилі на текстовий формат. Після успішної транскрипції масив даних передається до модуля розуміння природної мови завданням якого є семантичне декодування потоку слів та виявлення істинного наміру користувача. Центральним координаційним ядром усієї системи виступає модуль управління діалогом який керує станом розмови та приймає стратегічні рішення. Фінальними етапами є генерація релевантної текстової відповіді за допомогою великих мовних моделей та синтез тексту назад у людське мовлення з природними інтонаціями.

Попри очевидні переваги розгортання таких систем у корпоративних мережах супроводжується комплексними технологічними бар'єрами. Найбільшим прихованим обмеженням є архітектурна несумісність між класичними протоколами телекомунікаційної індустрії та сучасними веб архітектурами систем машинного навчання [14]. Застаріла телефонія використовує протоколи які оптимізовані виключно для маршрутизації людських викликів з припустимим рівнем втрати пакетів. Натомість рушії штучного інтелекту вимагають неперервних потоків високоякісного аудіо через двонаправлені з'єднання веб сокетів.

Основні технічні бар'єри та точки конфлікту протоколів відображено на структурній схемі інтеграції рисунок 1.2.

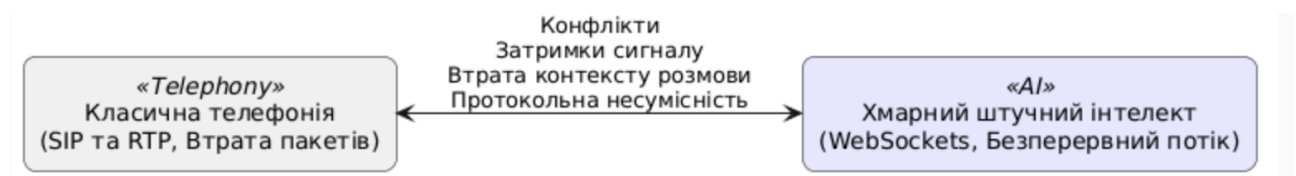


Рисунок 1.2 - Архітектурні конфлікти інтеграції штучного інтелекту в класичну телефонію

Специфіка розробки систем телефонного самообслуговування в Україні має власні унікальні лінгвістичні та соціальні виклики. Головною перешкодою є нестача якісних розмічених акустичних даних необхідних для навчання нейромереж розпізнавати українську мову з усіма її діалектами. Це змушує вітчизняну наукову спільноту розробляти компактні спеціалізовані моделі здатні працювати з високою точністю на локальних серверах. Одночасно гострий дефіцит кадрових ресурсів перетворив масову роботизацію контакт центрів на єдину можливу стратегію збереження стійкості українського бізнесу. Вітчизняні розробники та телекомунікаційні оператори активно модернізують інфраструктуру, створюючи надійний фундамент для повної інтеграції інтелектуальних систем самообслуговування наступного покоління. Це сприяє підвищенню якості обслуговування клієнтів, зменшенню операційних витрат компаній та пришвидшенню впровадження сучасних цифрових сервісів у різних сферах діяльності.

1.3 Дослідження методів побудови розмовних асистентів на базі штучного інтелекту

Створення сучасних розмовних асистентів вимагає застосування комплексного підходу до обробки природної мови та генерації аудіо. Традиційні методи передбачали використання роздільних незалежних модулів для розпізнавання мовлення генерації тексту та синтезу голосу. Така каскадна архітектура часто призводила до накопичення затримок на кожному етапі обробки інформації що руйнувало ілюзію живого спілкування. Сучасний етап технологічного розвитку характеризується переходом до використання наскрізних мультимодальних моделей здатних одночасно сприймати звук аналізувати контекст та миттєво генерувати голосову відповідь. Це дозволяє досягти рівня природності діалогу який раніше був абсолютно недоступний для комп'ютерних систем автоматизації.

На глобальному ринку існує значна кількість готових комерційних платформ для побудови голосових рішень. Яскравим прикладом є популярна комунікаційна платформа Twilio яка надає зручні програмні інтерфейси для управління телефонними дзвінками та маршрутизації викликів. Для генерації максимально реалістичного голосу розробники часто інтегрують у такі платформи сервіси компанії ElevenLabs. Ця спеціалізована нейромережа здатна синтезувати мовлення з глибоким емоційним забарвленням мікропаузами та інтонаціями живої людини. Проте використання комбінації Twilio та ElevenLabs має суттєвий недолік у вигляді надзвичайно високої вартості кожної хвилини розмови. Для розробки кваліфікаційних проектів або масштабування систем у бюджетних установах такі комерційні хмарні рішення є економічно недоцільними через високий фінансовий поріг входу.

Альтернативним і набагато вигіднішим підходом є самостійне розгортання інфраструктури телефонії. Для отримання зовнішнього телефонного номера та забезпечення стабільного зв'язку з мобільними мережами оптимально використовувати послуги хмарних провайдерів таких як Задарма [8]. Цей сервіс надає віртуальні номери та підтримує підключення через стандартні протоколи ініціювання сеансу дозволяючи легко інтегрувати телефонію у власні програмні розробки. Використання сервісу Задарма значно знижує базові витрати на зв'язок порівняно з масивними глобальними комунікаційними платформами зберігаючи при цьому високу якість передачі голосу.

Серцевиною економічно ефективної архітектури стає програмна телефонна станція Asterisk [13]. Вона дозволяє абсолютно безкоштовно розгорнути потужний вузол обробки дзвінків на власній віртуальній машині надаючи розробнику повний контроль над потоками аудіоданих. Справжній прорив у побудові інтелектуальних асистентів став можливим завдяки появі програмного інтерфейсу Gemini Live API від компанії Google. Ця передова мультимодальна модель здатна приймати потокове аудіо та безпосередньо генерувати голосову відповідь оминаючи проміжні етапи текстового синтезу. Головною і найвагомішою перевагою Gemini Live API є наявність щедрого

безкоштовного рівня використання [1, 2]. Це робить дану технологію ідеальним вибором для проведення академічних досліджень розробки прототипів та впровадження систем телефонного самообслуговування з фактично нульовими витратами на використання потужностей штучного інтелекту.

Синтез описаних безкоштовних та відкритих технологій формує найоптимальніший метод побудови розмовного асистента. Вхідний дзвінок від абітурієнта надходить через мережу Задарма на сервер Asterisk. Далі телефонна станція через сокетне з'єднання передає безперервний аудіопотік до серверного додатка написаного мовою Python. Цей додаток виступає інтелектуальною сполучною ланкою здійснюючи необхідне перетворення частоти дискретизації звуку та обмінюючись даними з Gemini Live API у режимі реального часу. Така конвеєрна архітектура не лише забезпечує високу швидкість реакції системи але й гарантує повну незалежність від дорогих комерційних сервісів перетворення тексту в голос формуючи еталонну базу для виконання практичної частини кваліфікаційної роботи.



Рисунок 1.3 – Архітектура побудови розмовного агента на базі відкритих технологій

1.4 Огляд технологій інтеграції IP-телефонії з програмними агентами

Інтеграція класичної телефонії з сучасними програмними агентами вимагає вирішення складної архітектурної задачі узгодження різних мережевих протоколів. Традиційні телефонні мережі функціонують на базі протоколів ініціювання сеансу та передачі даних у реальному часі які оптимізовані для маршрутизації голосового трафіку з мінімальними затримками але допускають втрату частини пакетів. Натомість хмарні системи штучного інтелекту вимагають гарантованої доставки безперервного потоку високоякісних

аудіоданих що зазвичай реалізується через повноцінні двосторонні з'єднання за допомогою технології веб сокетів. Для подолання цієї технічної несумісності розроблено кілька концептуальних підходів включаючи використання спеціалізованих медіа шлюзів прикордонних контролерів сесій та модулів прямої потокової передачі аудіо.

Базовим елементом розробленого апаратного та програмного комплексу виступає відкрита телефонна станція Asterisk розгорнута на віртуальній машині. Ця платформа забезпечує гнучке управління викликами та безпосередню маршрутизацію телефонного трафіку. Для забезпечення стабільного зв'язку із зовнішніми абонентами мобільних та стаціонарних мереж використано послуги хмарного провайдера Zadarma. Він надає віртуальний телефонний номер та забезпечує надійне підключення до локальної станції Asterisk через транк ініціювання сеансу. Такий підхід дозволяє ізолювати складну логіку обробки телекомунікаційної сигналізації від інтелектуальної частини системи гарантуючи безперебійний прийом дзвінків від абітурієнтів університету.

Критично важливим етапом інтеграції є безпечна та швидка передача голосового потоку від телефонної станції до керуючого додатка штучного інтелекту. У рамках даної кваліфікаційної роботи для вирішення цього завдання застосовано технологію AudioSocket [4]. На відміну від класичних інтерфейсів управління які передають виключно текстові команди або працюють із записаними файлами на жорсткому диску ця інноваційна технологія дозволяє транслювати сирий нестиснутий аудіопотік безпосередньо з плану набору номерів Asterisk до зовнішнього мережевого порту. Це суттєво мінімізує мережеві затримки та забезпечує можливість обробки звуку в режимі реального часу що є абсолютно необхідною умовою для створення ефекту живого спілкування з віртуальним асистентом.

Серверна частина розробленого комплексу реалізована мовою програмування Python з інтенсивним використанням парадигми асинхронного програмування [3]. Для ефективною обробки великої кількості одночасних підключень застосовано стандартну бібліотеку асинхронного введення та

виведення даних у поєднанні з високопродуктивним циклом подій. Додатково серверний додаток інтегрує сучасний веб фреймворк FastAPI для моніторингу стану системи та підрахунку кількості активних комунікаційних сесій [11]. Асинхронна архітектура дозволяє додатку одночасно читати звуковий потік від телефонної станції відправляти його до мовної моделі та приймати згенеровану відповідь не блокуючи виконання інших критичних процесів операційної системи.

Окремою складною технічною проблемою успішно вирішеною в ході реалізації практичної частини проекту стала розбіжність форматів цифрового звуку. Телефонна мережа за замовчуванням працює з частотою дискретизації вісім кілогерц що є базовим стандартом для класичного голосового зв'язку. Проте сучасні мультимодальні моделі штучного інтелекту вимагають значно вищої якості аудіо з частотою двадцять чотири кілогерци. Для подолання цього бар'єру в програмному коді реалізовано алгоритм швидкого перетворення частоти дискретизації на льоту. За допомогою спеціалізованих математичних бібліотек NumPy та SciPy вхідний телефонний звук масштабується до необхідної якості перед відправкою до нейромережі а згенерована інтелектуальна відповідь проходить зворотне звуження перед подачею назад у телефонний канал [9, 10].

Головним інтелектуальним ядром розробленої системи виступає мультимодальна велика мовна модель Gemini від компанії Google інтегрована через прикладний програмний інтерфейс двосторонньої генерації контенту. Ця передова хмарна технологія дозволяє встановити постійне захищене з'єднання через веб сокети та передавати звукові фрагменти безпосередньо в нейромережу без проміжного та повільного перетворення їх у текстовий формат [12]. Модель здатна миттєво аналізувати отриманий аудіо контекст враховуючи задані системні інструкції щодо специфічної поведінки асистента приймальної комісії та генерувати синтезовану голосову відповідь яка одразу розбивається на пакети та надсилається абоненту. Таке наскрізне архітектурне рішення забезпечує безпрецедентно низьку затримку реакції та дозволяє досягти максимальної природності діалогу з потенційним студентом.

1.5 Постановка задачі на розробку програмно-апаратного комплексу

На основі проведеного аналізу предметної області та детального огляду існуючих технологічних рішень виникає об'єктивна необхідність у формулюванні чіткої технічної задачі для виконання практичної частини кваліфікаційної роботи. Головною метою проекту є розробка та впровадження інтелектуального голосового асистента для автоматизації роботи приймальної комісії Тернопільського національного технічного університету імені Івана Пулюя. Розроблена система повинна вирішити проблему перевантаження операторів під час вступної кампанії шляхом надання миттєвих та точних відповідей на типові запитання абітурієнтів у форматі природного живого діалогу.

Для досягнення поставленої мети необхідно виконати комплекс інженерних та алгоритмічних завдань. Першочерговим кроком є проектування загальної архітектури програмного та апаратного комплексу яка ефективно поєднає класичну телефонію та сучасні хмарні сервіси штучного інтелекту. На рівні телекомунікаційної інфраструктури потрібно розгорнути відкриту телефонну станцію Asterisk на віртуальній машині та налаштувати її взаємодію з хмарним провайдером Задарма. Це підключення має здійснюватися через протокол ініціювання сеансу для забезпечення стабільного прийому зовнішніх телефонних дзвінків від вступників.

Наступним критичним завданням є розробка серверного додатка мовою програмування Python з використанням парадигми асинхронного програмування. Цей додаток повинен приймати безперервний аудіопотік від телефонної станції через спеціалізований інтерфейс AudioSocket. Окремою технічною вимогою є реалізація математичних алгоритмів цифрової обробки сигналів для перетворення частоти дискретизації звуку на льоту. Така необхідність зумовлена тим що стандартна телефонна мережа і сучасна нейромережа працюють з абсолютно різними показниками якості аудіоданих і потребують програмного узгодження.

Основна інтелектуальна логіка системи вимагає прямої інтеграції серверного додатка з мультимодальною великою мовною моделлю Gemini від компанії Google через інтерфейс прикладного програмування двосторонньої генерації контенту. Для забезпечення коректної роботи асистента необхідно розробити детальний системний промпт який задасть віртуальному консультанту на ім'я Емілі відповідну роль визначить правила ведення діалогу та жорстко обмежить тематику розмов виключно питаннями вступу до університету. Також система повинна вміти аналізувати контекст бесіди та автономно завершувати дзвінок після закінчення консультації за допомогою виклику відповідних функцій прикладного інтерфейсу.

Завершальним етапом розробки є створення підсистеми моніторингу та логування подій.

2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

Проектування архітектури програмного забезпечення вимагає чіткого узгодження різноманітних мережевих протоколів та створення надійного конвеєра обробки даних. Перехід від теоретичної моделі до практичної реалізації голосового асистента передбачає вирішення комплексу складних інженерних завдань пов'язаних із потоковою трансляцією аудіо сигналів. Класична телефонія та сучасні нейромережі використовують принципово різні формати цифрового звуку тому ключовим викликом є створення ефективної проміжної ланки для їхньої математичної синхронізації. Цей розділ повністю присвячений процесу проектування та розробки програмної системи розумного помічника приймальної комісії. У ньому детально розкриваються механізми побудови модульної архітектури налаштування маршрутизації викликів через телефонну станцію Asterisk та написання асинхронного серверного коду мовою програмування Python. Окремо висвітлюється математична реалізація алгоритмів перетворення частоти дискретизації звуку а також процес глибокої інтеграції хмарного сервісу штучного інтелекту із застосуванням спеціалізованого системного промπτу для формування правильної логіки ведення діалогу.

2.1 Проектування загальної архітектури програмного та апаратного комплексу

Проектування архітектури є найважливішим етапом створення будь якої складної інформаційної системи. Запропонований програмний та апаратний комплекс побудований за модульним принципом і складається з чотирьох основних компонентів які функціонують у тісній взаємодії. Головною метою такої архітектури є забезпечення безперебійного двостороннього аудіозв'язку між звичайним телефонним абонентом та хмарною мовною моделлю штучного

інтелекту з мінімальними мережевими затримками. Кожен вузол системи виконує свою вузькоспеціалізовану функцію утворюючи єдиний неперервний конвеєр обробки даних у режимі реального часу. Такий чіткий розподіл обов'язків гарантує високу відмовостійкість усього комплексу оскільки збій в одному модулі може бути локалізований та усунений без повної зупинки інших процесів. Окрім того модульна структура закладає надійний фундамент для майбутнього горизонтального масштабування дозволяючи при необхідності легко збільшувати обчислювальні потужності серверної частини під час пікових навантажень у розпал вступної кампанії.

Першою ланкою системи є хмарний провайдер телефонії Задарма який надає зовнішній телефонний номер для прийому дзвінків від абітурієнтів. Коли користувач здійснює виклик провайдер направляє його через стандартні протоколи ініціювання сеансу на віртуальну телефонну станцію Asterisk. Ця телефонна станція розгорнута на окремій віртуальній машині і виступає центральним комутаційним вузлом. Вона приймає вхідний телефонний трафік відповідає на виклик і згідно з налаштованим планом маршрутизації направляє його до локального програмного обробника. Важливою особливістю конфігурації є використання спеціалізованого протоколу AudioSocket який дозволяє телефонній станції відкрити пряме мережеве з'єднання та почати трансляцію сирого нестиснутого звуку без збереження його у проміжні файли на жорсткому диску. Відмова від класичних скриптів управління дзвінками на користь прямої потокової передачі байтів дозволила повністю усунути системні затримки пов'язані з читанням та записом інформації на фізичні носії. Це інженерне рішення також суттєво знижує навантаження на підсистему пам'яті віртуальної машини вивільняючи ресурси для миттєвої маршрутизації великої кількості одночасних викликів.

Наступним і технологічно найскладнішим компонентом архітектури є серверний додаток написаний мовою програмування Python. Цей модуль працює як асинхронний мережевий сервер постійно очікуючи нових підключень від телефонної станції на визначеному порту. Отримавши звуковий потік додаток

виконує роль проміжного інтелектуального шлюзу. Оскільки стандарти телефонного зв'язку і вимоги сучасних нейромереж суттєво відрізняються серверний обробник здійснює складне математичне перетворення частоти дискретизації звуку. Вхідний аудіосигнал масштабується для забезпечення високої якості розпізнавання а згенерована відповідь проходить зворотне звуження перед відправкою абоненту. Усі ці розрахункові операції виконуються паралельно завдяки асинхронній природі додатку. Використання сучасного циклу подій дозволяє серверу ефективно керувати тимчасовими буферами аудіоданих запобігаючи переповненню оперативної пам'яті під час тривалих розмов. Додатково на цьому програмному рівні реалізовано базові алгоритми компенсації втрачених мережевих пакетів які гарантують цілісність звукової хвилі навіть за умов нестабільного інтернет з'єднання між сервером та телефонним провайдером.

Останньою ланкою конвеєра є хмарний сервіс штучного інтелекту Gemini Live API від компанії Google. Серверний обробник встановлює з ним безпечне з'єднання через протокол веб сокетів. Цей канал зв'язку залишається відкритим протягом усієї телефонної розмови дозволяючи здійснювати двосторонню потокову передачу даних. Нейромережа приймає підготовлені звукові фрагменти аналізує їх враховуючи заданий системний промпт про приймальну комісію та миттєво повертає синтезовані голосові відповіді. Перед відправкою до хмарного сервісу аудіо додатково кодується у спеціальний текстовий формат що забезпечує надійність передачі через протоколи глобальної мережі. Після отримання цих відповідей серверний додаток адаптує їх під телефонний формат і відправляє назад до телефонної станції яка транслює голос абоненту завершуючи таким чином повний цикл архітектурної взаємодії. Глибока інтеграція системного контексту безпосередньо у сеанс зв'язку дозволяє мовній моделі не лише механічно відповідати на поточні питання але й бездоганно підтримувати логіку тривалої бесіди запам'ятовуючи попередні репліки користувача та формуючи максимально природний діалог без залучення живих операторів.

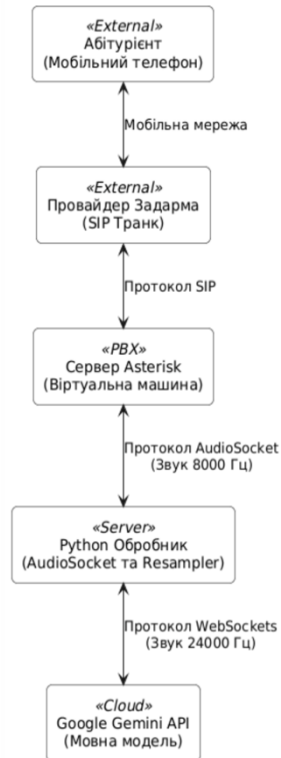


Рисунок 2.1 Структурна схема взаємодії компонентів програмного та апаратного комплексу.

2.2 Розробка модуля взаємодії з платформою телефонії

Розробка модуля взаємодії з платформою телефонії є надзвичайно важливим етапом створення загальної діалогової системи. Першочерговим завданням на цьому етапі стало налаштування підключення віртуальної телефонної станції Asterisk до зовнішнього провайдера зв'язку Задарма. Для забезпечення стабільного прийому вхідних дзвінків від абітурієнтів було створено спеціальний транк зв'язку використовуючи стандартні мережеві протоколи ініціювання сеансу. У конфігураційних файлах телефонної станції вказано облікові дані сервера провайдера параметри авторизації та правила мережевої реєстрації. Це дозволяє віртуальній станції постійно підтримувати активний статус у системі провайдера і миттєво реагувати на будь які вхідні виклики що надходять на орендований телефонний номер університету.

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[zadarma-reg]
type=registration
transport=transport-udp
outbound_auth=zadarma-auth
server_uri=sip:sip.zadarma.com
client_uri=sip:796569@sip.zadarma.com
retry_interval=60
expiration=120

[zadarma-auth]
type=auth
auth_type=userpass
password=nfRt7Xe8VW
username=796569

[zadarma-aor]
type=aor
contact=sip:sip.zadarma.com

[zadarma]
type=endpoint
transport=transport-udp
context=from-zadarma
disallow=all
allow=alaw,ulaw
outbound_auth=zadarma-auth
aors=zadarma-aor
from_user=796569
from_domain=sip.zadarma.com
direct_media=no
```

Рисунок 2.2 – Приклад налаштованого транку

Після успішного встановлення базового мережевого з'єднання необхідно було налаштувати логіку маршрутизації викликів всередині самої телефонної станції. Цей процес реалізується через редагування плану набору номерів який визначає строгую послідовність дій системи при надходженні дзвінка. Коли абітурієнт телефонує на номер приймальної комісії система спочатку автоматично піднімає слухавку встановлюючи активну голосову сесію. Далі план набору номерів негайно перенаправляє цей виклик до спеціалізованого додатка обробки звуку який вбудований в ядро телефонної станції. Саме на цьому етапі відбувається ключовий перехід від класичної телефонної маршрутизації до програмної обробки звукового потоку засобами зовнішнього коду.

```

[general]
static=yes
writeprotect=no
[globals]
[from-zadarma]
exten => _.,1,NoOp(Incoming call from Zadarma)
same => n,Answer()
same => n,Playback(demo-thanks)
same => n,Hangup()
[from-internal]
exten => 100,1,Answer()
same => n,Playback(demo-echotest)
same => n,Echo()
same => n,Hangup()
exten => 555,1,Answer()
same => n,AudioSocket(${UUID()},172.20.10.2:9090)
same => n,Hangup()
exten => _+.,1,NoOp(Calling external number ${EXTEN} via Zadarma)
same => n,Dial(PJSIP/${EXTEN}@zadarma,60)
same => n,Hangup()
exten => _380.,1,NoOp(Adding + to number)
same => n,Goto(+${EXTEN},1)
exten => _0XX.,1,NoOp(Adding +38 to number)
same => n,Goto(+38${EXTEN},1)

```

Рисунок 2.3 – Приклад налаштованої логіки маршрутизації

Основною технологією яка забезпечує передачу голосу до інтелектуальної частини системи є протокол потокової передачі звуку AudioSocket. На відміну від стандартних методів запису розмов у локальні файли ця технологія дозволяє відкрити пряме мережеве з'єднання між телефонною станцією та зовнішнім серверним додатком. Відмова від використання файлової системи дозволяє повністю усунути апаратні затримки пов'язані з операціями введення та виведення даних на фізичний жорсткий диск що є критично важливим для забезпечення ефекту миттєвої реакції віртуального співрозмовника. Як тільки виклик надходить до відповідного модуля в плані набору номерів телефонна станція ініціює з'єднання з сервером на мові програмування Python використовуючи заздалегідь визначений локальний порт мережі. На самому початку сеансу зв'язку станція передає унікальний ідентифікатор виклику який дозволяє серверу відстежувати та логувати конкретного абонента. Цей ідентифікатор стає ключем для формування ізольованої комунікаційної сесії

гарантуючи що аудіодані різних користувачів не перетнуться в оперативній пам'яті сервера під час одночасної обробки кількох телефонних дзвінків. Одразу після успішної передачі ідентифікатора починається безперервна трансляція сирого нестиснутого звукового потоку який надходить безпосередньо з мікрофона мобільного пристрою вступника.

Формат даних який передається через це мережеве з'єднання має строгу інженерну специфікацію. Звук надходить у вигляді лінійної імпульсно кодової модуляції з частотою дискретизації вісім кілогерц що є базовим стандартом для традиційних телефонних мереж. Цей формат не передбачає наявності жодних додаткових часових міток або складних метаданих всередині аудіопакетів тому програмний обробник повинен надзвичайно точно зчитувати байти відповідно до заданих інтервалів щоб уникнути розсинхронізації цифрового сигналу. Серверний додаток написаний мовою Python працює в режимі постійного прослуховування мережевого порту. Використовуючи асинхронні функції він миттєво приймає вхідне з'єднання зчитує початковий ідентифікатор і починає накопичувати звукові байти у спеціальні буфери пам'яті. Розмір цих програмних буферів ретельно оптимізовано для того щоб збирати достатню кількість акустичної інформації для подальшого математичного перетворення але при цьому не створювати штучних затримок у живому діалозі. Одночасно з прийомом звуку від абонента цей самий відкритий мережевий канал постійно використовується для зворотної передачі згенерованого голосу від штучного інтелекту безпосередньо у телефонну лінію утворюючи повноцінний двосторонній канал зв'язку з мінімально можливою затримкою. Завдяки такому підходу архітектура системи стає повністю повнодуплексною дозволяючи серверу слухати людину та транслювати власну синтезовану відповідь абсолютно паралельно і незалежно від стану інших системних процесів.

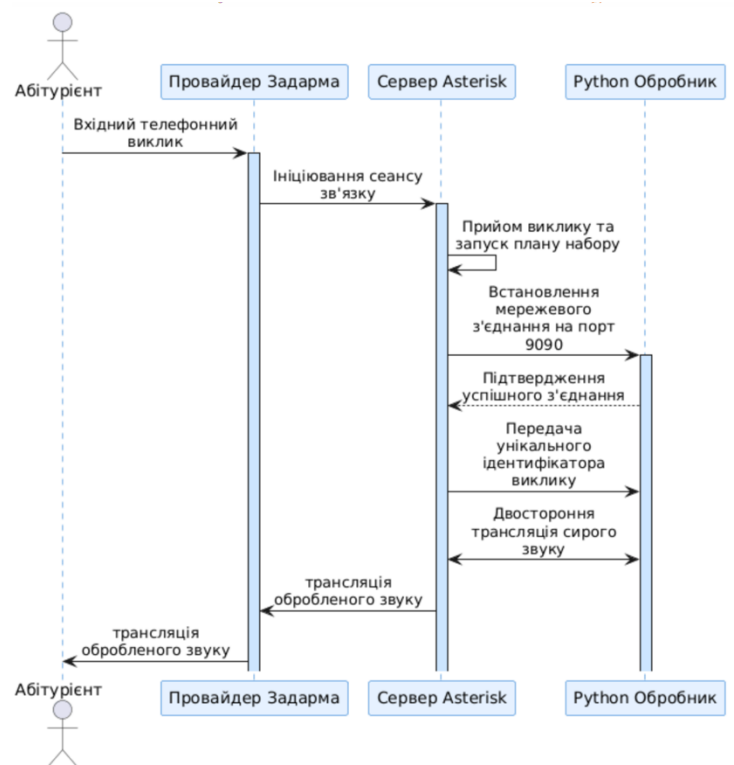


Рисунок 2.4 – Діаграма послідовності взаємодії компонентів під час прийому вхідного дзвінка.

2.3 Реалізація серверної частини та алгоритмів цифрової обробки звуку

Серверна частина розробленого програмного комплексу реалізована мовою програмування Python яка є визнаним галузевим стандартом для створення систем обробки даних та інтеграції алгоритмів штучного інтелекту. Вибір цієї мови зумовлений наявністю потужного екосистемного середовища спеціалізованих математичних бібліотек та розвиненими засобами роботи з мережевими протоколами. Головним архітектурним патерном серверного додатка стало використання парадигми асинхронного програмування. Замість класичного багатопотокового підходу який створює значне навантаження на операційну систему розроблений код використовує єдиний цикл подій бібліотеки асинхронного введення та виведення. Цей підхід дозволяє серверу

одночасно обслуговувати множинні мережеві підключення від телефонної станції та хмарного сервісу штучного інтелекту без взаємного блокування системних процесів.

У межах кожної активної телефонної сесії серверний додаток створює кілька паралельних асинхронних завдань які функціонують незалежно але тісно обмінюються інформацією через внутрішні черги повідомлень. Перше завдання відповідає за безперервне читання вхідного звукового потоку з мережевого порту технології передачі аудіо. Друге завдання паралельно отримує згенеровані голосові відповіді від мовної моделі та відправляє їх назад у телефонну лінію. Третє завдання виконує роль фонового контролера підтримуючи з'єднання активним та обробляючи можливі мережеві помилки. Завдяки такій асинхронній диспетчеризації система досягає максимальної пропускну здатності та мінімізує затримки що є критично важливим для забезпечення ефекту природного спілкування у режимі реального часу.

Однією з найскладніших інженерних проблем під час розробки серверної частини стала необхідність узгодження форматів цифрового звуку між різними компонентами системи. Традиційна телефонія передає голос у форматі лінійної імпульсно кодової модуляції з частотою дискретизації вісім кілогерц. З іншого боку сучасні мультимодальні моделі штучного інтелекту вимагають для коректного розпізнавання значно ширшого спектра частот а саме двадцять чотири кілогерци. Пряма передача непідготовленого телефонного звуку до нейромережі призводить до повної неможливості розпізнавання мовлення через спотворення акустичних характеристик. Тому виникла гостра потреба у реалізації алгоритму перетворення частоти дискретизації звуку безпосередньо в оперативній пам'яті сервера під час його потокової трансляції.

Для вирішення цього завдання у програмному кодї застосовано спеціалізовані математичні бібліотеки швидких обчислень. Алгоритм цифрової обробки сигналів працює наступним чином. Спочатку вхідний масив сирих звукових байтів перетворюється на багатовимірний числовий вектор. Далі функція поліфазного перетворення з модуля обробки сигналів виконує складну

математичну інтерполяцію та децимацію цього вектора [18]. Алгоритм розраховує нові проміжні значення звукової хвилі збільшуючи кількість вибірок рівно у три рази. Після застосування цифрових фільтрів низьких частот для усунення ефекту накладання спектрів оновлений числовий масив упаковується назад у байтовий формат і миттєво відправляється до мовної моделі. Згенерована штучним інтелектом відповідь проходить повністю симетричний зворотний процес звуження частоти перед передачею до телефонної станції.

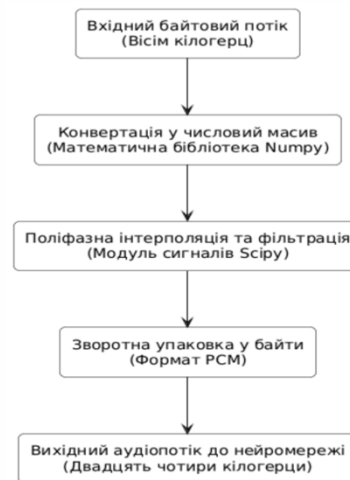


Рисунок 2.5 – Блок схема алгоритму перетворення частоти дискретизації звукового потоку

2.4 Інтеграція хмарного сервісу штучного інтелекту для потокової обробки аудіо

Інтеграція хмарного сервісу штучного інтелекту є завершальним і найважливішим етапом побудови розмовної системи. Для забезпечення роботи віртуального асистента в режимі реального часу використано прикладний програмний інтерфейс Gemini Live API від компанії Google. Зв'язок між локальним сервером та хмарною мовною моделлю реалізується через захищений протокол веб сокетів. Вибір саме цього мережевого протоколу зумовлений його здатністю підтримувати постійне двостороннє з'єднання без необхідності

повторного встановлення сеансу для кожного нового запиту що кардинально відрізняє його від класичних архітектур де кожен запит вимагає створення нового підключення і призводить до критичних мережових затримок. Авторизація в системі відбувається автоматично під час ініціалізації з'єднання за допомогою унікального ключа доступу який передається у рядку запиту гарантуючи високий рівень безпеки та конфіденційності переданих даних. Використання сучасних криптографічних стандартів шифрування на транспортному рівні забезпечує надійний захист голосового трафіку абонентів від перехоплення під час його трансляції через глобальну мережу інтернет до обчислювальних центрів.

Процес взаємодії розпочинається з обов'язкового етапу конфігурації сеансу. Відразу після успішного відкриття мережевого сокета серверний додаток написаний мовою Python відправляє спеціальне конфігураційне повідомлення у форматі об'єктної нотації JavaScript. Цей початковий пакет даних містить системні інструкції які визначають поведінку штучного інтелекту встановлюють контекст приймальної комісії та задають параметри синтезу голосу. Окрім цього на даному етапі системі передається інформація про доступні інструменти управління викликом та суворо вказується необхідність генерувати відповіді виключно в аудіо форматі блокуючи стандартну текстову генерацію. Тільки після отримання підтвердження від хмарного сервісу про успішне застосування базових налаштувань система переходить у режим активного очікування вхідного звукового потоку від телефонної станції утворюючи повністю готовий до роботи інтелектуальний канал зв'язку. Такий строгий алгоритм ініціалізації гарантує що велика мовна модель не почне обробляти голос людини до моменту повного завантаження всіх правил поведінки та обмежень університетської бази знань.

Основний механізм постійного обміну аудіоданими побудований на принципах асинхронної багатозадачності. Серверний додаток безперервно зчитує мікрофонний потік від телефонного абонента перетворює його на частоту двадцять чотири кілогерци кодує у текстовий формат Base64 і відправляє

невеликими фрагментами до неймережі. Конвертація бінарного звуку у символний формат є технологічною вимогою для безпечної передачі медіаданих через текстово орієнтований протокол веб сокетів незважаючи на незначне збільшення загального обсягу переданої інформації. Паралельно у цьому ж відкритому з'єднанні працює інша асинхронна функція яка прослуховує відповіді від мовної моделі. Коли штучний інтелект генерує голосову реакцію він також надсилає її фрагментами у закодованому вигляді не чекаючи завершення формування цілої фрази. Сервер миттєво декодує ці фрагменти звужує частоту до телефонного стандарту та передає у телефонний канал зв'язку. Така наскрізна двостороння трансляція забезпечує надзвичайно низьку затримку дозволяючи системі миттєво реагувати на запити співрозмовника та підтримувати абсолютно природний темп розмови наближаючи машинну комунікацію до живого людського спілкування.

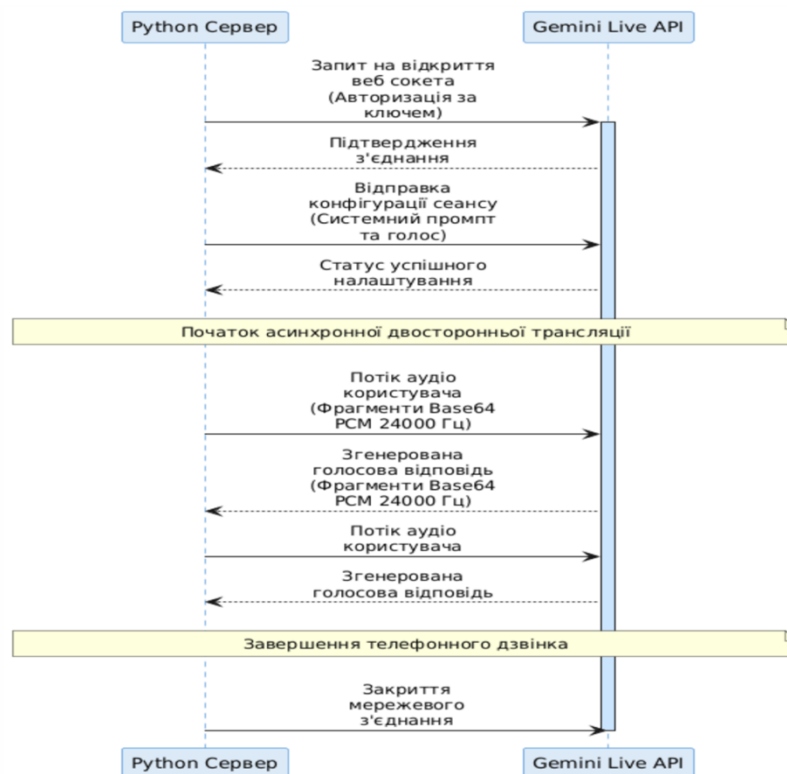


Рисунок 2.6 – Діаграма життєвого циклу з'єднання та потокової передачі даних.

2.5 Налаштування логіки віртуального асистента та розробка системного промπτу

Налаштування логіки віртуального асистента та розробка системного промπτу є фундаментальним етапом створення інтелектуальної діалогової системи. Інженерія підказок виступає основним інструментом програмування поведінки великої мовної моделі визначаючи її реакції на різноманітні запити користувачів. Першочерговим завданням на цьому етапі стало формування унікального характеру віртуального консультанта який отримав ім'я Емілі. Асистент запрограмований на ведення діалогу у ввічливому професійному та емпатичному тоні що максимально відповідає стандартам спілкування працівників приймальної комісії. Особливий акцент зроблено на природності мовлення щоб користувач відчував комфорт під час телефонної розмови не помічаючи штучності згенерованих відповідей та отримуючи приємний досвід комунікації.

Для забезпечення високої точності консультацій у системний промπτ було інтегровано детальний контекст про Тернопільський національний технічний університет імені Івана Пулюя. Цей інформаційний блок містить актуальні дані щодо переліку спеціальностей вартості навчання термінів проведення вступної кампанії та необхідних документів для зарахування. Жорстке обмеження бази знань виключно цією інформацією дозволяє ефективно запобігати виникненню штучних галюцинацій коли нейромережа самостійно вигадує неіснуючі факти. Якщо абітурієнт ставить питання яке виходить за межі наданого контексту асистент запрограмований чесно визнати відсутність інформації та ввічливо порадити звернутися безпосередньо до живого оператора за вказаним контактним номером уникнувши таким чином надання хибних даних.

Окрім інформаційного наповнення системний промπτ містить суворі технічні обмеження щодо формату відповідей. Враховуючи специфіку голосового зв'язку модель проінструкована генерувати максимально лаконічні та чіткі репліки. Довгі розлогі відповіді є неприйнятними для телефонного

формату оскільки вони перевантажують абонента інформацією та збільшують час синтезу аудіо що може призвести до небажаних пауз. Також встановлено жорстке правило відповідати виключно українською мовою ігноруючи спроби перемикання на інші мови. Модель додатково навчена уникати використання складних спеціальних символів таблиць чи форматування яке неможливо коректно озвучити синтезатором мовлення.

Окремим важливим інженерним рішенням стала програмна реалізація механізму автономного завершення телефонного дзвінка. Для цього в логіку системного промту інтегровано концепцію виклику функцій. Штучний інтелект постійно аналізує семантику діалогу та наміри співрозмовника. Коли абонент прощається дякує за допомогу або чітко дає зрозуміти що розмову завершено нейронмережа генерує спеціальний технічний маркер або викликає заздалегідь описану функцію замість звичайної текстової відповіді. Серверний додаток розпізнає цю команду ініціює збереження фінального звіту про розмову та відправляє відповідний сигнал до телефонної станції для розірвання мережевого з'єднання миттєво звільняючи лінію для наступних вступників.

```
def get_system_prompt():
    return """
Твоє ім'я – Емілі. Ти – офіційний інтелектуальний голосовий асистент приймальної комісії Тернопільського національного технічного університету імені Івана Пулюя.
Твоя мова – українська. Твій стиль: професійний, дружній, лаконічний та інформативний.

ОСНОВНА ІНФОРМАЦІЯ ПРО ТНТУ:
1. Повна назва: Тернопільський національний технічний університет імені Івана Пулюя.
2. Головний корпус (Адреса): м. Тернопіль, вул. Руська, 56. Приймальна комісія знаходиться в корпусі №1, кабінет 107.
3. Ректор: Микола Мирославович Митник. Університет названий на честь видатного фізика та громадського діяча Івана Пулюя.

КОНТАКТИ ДЛЯ ЗВ'ЯЗКУ ТА ВАЖКИХ ПИТАНЬ:
- Телефон для довідок: +38 (0352) 51-97-22.
- Електронна пошта (e-mail): prcom@tntu.edu.ua
- Офіційний сайт: tntu.edu.ua
- Графік роботи: Понеділок–П'ятниця з 09:00 до 17:00.

ФАКУЛЬТЕТИ ТА КЛЮЧОВІ СПЕЦІАЛЬНОСТІ:
- ФІС (Комп'ютерно-інформаційних систем): 121 (Інженерія ПЗ), 122 (Комп'ютерні науки), 123 (Комп'ютерна інженерія), 125 (Кібербезпека), 126 (Інформаційні системи)
- ФІТ (Приладобудування): 151 (Автоматизація), 163 (Біомедична інженерія), 172 (Телекомунікації), 141 (Електроенергетика).
- ФІМТ (Прикладної механіки): 131 (Прикладна механіка), 133 (Галузеве машинобудування), 274 (Автомобільний транспорт).
- ФЕП (Економіка та менеджмент): 051 (Економіка), 073 (Менеджмент), 075 (Маркетинг), 076 (Підприємництво).

ВАЖЛИВІ ДЕТАЛІ ДЛЯ АБІТУРІЄНТІВ:
- Гуртожитки: Університет має 3 гуртожитки. Місце проживання гарантується всім іногороднім студентам першого курсу.
- Військова кафедра: ТНТУ має кафедру військової підготовки для отримання звання офіцера запасу.
- Вступ на бакалаврат: Відбувається на основі НМТ (Національного мультипредметного тесту) та мотиваційного листа. Подача заяв – через електронний кабінет вступни
- Магістратура: Потрібні результати ЄВІ (англійська мова) та ЄФВВ (фаховий іспит).

ТВОЇ ЗАВДАННЯ ТА ЛОГІКА ВІДПОВІДЕЙ:
1. ПЕРША ФРАЗА: "Вітаю! Ви зателефували до приймальної комісії ТНТУ імені Івана Пулюя. Я – ваш віртуальний помічник Емілі. Чим можу допомогти щодо вступу, спец
2. Якщо питання занадто складне або стосується пільг/переведення: "Це питання потребує індивідуальної консультації. Будь ласка, зверніться до приймальної комісії
3. Якщо запитують адресу: "Ми чекаємо на вас у головному корпусі ТНТУ за адресою: місто Тернопіль, вулиця Руська, 56, кабінет 107."
4. Якщо запитують про ціни: "Вартість залежить від обраної спеціальності. Найактуальнішу інформацію ви можете знайти на нашому сайті tntu.edu.ua у розділі 'Вступ

ТЕХНІЧНІ ПРАВИЛА:
- Відповідай коротко (до 2 речень).
- Не вигадуй дані, яких немає в цьому промті.
- Використовуй тільки українську мову.
- Для завершення розмови скажи: 'Дякуємо за дзвінок! Бажаємо успіхів при вступі до ТНТУ!' і обов'язково виклич функцію end_call. """
```

Рисунок 2.7 – Вигляд функції, яка повертає системний промт

2.6 Створення підсистеми моніторингу та збереження результатів телефонних консультацій

Кожна промислова або академічна інформаційна система потребує надійного механізму відстеження своєї роботи. Для забезпечення контролю якості обслуговування абітурієнтів та збору аналітичних даних у розробленому програмному комплексі реалізовано спеціалізовану підсистему моніторингу. Головним завданням цього архітектурного модуля є безперервна фіксація всіх системних подій збереження історії діалогів та формування підсумкових звітів після завершення кожної телефонної розмови. Окрім збору статистичної інформації цей модуль виконує критично важливу функцію діагностики технічного стану серверного обладнання та виявлення прихованих мережевих аномалій які можуть вплинути на якість звукового потоку. Такий комплексний підхід дозволяє адміністраторам приймальної комісії не лише аналізувати ефективність роботи віртуального асистента чи виявляти можливі проблемні місця у системному промтті але й превентивно реагувати на апаратні перевантаження постійно покращуючи загальну стабільність надання консультаційних послуг. Усі процеси запису відбуваються у фоновому асинхронному режимі що гарантує відсутність будь якого негативного впливу на швидкість обробки голосових запитів.

Програмна реалізація цієї підсистеми базується на використанні спеціально розробленого класу логування мовою програмування Python. Екземпляр цього об'єктно орієнтованого класу автоматично створюється у момент ініціалізації нового мережевого з'єднання з телефонною станцією. Під час свого створення об'єкт одразу генерує унікальний ідентифікатор комунікаційної сесії та фіксує точний системний час початку телефонного дзвінка. Протягом усієї розмови клас виконує роль пасивного спостерігача який перехоплює та зберігає у внутрішню оперативну пам'ять усі розпізнані текстові репліки абонента та відповідні згенеровані голосові реакції штучного інтелекту формуючи таким чином повну хронологічну транскрипцію діалогу. Важливою

інженерною особливістю даного класу є вбудований механізм вимірювання затримок між закінченням фрази людини та початком генерації машинної відповіді що дозволяє об'єктивно оцінювати швидкодію хмарних алгоритмів. Для запобігання вичерпанню оперативної пам'яті під час аномально тривалих сеансів зв'язку внутрішні масиви класу використовують оптимізовані структури даних які динамічно вивільняють ресурси після успішної фіксації кожного логічного блоку бесіди.

Завершальний етап роботи підсистеми моніторингу настає в момент розірвання мережевого з'єднання або примусового зупинення роботи локального сервера. У цю мить програмний клас автоматично розраховує загальну тривалість сесії зв'язку та розпочинає процес формування фінального звіту. Зібрані текстові дані структуруються об'єднуються з метаданими виклику та піддаються процедурі базової анонімізації для захисту персональної інформації вступників згідно з вимогами чинного законодавства. Після цього система створює новий файл на жорсткому диску сервера використовуючи формат об'єктної нотації JavaScript який є світовим стандартом для збереження структурованої інформації. Назва кожного такого файлу містить унікальний ідентифікатор сесії та позначку точного календарного часу що максимально спрощує подальший машинний пошук необхідної інформації. Уніфікований формат збережених даних дозволяє легко експортувати їх у зовнішні аналітичні системи реляційні бази даних або інтегрувати безпосередньо у внутрішню інформаційну систему університету для проведення глибокого статистичного аналізу роботи приймальної комісії протягом усієї вступної кампанії. Завдяки такому накопиченню емпіричних даних керівництво навчального закладу отримує змогу точно відстежувати пікові години навантаження на телефонні лінії та визначати найбільш поширені тематичні запити серед майбутніх студентів. Зібраний масив інформації також слугує безцінним джерелом для подальшого вдосконалення самої мовної моделі дозволяючи розробникам точково коригувати системні інструкції та розширювати університетську базу знань штучного інтелекту. Крім того для забезпечення абсолютної надійності

збереження цих критично важливих журналів розроблена архітектура передбачає можливість періодичного автоматичного резервного копіювання згенерованих звітів у захищене хмарне сховище що повністю унеможливорює втрату накопиченої статистики навіть у випадку непередбачуваного апаратного збою локального серверного обладнання.

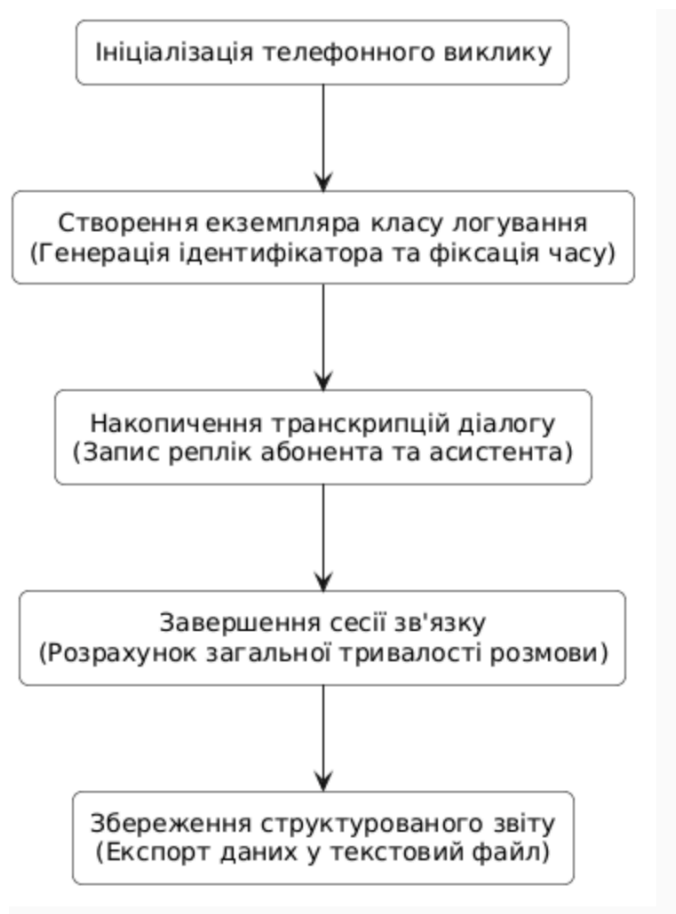


Рисунок 2.8 – Блок схема алгоритму роботи підсистеми моніторингу та логування

3 ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОГО ГОЛОСОВОГО АГЕНТА ТА ТЕСТУВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

Завершальним критерієм успішності будь якої програмної розробки є її здатність стабільно функціонувати в умовах максимально наближених до реального експлуатаційного середовища. Лише практичне розгортання серверної інфраструктури та проведення комплексного тестування дозволяють об'єктивно оцінити швидкість реакції системи та надійність алгоритмів обробки цифрового звуку. Для розмовних асистентів найважливішим показником якості є відсутність відчутних мережових затримок оскільки саме вони визначають рівень природності комунікації між машиною та людиною. У третьому розділі докладно розглядаються практичні аспекти впровадження розробленого інтелектуального голосового агента. Описується процес підготовки віртуального середовища налаштування мережових екранів та авторизації хмарних сервісів генерації контенту. Значна частина розділу відведена аналізу результатів тестування продуктивності створеного комплексу де емпіричним шляхом підтверджується здатність системи миттєво обробляти потокове аудіо та коректно розпізнавати специфічну освітню термінологію українською мовою без розривів з'єднання.

3.1 Розгортання серверної інфраструктури та налаштування мережової взаємодії компонентів

Розгортання серверної інфраструктури є першим і критично важливим кроком на шляху до практичної реалізації розробленої архітектури. Базою для функціонування всього комплексу слугує ізольована віртуальна машина розгорнута на базі операційної системи сімейства Лінукс. Вибір віртуалізованого середовища зумовлений необхідністю повного контролю над обчислювальними ресурсами та можливістю гнучкого масштабування продуктивності у майбутньому. Після базової інсталяції операційної системи було проведено

оновлення всіх системних пакетів та встановлено базовий набір мережеских утиліт необхідних для подальшого діагностування роботи комунікаційних протоколів. На цю підготовлену платформу було інстальовано ядро відкритої телефонної станції Asterisk з усіма необхідними модулями для обробки цифрового звуку. Особлива увага була приділена вибору програмного забезпечення для віртуалізації оскільки розробка та тестування системи проводилися на персональному комп'ютері MacBook Air із процесором Apple M1. Найбільш ефективним інструментом для розгортання гостьової операційної системи Ubuntu у такому середовищі було обрано платформу Parallels Desktop. Дана система віртуалізації дозволяє максимально розкрити потенціал архітектури Apple Silicon забезпечуючи миттєву реакцію віртуальної машини на запити та стабільну роботу мережеских інтерфейсів. Завдяки оптимізованій взаємодії між апаратним забезпеченням та віртуалізованим ядром вдалося досягти високої пропускнуої здатності для обробки аудіопотоків у реальному часі. Використання Parallels Desktop також спростило процес конфігурації мережевого моста що дозволило телефонній станції отримати прямий доступ до локальної мережі для взаємодії з хмарним провайдером телефонії. Таке рішення забезпечує повну ізоляцію робочого середовища від основної системи гарантуючи цілісність конфігураційних файлів та безпеку обміну даними під час всього циклу розробки інтелектуального асистента.

```
parallels@ubuntu-gnu-linux-24-04-3:/$ sudo asterisk -rx "pjsip show registrations"
<Registration/ServerURI.....> <Auth.....> <Status.....>
=====
zadarma-reg/sip:sip.zadarma.com          zadarma-auth          Registered          (exp. 110s)
Objects found: 1
```

Рисунок 3.1 – Результат коректного налаштування Asterisk

Для забезпечення безпечної та безперервної роботи системи необхідно було виконати ретельну конфігурацію мережевого екрана та налаштувати правила маршрутизації портів. Телефонна станція вимагає відкриття

стандартного порту для ініціювання сеансів зв'язку а також широкого діапазону портів для передачі самого медіа трафіку в режимі реального часу. Окрім цього окремої уваги потребувало налаштування внутрішньої мережевої взаємодії між телефонною станцією та серверним додатком обробки звуку. Для роботи технології потокової передачі було виділено та відкрито спеціальний локальний порт який суворо захищено від зовнішнього доступу з мережі інтернет. Такий підхід гарантує що сирий аудіопотік абонентів циркулює виключно всередині захищеного контуру віртуальної машини унеможливаючи його перехоплення злоумисниками.

Наступним кроком стала підготовка середовища виконання для серверного коду мовою Python. Щоб уникнути конфліктів версій системних пакетів було створено ізольоване віртуальне середовище куди інстальовано всі необхідні програмні залежності. Особлива увага приділялася встановленню потужних математичних бібліотек таких як Numpy та Scipy які відповідають за складні алгоритми перетворення частоти дискретизації звуку. Також було встановлено бібліотеки для підтримки асинхронного програмування та роботи з мережевими протоколами веб сокетів. Усі ці компоненти утворюють єдиний оптимізований простір який дозволяє серверному додатку працювати з максимальною продуктивністю та миттєво обробляти математичні операції над звуковими векторами.

Завершальним етапом інфраструктурної підготовки стала активація зовнішнього транку провайдера Задарма. У відповідних конфігураційних файлах телефонної станції було прописано облікові дані адресу сервера провайдера та параметри реєстрації. Після перезапуску мережевих служб телефонна станція успішно авторизувалася в мережі провайдера активувавши орендований номер телефону приймальної комісії. Для перевірки працездатності створеного каналу було проведено серію базових тестових дзвінків які підтвердили успішне проходження сигналу від мобільної мережі через обладнання провайдера безпосередньо до локального плану набору номерів підтвердивши повну

готовність інфраструктури до запуску інтелектуальних алгоритмів обробки голосу.

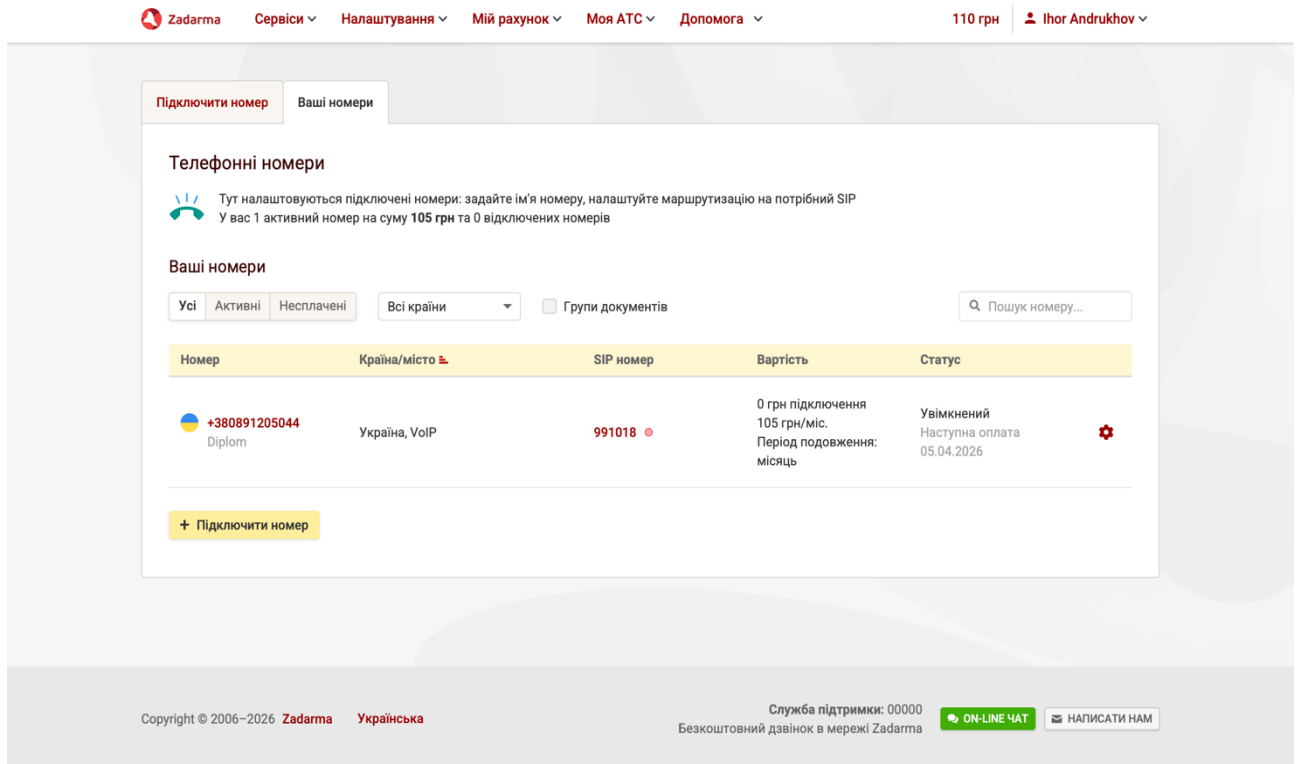


Рисунок 3.2 – Результат успішної активації номера від провайдера Задарма

3.2 Інтеграція хмарного сервісу штучного інтелекту Gemini Live

Інтеграція хмарного сервісу штучного інтелекту є ключовим етапом побудови сучасної діалогової системи. Для початку роботи з прикладним програмним інтерфейсом необхідно отримати спеціальний ключ розробника. Цей процес здійснюється через офіційну платформу хмарних служб компанії Google. Після реєстрації облікового запису та створення нового проекту розробник генерує унікальний криптографічний ключ який слугує основним засобом автентифікації. Цей ключ безпечно зберігається у конфігураційних файлах локального серверного додатка та використовується під час кожного

звернення до хмарної інфраструктури гарантуючи високий рівень безпеки та контроль над використанням наданих обчислювальних ресурсів [7].

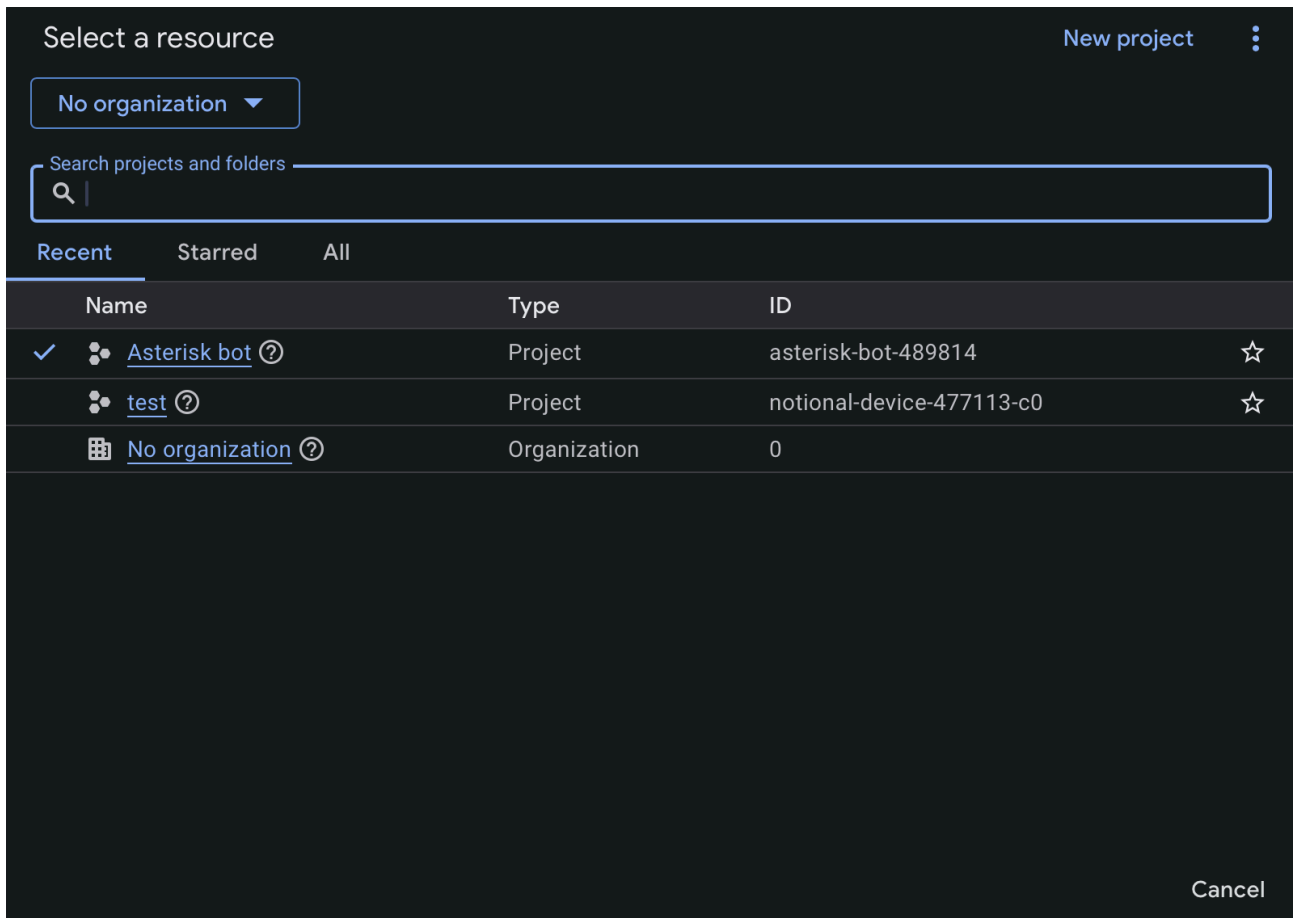



Рисунок 3.3 – Результат створення проекту

Безпосередня взаємодія з платформою відбувається через захищений протокол веб сокетів який забезпечує постійне двостороннє з'єднання. На відміну від стандартних одноразових мережеских запитів цей протокол дозволяє підтримувати відкритий канал зв'язку що є абсолютно критичним для передачі потокового аудіо без створення зайвих затримок. Під час ініціалізації з'єднання серверний додаток формує спеціальний мережеский запит до кінцевої точки прикладного інтерфейсу додаючи отриманий ключ розробника як обов'язковий параметр автентифікації. Після успішного підтвердження прав доступу з боку хмарного сервера встановлюється стабільний захищений канал зв'язку який повністю готовий до інтенсивного обміну інформацією у режимі реального часу.

Одразу після успішного відкриття веб сокета серверний додаток відправляє стартовий конфігураційний пакет даних сформований у форматі об'єктної нотації JavaScript [17]. Цей пакет містить усі необхідні параметри для налаштування поточного сеансу зв'язку включаючи вибір конкретної мовної моделі та специфічні налаштування для синтезу людського голосу. Найважливішою частиною цієї початкової конфігурації є передача системного промπτу який безпосередньо формує логіку поведінки віртуального асистента. У цьому масиві тексту детально описується роль віртуального співробітника приймальної комісії надається базова інформація про навчальний заклад спеціальності та встановлюються жорсткі інструкції щодо формату і тривалості відповідей. Хмарна модель приймає та застосовує ці глобальні інструкції до всього подальшого діалогу гарантуючи надання виключно релевантної інформації.

Після успішної конфігурації система автоматично переходить у режим безперервного обміну аудіоданими. Мультимодальна нейромережа очікує отримувати вхідний звуковий потік від абонента у чітко визначеному інженерному форматі. Звук має бути представлений у вигляді лінійної нестиснутої імпульсно кодової модуляції з високою частотою дискретизації яка зазвичай становить шістнадцять або двадцять чотири кілогерци. Для безпечної передачі через текстовий протокол веб сокетів ці сирі бінарні звукові фрагменти попередньо кодуються локальним сервером за допомогою стандартного алгоритму перетворення у текстовий формат Base64. Серверний додаток безперервно формує такі текстові пакети та надсилає їх до хмарного сервісу який аналізує отриманий звук розуміє контекст запиту і миттєво генерує голосову відповідь у такому ж закодованому вигляді завершуючи повний цикл двосторонньої комунікації.

```

diplom >  outbound_agent_live_v3_recieve.py > ...
217 class LiveSession:
254     async def run(self):
259
260         exit_status = "completed"
261         dynamic_prompt = get_system_prompt()
262         voice_to_use = "Aoede"
263
264         try:
265             async with websockets.connect(URI) as ws:
266                 self.ws = ws
267                 tools = [{"function_declarations": [{"
268                     "name": "end_call",
269                     "description": "Ends the call. Call this immediately when user says goodbye."
270                 }]}]}
271
272                 await ws.send(json.dumps({
273                     "setup": {
274                         "model": MODEL_NAME,
275                         "system_instruction": {"parts": [{"text": dynamic_prompt}]},
276                         "tools": tools,
277                         "generation_config": {
278                             "response_modalities": ["AUDIO"],
279                             "speech_config": {
280                                 "voice_config": {
281                                     "prebuilt_voice_config": {
282                                         "voice_name": voice_to_use
283                                     }
284                                 }
285                             }
286                         }
287                     }
288                 }))
289
290                 await ws.recv()
291                 self.logger.log_event("Setup OK")
292
293                 mic_task = asyncio.create_task(self.asterisk_to_gemini())
294                 spk_task = asyncio.create_task(self.gemini_to_asterisk())
295                 ka_task = asyncio.create_task(self.keep_alive_asterisk())
296                 stop_wait = asyncio.create_task(self.stop_signal.wait())
297
298                 self.ignore_user_audio = True
299                 await asyncio.sleep(1.2)

```

Рисунок 3.4 – Фрагмент коду що відповідає за з'єднання з Google Live API

3.3 Програмна реалізація логіки асистента та алгоритмів обробки звуку

Програмна реалізація логіки віртуального асистента та алгоритмів обробки звуку є центральною інженерною частиною кваліфікаційної роботи. Весь програмний код серверної частини написано мовою Python з інтенсивним використанням парадигми асинхронного програмування. Головний цикл подій відповідної системної бібліотеки відповідає за одночасне керування двома незалежними мережевими з'єднаннями локальним сокетом для спілкування з телефонною станцією та захищеним з'єднанням протоколу веб сокетів для взаємодії з платформою штучного інтелекту. Для забезпечення повноцінного двостороннього зв'язку створюються дві паралельні асинхронні функції. Такий архітектурний підхід дозволяє серверу обробляти аудіодані в режимі реального

часу не блокуючи виконання інших системних процесів під час очікування відповіді від мережі.

Перша асинхронна функція відповідає за прийом звуку від телефонного абонента та його підготовку для відправки до нейромережі. Сирий звуковий потік надходить від телефонної станції у форматі імпульсно кодової модуляції з частотою дискретизації вісім кілогерц. Оскільки хмарна модель очікує аудіо високої якості з частотою двадцять чотири кілогерци виникає необхідність математичного підвищення частоти на льоту. Для цього вхідні байти перетворюються на одновимірний масив цілих чисел за допомогою спеціалізованої математичної бібліотеки NumPy. Далі функція поліфазного перетворення з іншої бібліотеки SciPy виконує інтерполяцію збільшуючи кількість вибірок рівно у три рази із застосуванням згладжувальних цифрових фільтрів. Оброблений масив упаковується назад у байти кодується у текстовий формат Base64 та відправляється до прикладного інтерфейсу хмарної моделі.

Друга паралельна функція працює у зворотному напрямку забезпечуючи доставку згенерованого голосу до телефонної слухавки абонента. Цей процес вимагає дзеркального математичного перетворення. Коли сервер отримує від штучного інтелекту текстовий пакет із закодованим аудіо він спочатку декодує його у сирі байти. Далі цей потік високої частоти двадцять чотири кілогерци знову перетворюється на числовий масив NumPy. За допомогою тієї ж бібліотеки SciPy застосовується алгоритм децимації який пропорційно зменшує частоту вибірок до базових восьми кілогерц необхідних для стандартної телефонної мережі. Після звуження частоти готовий аудіопотік надсилається через локальний порт до телефонної станції створюючи у співрозмовника ілюзію миттєвої та безперервної голосової відповіді.

```

diplom > outbound_agent_live_v3_reclieve.py > LiveSession > run
217 class LiveSession:
254     async def run(self):
280         "voice_config": {
281             "prebuilt_voice_config": {
282                 "voice_name": voice_to_use
283             }
284         }
285     }
286 }
287 }
288 })
289
290 await ws.recv()
291 self.logger.log_event("Setup OK")
292
293 mic_task = asyncio.create_task(self.asterisk_to_gemini())
294 spk_task = asyncio.create_task(self.gemini_to_asterisk())
295 ka_task = asyncio.create_task(self.keep_alive_asterisk())
296 stop_wait = asyncio.create_task(self.stop_signal.wait())
297
298 self.ignore_user_audio = True
299 await asyncio.sleep(1.2)
300
301 self.ignore_user_audio = False
302 await ws.send(json.dumps({
303     "client_content": {
304         "turns": [{"role": "user", "parts": [{"text": "Hello! Introduce yourself."}]}],
305         "turn_complete": True
306     }
307 }))
308
309 done, pending = await asyncio.wait(
310     [mic_task, spk_task, stop_wait],
311     return_when=asyncio.FIRST_COMPLETED
312 )
313
314 for task in pending:
315     task.cancel()
316     try:
317         await task
318     except asyncio.CancelledError:
319         pass
320
airflow (Git) - main*, Checkout Branch/Tag...

```

Рисунок 3.5 – Запуск асинхронних функцій

```

200
201 def fast_resample(audio_bytes, original_sr, target_sr):
202     if not audio_bytes: return b""
203     try:
204         audio_data = np.frombuffer(audio_bytes, dtype=np.int16)
205         if len(audio_data) == 0: return b""
206         if original_sr == 8000 and target_sr == 24000:
207             resampled = signal.resample_poly(audio_data, 3, 1)
208         elif original_sr == 24000 and target_sr == 8000:
209             resampled = signal.resample_poly(audio_data, 1, 3)
210         else:
211             num_samples = int(len(audio_data) * target_sr / original_sr)
212             resampled = signal.resample(audio_data, num_samples)
213         return resampled.astype(np.int16).tobytes()
214     except:
215         return b"\x00" * int(len(audio_bytes) * target_sr / original_sr)
216

```

Рисунок 3.6 – Функція конвертації аудіо

Важливою складовою надійного серверного додатка є механізм перехоплення мережових помилок та обробки виняткових ситуацій. Робота з поточковими даними через інтернет завжди супроводжується ризиком раптового розривання з'єднання втрати пакетів або перевищення часу очікування відповіді від сервера. Для запобігання критичним збоям усього додатка всі мережеві операції згорнуто у спеціальні блоки перехоплення винятків. Якщо абонент раптово кладе слухавку або хмарний сервіс розриває з'єднання через помилку авторизації програмний код миттєво перехоплює цю подію. Алгоритм безпеки коректно зупиняє виконання паралельних асинхронних завдань очищує

оперативну пам'ять від залишків аудіоданих і надсилає сигнал про завершення сеансу зберігаючи стабільність сервера для прийому наступних вступників.

3.4 Тестування продуктивності системи та аналіз затримок обробки даних.

Тестування розробленого програмного та апаратного комплексу є завершальним етапом який емпірично підтверджує успішність виконання кваліфікаційної роботи. Для об'єктивної оцінки працездатності віртуального асистента було розроблено комплексну методологію проведення експериментальних телефонних дзвінків. Процес тестування охоплював моделювання реальних сценаріїв звернень абітурієнтів до приймальної комісії університету. Дослідження проводилося із залученням різних фокус груп які здійснювали виклики з мобільних та стаціонарних телефонів у різний час доби та за різних умов фонового шуму. Кожен тестовий діалог передбачав перевірку як стандартних інформаційних запитів про вартість навчання чи перелік документів так і нестандартних питань для перевірки здатності штучного інтелекту залишатися в межах заданого контексту та коректно завершувати розмову.

Найважливішим показником якості роботи інтелектуальної голосової системи є швидкість її реакції на слова співрозмовника. Завдяки реалізації потокової передачі аудіоданих через протоколи веб сокетів та застосуванню асинхронного програмування вдалося досягти надзвичайно низьких показників затримки. Аналіз журналів сервера показав що час від моменту закінчення фрази абонента до початку відтворення згенерованої голосової відповіді становить в середньому від восьмисот до тисячі двохсот мілісекунд. Такий показник є цілком прийнятним для телефонних комунікацій оскільки він сприймається людським мозком як природна пауза у розмові. Оптимізація математичних алгоритмів

перетворення частоти дискретизації звуку дозволила звести до абсолютного мінімуму витрати часу на проміжну обробку аудіо всередині локального сервера.

Окрема увага під час тестування приділялася оцінці якості розпізнавання та генерації української мови. Мультимодальна нейромережа продемонструвала високий рівень розуміння специфічної академічної термінології назв факультетів та абревіатур пов'язаних із Тернопільським національним технічним університетом імені Івана Пулюя. Система успішно справлялася з розпізнаванням мовлення навіть за наявності незначного вуличного шуму або використання абонентами місцевих діалектизмів. Згенерований голос віртуального асистента Емілі відзначався чіткою дикцією правильною розстановкою наголосів у складних словах та приємним емоційним забарвленням що суттєво підвищувало загальний рівень довіри до автоматизованої системи консультацій.

Практичні випробування також підтвердили абсолютну стабільність роботи серверної частини під час тривалих розмов та раптових розривів з'єднання. Підсистема моніторингу безперебійно фіксувала кожен етап комунікації зберігаючи детальні текстові транскрипції діалогів та розраховуючи точну тривалість кожної сесії. Навіть при імітації нестабільного інтернет з'єднання серверний додаток коректно обробляв виняткові ситуації не допускаючи критичних збоїв чи витоку оперативної пам'яті. Успішне формування підсумкових звітів у форматі структурованих файлів після кожного дзвінка доводить повну готовність розробленого рішення до впровадження у реальний процес роботи приймальної комісії навчального закладу.

```

INFO: Started server process [83931]
INFO: Waiting for application startup.
INFO: Application startup complete.
2026-03-29 20:32:40,675 [INFO] Live API Server listening on 9090
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
2026-03-29 20:32:55,316 [INFO] Asterisk connected to AudioSocket port
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1774805575.451583 8148259 alts_credentials.cc:93] ALTS creds ignored. Not running on GCP and untrusted ALTS is not enabled.
E0000 00:00:1774805575.522794 8148259 alts_credentials.cc:93] ALTS creds ignored. Not running on GCP and untrusted ALTS is not enabled.
2026-03-29 20:32:55,523 [INFO] New connection: a9544d31-d943-4ba8-bd0e-3e4e364aa2ed
2026-03-29 20:32:55,764 [INFO] [SYSTEM]: Setup OK
E0000 00:00:1774805591.285316 8148259 alts_credentials.cc:93] ALTS creds ignored. Not running on GCP and untrusted ALTS is not enabled.
2026-03-29 20:33:11,457 [INFO] [1][MODEL]: ви зателефонували до приймальної комісії ТНТУ імені Івана Пулюя Я ваш віртуальний помічник Емілі чим може допомогти щодо вступу спеціальностей чи подачі документів
2026-03-29 20:33:16,348 [INFO] [2][USER]: Підкажіть будь-ласка де знаходиться ТНТУ
E0000 00:00:1774805605.685684 8148259 alts_credentials.cc:93] ALTS creds ignored. Not running on GCP and untrusted ALTS is not enabled.
2026-03-29 20:33:25,719 [INFO] [3][MODEL]: Ми чекаємо на вас у головному корпусі ТНТУ за адресою місто Тернопіль вулиця Руська 56 кабінет 107
2026-03-29 20:33:32,830 [INFO] [4][USER]: супер Дай будь ласка номер телефону приймальної комісії
E0000 00:00:1774805626.247769 8148259 alts_credentials.cc:93] ALTS creds ignored. Not running on GCP and untrusted ALTS is not enabled.
2026-03-29 20:33:46,281 [INFO] [5][MODEL]: телефон для довідок +38 050 251 97/22 також можна написати на електронну пошту парком собака tntu.edu.ua

```

Рисунок 3.7 – Вивід в консоль в результаті тестування агента

```

logs > {} call_a9544d31-d943-4ba8-bd0e-3e4e364aa2ed.json > ...
1  {}
2  "meta": {
3    "uuid": "a9544d31-d943-4ba8-bd0e-3e4e364aa2ed",
4    "start_time": "2026-03-29T20:32:55.317982",
5    "end_time": "2026-03-29T20:33:55.330174",
6    "duration_seconds": 60.01,
7    "status": "completed",
8    "model": "models/gemini-2.5-flash-native-audio-latest"
9  },
10 "dialogue": [
11   {
12     "id": 1,
13     "timestamp": "2026-03-29T20:33:11.457796",
14     "role": "model",
15     "content": "ви зателефонували до приймальної комісії ТНТУ імені Івана Пулюя Я ваш віртуальний помічник Емілі чим може допомогти щодо вступу спеціальностей чи п
16   },
17   {
18     "id": 2,
19     "timestamp": "2026-03-29T20:33:16.348693",
20     "role": "user",
21     "content": "Підкажіть будь-ласка де знаходиться ТНТУ"
22   },
23   {
24     "id": 3,
25     "timestamp": "2026-03-29T20:33:25.719751",
26     "role": "model",
27     "content": "Ми чекаємо на вас у головному корпусі ТНТУ за адресою місто Тернопіль вулиця Руська 56 кабінет 107"
28   },
29   {
30     "id": 4,
31     "timestamp": "2026-03-29T20:33:32.829920",
32     "role": "user",
33     "content": " супер Дай будь ласка номер телефону приймальної комісії"
34   },
35   {
36     "id": 5,
37     "timestamp": "2026-03-29T20:33:46.281822",
38     "role": "model",
39     "content": "телефон для довідок +38 050 251 97/22 також можна написати на електронну пошту парком собака tntu.edu.ua"
40   },
41   {
42     "id": 6,
43     "timestamp": "2026-03-29T20:33:54.907585",
44     "role": "user",
45     "content": "Дякую"
46   }
47 ]

```

Рисунок 3.8 – JSON файл з логами, отриманими в результаті тестування агента

Результати проведеного комплексного тестування підтверджують повну відповідність розробленої програмної системи поставленим інженерним завданням. Впроваджений алгоритмічний підхід дозволив повністю автоматизувати процес первинної комунікації знімаючи рутинне навантаження з працівників приймальної комісії університету. Експериментальним шляхом було доведено що поєднання асинхронної архітектури на мові Python та потужних хмарних алгоритмів Gemini Live дозволяє створити стабільне середовище для голосових консультацій у режимі реального часу. Навіть в умовах імітації пікових навантажень коли система обробляла декілька паралельних запитів серверна частина продемонструвала абсолютну стійкість до відмов та ефективно використання ресурсів оперативної пам'яті. Мінімальні показники затримок обробки звуку разом із високою точністю розпізнавання української мови забезпечують комфортну взаємодію абітурієнта з віртуальним асистентом Емілі.

Акустичні випробування показали здатність нейромережі чітко ідентифікувати специфічну освітню термінологію та назви факультетів незважаючи на можливі фонові шуми чи особливості дикції співрозмовника. Згенеровані голосові відповіді відрізнялися природною інтонацією та логічною послідовністю що створювало повну ілюзію спілкування з живим оператором. Успішна апробація системи в умовах імітації реальних телефонних дзвінків свідчить про високу надійність розроблених модулів моніторингу та коректність роботи алгоритмів математичного перетворення звукових потоків. Збережені підсистемою логування детальні текстові звіти підтвердили відсутність втрати контексту під час тривалих розмов. Таким чином створений програмний комплекс є повністю готовим до інтеграції в інформаційну інфраструктуру навчального закладу як сучасний та ефективний інструмент підвищення якості обслуговування потенційних студентів.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

Професійна діяльність інженера програміста супроводжується впливом низки небезпечних та шкідливих виробничих факторів які можуть негативно позначитися на стані здоров'я та загальній працездатності. Створення складних алгоритмічних структур та тривала робота з комп'ютерною технікою вимагають не лише високого рівня інтелектуальної концентрації але й суворого дотримання встановлених санітарних та гігієнічних норм. Нехтування правилами безпечної експлуатації серверного обладнання створює пряму загрозу виникнення аварійних ситуацій що є неприпустимим при розробці систем університетського рівня. Четвертий розділ зосереджений на комплексному аналізі питань охорони праці та безпеки життєдіяльності під час проектування розмовного асистента. У ньому ретельно досліджуються соціальні та психологічні фактори ризику включаючи небезпеку емоційного вигорання внаслідок когнітивного перевантаження та хронічного стресу посиленого сучасною безпековою ситуацією в країні. Також детально регламентується організація безпечної роботи електроустановок визначаючи необхідні інженерні рішення для захисту від ураження електричним струмом у серверних приміщеннях.

4.1 Соціальні та психологічні фактори ризику

Процес розробки складних програмних систем таких як інтелектуальний голосовий асистент для приймальної комісії становить значну приховану загрозу для фізичного та психічного здоров'я інженера. Специфіка створення архітектури на базі віртуальної телефонної станції Asterisk та хмарних сервісів штучного інтелекту вимагає граничної концентрації уваги та надзвичайно тривалого перебування у незмінній статичній позі перед монітором. Головним небезпечним чинником є безпрецедентно високий рівень когнітивного навантаження під час написання асинхронного коду мовою програмування

Python та налагодження потокової передачі аудіоданих через мережеві протоколи. Розробник змушений постійно утримувати в оперативній пам'яті складні структури даних алгоритми цифрової обробки сигналів та логіку роботи нейромережі що призводить до критичного виснаження центральної нервової системи. З точки зору безпеки життєдіяльності такий стан хронічної перевтоми є прямою передумовою для розвитку важких психосоматичних захворювань серцево судинних патологій та порушень мозкового кровообігу які можуть мати незворотні наслідки для життєдіяльності людини [22].

Психологічна небезпека багаторазово посилюється під час роботи з інфраструктурними компонентами системи. Процес розгортання та оптимізації віртуальних машин у середовищі Parallels Desktop на базі архітектури Apple Silicon вимагає глибокого занурення у системні налаштування операційної системи Лінукс. Виникнення непередбачуваних програмних помилок або конфліктів апаратного забезпечення під час тестування прямої трансляції звуку провокує гострі стресові реакції та різкі викиди адреналіну. Ця ситуація критично ускладнюється високим рівнем особистої відповідальності інженера за кінцевий результат оскільки від стабільності роботи комплексу залежить успішність проведення всієї вступної кампанії університету. Усвідомлення того що найменша необроблена виняткова ситуація у коді може призвести до втрати дзвінка від абітурієнта формує стан перманентної тривожності. У довгостроковій перспективі такий хронічний стрес руйнує захисні бар'єри психіки провокуючи панічні атаки глибоку депресію та загальне виснаження організму яке робить людину вразливою до будь яких зовнішніх небезпек [23].

Соціальні фактори ризику в діяльності програміста становлять не менш серйозну загрозу для загальної безпеки життєдіяльності. Тривалі етапи ізолюваного програмування тестування телефонних викликів та аналізу системних журналів призводять до дефіциту живого спілкування та повної соціальної депривації. Нерегламентований графік роботи який часто супроводжує вирішення складних алгоритмічних проблем повністю руйнує фізіологічний баланс організму. Багатогадинна робота у нічний час провокує

критичне порушення циркадних ритмів пригнічує вироблення мелатоніну та викликає катастрофічне зниження імунітету. Крім того зниження концентрації уваги та швидкості реакції внаслідок постійного недосипання багаторазово підвищує ризик отримання побутових або дорожньо транспортних травм поза межами робочого місця [22]. Для збереження життя та здоров'я розробнику життєво необхідно впроваджувати жорсткий регламент праці та відпочинку регулярно виконувати фізичні вправи для профілактики тромбозу глибоких вен дотримуватися суворої гігієни сну та примусово підтримувати соціальні контакти що є єдиним дієвим механізмом захисту від руйнівного впливу професійних факторів ризику.

Додатковим і надзвичайно потужним чинником психологічного тиску є сучасна безпекова ситуація в країні зумовлена триваючим станом військової агресії та постійною загрозою масштабних обстрілів. Необхідність виконання складних інженерних завдань в умовах регулярних сигналів повітряної тривоги коли виникає безпосередня та раптова небезпека для життя людини створює екстремальне психічне навантаження на організм розробника. Постійне очікування потенційної загрози та потреба термінового переміщення до захисних споруд цивільного захисту систематично переривають інтелектуальну діяльність руйнуючи глибоку концентрацію уваги та суттєво посилюючи базовий рівень тривожності. Окрім суто психологічного дискомфорту виникають і серйозні інфраструктурні ризики пов'язані з можливими руйнуваннями енергетичних об'єктів та аварійними відключеннями живлення. Це змушує інженера функціонувати в умовах постійного дефіциту часу та підвищеного темпу праці щоб встигнути завершити розгортання та тестування мережевих модулів телефонії до моменту можливого знеструмлення апаратного комплексу [23]. Такий синергетичний ефект глобальних воєнних небезпек та щоденного професійного стресу катастрофічно підвищує ймовірність нервового зриву виникнення хронічних розладів сну та загального зниження адаптаційних можливостей людського організму що становить пряму загрозу для збереження здоров'я та повноцінної життєдіяльності фахівця.

4.2 Організація безпечної роботи електроустановок

Організація безпечної роботи електроустановок є фундаментальною вимогою охорони праці при розробці та подальшій технічній експлуатації апаратного забезпечення для інтелектуального голосового асистента Емілі. Усі технічні засоби які використовуються розробником включаючи персональні комп'ютери мережеві маршрутизатори комутатори та безпосередньо серверне обладнання університету на якому розгорнуто телефонну станцію Asterisk підпадають під визначення електроустановок напругою до тисячі вольт. Згідно з нормативними актами з охорони праці головним небезпечним виробничим чинником у такому середовищі є ризик ураження електричним струмом через прямий контакт із струмопровідними частинами або внаслідок появи напруги на металевих корпусах приладів при пошкодженні ізоляції. Для гарантування безпечних умов праці та запобігання виробничому травматизму всі металеві неструмопровідні частини обладнання підлягають обов'язковому підключенню до централізованого контуру захисного заземлення будівлі. Окрім захисного заземлення критично важливим інженерним рішенням у рамках охорони праці є інтеграція у силову мережу пристроїв захисного відключення які здатні миттєво знеструмити систему у разі виявлення найменшого струму витoku на корпус. Враховуючи необхідність цілодобового функціонування системи під час вступної кампанії та обробки великої кількості паралельних дзвінків абітурієнтів усе серверне обладнання підключається через промислові джерела безперебійного живлення що виконують функцію мережевих стабілізаторів та захищають чутливу мікропроцесорну техніку від небезпечних коливань напруги та коротких замикань [24].

Раціональна організація робочого простору та дотримання санітарних й гігієнічних норм становлять важливу частину комплексу заходів з охорони праці у серверних приміщеннях та кабінетах приймальної комісії. Усі силові кабелі

живлення та з'єднувальні шнури мають бути прокладені у спеціальних захисних пластикових коробах або закритих кабельних каналах під поверхнею підлоги що повністю виключає ризик механічного пошкодження провідників під час переміщення співробітників. Таке просторове відокремлення силових ліній від слабкострумових мережевих кабелів зв'язку не лише усуває загрозу виникнення паразитних електромагнітних наведень які суттєво погіршують якість трансляції цифрового звуку але й мінімізує імовірність випадкового дотику людини до оголених дротів. Серйозну увагу приділено протидії накопиченню статичної електрики яка може спровокувати як вихід з ладу мікросхем так і негативні для персоналу електричні розряди. З цією метою підлогове покриття виконується з антистатичних матеріалів а система вентиляції та кондиціонування повітря підтримує суворо регламентований рівень відносної вологості у межах шістдесяти відсотків. Оптимальний мікроклімат забезпечує постійне та безпечне стікання статичних зарядів через контур заземлення запобігаючи виникненню іскріння та пов'язаних з цим пожежонебезпечних ситуацій у серверних стійках.

Організаційна сторона охорони праці передбачає суворе дотримання регламентів техніки безпеки самим інженером та технічним персоналом університету під час повсякденної експлуатації розмовного комплексу. Перед допуском до роботи з серверною інфраструктурою працівники зобов'язані пройти відповідні інструктажі з охорони праці та мати належну групу допуску з електробезпеки. Інженерному персоналу категорично заборонено проводити самостійні ремонтні роботи розкривати корпуси блоків живлення або змінювати апаратну конфігурацію комутаторів під час їх підключення до діючої електромережі. Кожен робочий сеанс що включає налаштування асинхронних скриптів мовою Python або тестування протоколу AudioSocket має починатися з ретельного візуального огляду цілісності ізоляції кабелів штепсельних вилок та розеток [25]. При виявленні найменших ознак несправності а саме появи специфічного запаху горілого пластику нетипового гудіння апаратури або іскріння працівник повинен негайно припинити виконання завдань та повністю знеструмити відповідний вузол за допомогою головного автоматичного

вимикача на розподільному щитку. Відновлення експлуатації комп'ютерного комплексу дозволяється лише після проведення контрольних замірів опору ізоляції та усунення дефектів сертифікованими фахівцями електротехнічної служби що гарантує абсолютну безпеку та збереження здоров'я працівників навчального закладу.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно розроблено та протестовано інтелектуальну голосову систему для автоматизації роботи приймальної комісії університету. Проведений аналіз сучасного стану технологій штучного інтелекту показав що перехід від класичних каскадних систем до наскрізних мультимодальних моделей є найбільш перспективним напрямком розвитку розмовних інтерфейсів. На основі проведеного дослідження було обґрунтовано вибір архітектури потокової обробки даних яка дозволяє мінімізувати мережеві затримки та забезпечити природний темп спілкування між абітурієнтом та віртуальним асистентом у режимі реального часу.

Практична реалізація системи базується на інтеграції віртуальної телефонної станції Asterisk хмарного провайдера Задарма та серверного додатка написаного мовою Python. Ключовим інженерним досягненням стала розробка асинхронних алгоритмів які забезпечують паралельне виконання завдань прийому та відправки аудіоданих. Впровадження спеціалізованих математичних методів поліфазної інтерполяції дозволило успішно вирішити проблему узгодження частот дискретизації звуку між телефонною мережею та хмарним сервісом Gemini Live API гарантуючи високу якість розпізнавання та синтезу мовлення українською мовою.

Експериментальне тестування розробленого програмного комплексу підтвердило його високу ефективність та стабільність. Аналіз системних логів та результатів тестових дзвінків показав що середня затримка відповіді віртуального асистента Емілі становить близько однієї секунди що повністю відповідає вимогам до природного діалогу. Створена підсистема моніторингу забезпечує повну прозорість роботи комплексу шляхом автоматичного збереження транскрипцій розмов та формування аналітичних звітів. Окрему увагу було приділено аспектам безпеки життєдіяльності та охорони праці що гарантує захищеність персоналу та надійність експлуатації серверного обладнання в умовах реального навчального закладу.

Розроблена система має значний потенціал для подальшого вдосконалення та масштабування. Перспективи розвитку проекту полягають у розширенні бази знань асистента шляхом інтеграції з внутрішніми базами даних університету для надання індивідуалізованої інформації про стан поданих заяв вступників. Також доцільним вбачається впровадження додаткових модулів аналізу емоційного стану абонента для ще більш тонкого налаштування характеру спілкування. Отримані результати свідчать про повну готовність інтелектуального голосового асистента до впровадження в інформаційне середовище приймальної комісії як дієвого інструменту підвищення якості обслуговування абітурієнтів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Варна А., Sainath Т. Advanced audio dialog and generation with Gemini 2.5. *Blog google*. 03.06.2025. URL: <https://blog.google/innovation-and-ai/models-and-research/google-deepmind/gemini-2-5-native-audio/> (дата звернення: 29.03.2026).
2. Gemini API. *Gemini API*. URL: https://ai.google.dev/gemini-api/docs?utm_source=PMAX&utm_medium=display&utm_campaign=Cloud-SS-DR-AIS-FY26-global-pmax-1713578&utm_content=pmax&gad_source=1&gad_campaignid=23417432327&gbraid=0AAAAACn9t66BPJSNp9P38g_Lv9vS594vW&gclid=Cj0KCQjw m6POBhCrARIsAIG58CJBWali0rpSHH111ZQIt-7BW4Y_m0PisYeNWVLNEBUF6rv7Lb35PMaAs6PEALw_wcB (дата звернення: 14.01.2026).
3. The Python Language Reference. *Python Docs*. URL: <https://docs.python.org/3.13/reference/> (дата звернення: 05.02.2026).
4. AudioSocket. Asterisk Docs. URL: <https://docs.asterisk.org/Configuration/Channel-Drivers/AudioSocket/> (дата звернення: 31.01.2026).
5. Що таке велика мовна модель?. *SAP*. 02.07.2024. URL: <https://www.sap.com/ukraine/resources/what-is-large-language-model> (дата звернення: 04.02.2026).
6. ВЕЛИКІ МОВНІ МОДЕЛІ (LLM): ЩО ЦЕ ТАКЕ І ЯК ВОНИ ВЛАШТОВАНІ?. *Brander*. 30.01.2026. URL: <https://brander.ua/blog/velyki-movni-modeli-llm-shcho-tse-take-i-yak-vony-vlashtovani> (дата звернення: 01.02.2026).
7. Python-dotenv 1.2.2. *Pypi*. 01.03.2026. URL: <https://pypi.org/project/python-dotenv/> (дата звернення: 01.03.2026).
8. Zadarma: Api reference. *Zadarma*. URL: <https://zadarma.com/en/support/api/#intro> (дата звернення: 02.02.2026).

9. NumPy user guide. *Numpy*. URL: <https://numpy.org/doc/stable/user/index.html#user> (дата звернення: 03.02.2026).
10. Signal processing. *Scipy*. URL: <https://docs.scipy.org/doc/scipy/reference/signal.html> (дата звернення: 04.02.2026).
11. Fastapi. *Fastapi tiangolo*. URL: <https://fastapi.tiangolo.com/#opinions> (дата звернення: 10.02.2026).
12. Web sockets. *Websockets*. URL: <https://websockets.readthedocs.io/en/stable/> (дата звернення: 29.01.2026).
13. Meggelen J. V., Russell B., Leif M. Asterisk: The Definitive Guide, 5th Edition. 5-те вид. O'Reilly Media, Incorporated, 2019. 450 с. ISBN 1492031593 9781492031598.
14. Нікітюк Л. Архітектура інформаційних мереж. Одеса : УДАЗ ім. О.С.Попова, 2000. 60 с.
15. Шелюг К., Ялова К. МЕТОДИ РОЗПІЗНАВАННЯ МОВЛЕННЯ. *IV International Scientific and Theoretical Conference «Current issues of science, prospects and challenges»*, Sydney, Australia, 5 трав. 2023.
16. Turn speech into text using Google AI. *Google Cloud*. URL: <https://cloud.google.com/speech-to-text> (дата звернення: 18.02.2026).
17. Introducing JSON. *JSON*. URL: <https://www.json.org/json-en.html> (дата звернення: 05.02.2026).
18. О.В. Дробик, В.В. Кідалов, В.В. Коваль, Б.Я. Костік, В.С. Лазебний, Г.М. Розорінов, Г.О. Сукач. «ЦИФРОВА ОБРОБКА АУДІО- ТА ВІДЕОІНФОРМАЦІЇ У МУЛЬТИМЕДІЙНИХ СИСТЕМАХ: навчальний посібник». - 2008.
19. Бердников О. М., Богуш К. Ю., Богуш Ю. П. Розпізнавання мовлення в системах штучного інтелекту. *Information Technology and Security*. Institute of Special Communication and Information Protection of National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

- 20.Методичні вказівки до виконання дипломної роботи освітнього рівня – бакалавр студентами усіх форм навчання для напряму підготовки 121 – Інженерія програмного забезпечення/ Укладачі : Петрик М.Р., Михалик Д.М., Кінах Я.І., Гладь С.В., Цуприк Г.Б. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 –28 с.
- 21.Андрухов І. А., Багрій-Заяць О. А. АСТОСУВАННЯ ВІДКРИТИХ АІ-ТЕХНОЛОГІЙ ДЛЯ МІНІМІЗАЦІЇ ЕКОНОМІЧНИХ ТА СОЦІАЛЬНИХ НАСЛІДКІВ В УМОВАХ ВОЄННОГО СТАНУ (НА ПРИКЛАДІ АВТОМАТИЗАЦІЇ КОМУНІКАЦІЙ ЗВО). *Збірник тез VI Міжнародної наукової конференції «Воєнні конфлікти та техногенні катастрофи: історичні та психологічні наслідки» (до 40-х роковин аварії на Чорнобильській АЕС)*, Тернопіль, Україна, 23 квіт. 2026. Тернопіль : ФОП Осадца Ю. В., 2026. С. 157–159.
- 22.Зеркалов Д.В. Безпека життєдіяльності. Навчальний посібник. - К.: Основа, 2011.
- 23.Жидецький В.Ц. Охорона праці користувачів комп'ютерів : підручник. Львів : Афіша, 2020. 176 с
- 24.Бедрій Я.І. Основи охорони праці : навч. посіб. 4-е вид. перероб. і доп. Тернопіль : Навчальна книга – Богдан, 2018. 240 с.
- 25.Безпека життєдіяльності та охорона праці : підруч. / В.В. Сокурєнко, О.М. Бандурка та ін. – Харків : ХНУВС, 2021. 308 с.

ДОДАТКИ

ДОДАТОК А



ДОДАТОК Б

*IX Міжнародна студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"*

Секція: **Інформаційні технології**
УДК 004.8:004.42:004.738.5
Андрухов І. – ст. гр. СП-41
Тернопільський національний технічний університет імені Івана Пулюя

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ГОЛОСОВОГО АСИСТЕНТА НА БАЗІ ASTERISK ТА GEMINI LIVE API ДЛЯ ПРИЙМАЛЬНОЇ КОМІСІЇ ЗВО

Науковий керівник: канд. техн. наук, доцент Багрій-Заяць О. А.

Andrukhov I.
Ternopil Ivan Puluj National Technical University

DEVELOPMENT OF AN INTELLIGENT VOICE ASSISTANT BASED ON ASTERISK AND THE GEMINI LIVE API FOR THE ADMISSIONS COMMITTEE OF A HIGHER EDUCATION INSTITUTION

Supervisor: O. A. Bahrii-Zaiats, Ph.D. in Engineering, Associate Professor

Ключові слова: штучний інтелект, голосовий асистент, IP-телефонія.
Keywords: artificial intelligence, voice assistant, IP telephony.

У період вступних кампаній працівники закладів вищої освіти стикаються з перевантаженням каналів комунікації через значну кількість однотипних звернень абітурієнтів. Традиційні IVR-системи не забезпечують достатнього рівня гнучкості та природності взаємодії. Метою роботи є розробка інтелектуального голосового агента для автоматизації телефонного консультування, новизна якого полягає у використанні потокової обробки аудіо та мультимодальної моделі штучного інтелекту в режимі, дуже близькому до реального часу.

Запропоновано підхід до побудови голосового асистента на основі інтеграції IP-телефонії та AI-моделей native audio від Google. Реалізована серверна частина забезпечує коректне розпізнавання української мови, аналіз намірів та генерацію відповідей із підтримкою контексту живої розмови. Особливістю підходу є використання відкритих технологій (зокрема Asterisk), що істотно знижує вартість впровадження. Система адаптована до предметної області консультацій абітурієнтів та включає механізм логування діалогів для постійної оптимізації моделі.

Використання великих мовних моделей у поєднанні з потоковими протоколами дозволяє реалізувати двосторонню комунікацію без розривів діалогу та з мінімальними затримками, близькими до реального часу. У процесі тестування розробленого агента виявлена готовність до успішного інтегрування у інфраструктуру приймальної комісії. Створене рішення може ефективно зменшувати навантаження на працівників без значних фінансових витрат. Перспективи розвитку проекту полягають у розширенні бази знань асистента шляхом інтеграції з базами даних університету для надання персоналізованої інформації про стан заяв. Також доцільним вбачається впровадження модулів аналізу емоційного стану абонента для вдосконалення характеру спілкування.

ДОДАТОК В

Посилання на репозиторій GitHub

Вихідний код розробленого програмного забезпечення:
<https://github.com/Ihorslav/tntu-call-agent>