



Boston University
Electrical & Computer
Engineering

Boston University
Electrical & Computer Engineering
EC 463 Senior Design Project

Second Prototype Testing Plan

BUtLAR



By
Team 12
Digital Human - Yobe

Team Members

Noa Margolin noam@bu.edu
Suhani Mitra suhanim@bu.edu
Jackie Salamy jesalamy@bu.edu
Andrew Sasamori sasamori@bu.edu

Required Materials:

Hardware:

- Raspberry Pi V5
- Two Røde Microphones
- LCD Screen (PHO 113 computer lab monitor)

Software:

- Shell Script
 - g++
 - Python Virtual Environment
- Live Audio Processing
 - *miniaudio_stream.c*
 - Utilizes C++ miniaudio library to capture audio in live time
- Yobe SDK (GrandE)
- Audio Generation
 - *IDListener_demo.cpp*
- Context-Specific Database
 - *school.db* (two table database with one about Photonics-based professors and course-specific information)
- Google ASR Speech-To-Text API, LLM API
 - *main.py*
 - Call the files below
 - *voiceAssistant.py*
 - Processes streamed audio and prepares for LLM processing
 - Performs API calls
 - *getAsssitance.py*
 - Corrects last names, generates SQL queries, provides LLM responses

Second Prototype Goal: supports a full conversation, employing live audio processing and low latency LLM-generated relevant responses.

Setup:

Our system setup begins with the hardware components: a Raspberry Pi connected via Ethernet to host the software on a Linux machine and two Rode Microphones for capturing audio input. The microphones are set at a standard of 9 inches apart, facing upward. The pipeline is driven by a shell script that automates the processes of audio capture, processing, and response generation. The backend workflow captures audio processes using Yobe's SDK in live time (Figure 1). The pipeline then performs speech-to-text transcription in live time. Once the full question is processed, the OpenAI-powered LLM generates a response based on public information for general questions. It utilizes our prompt engineering document for use-case-specific instances. For this test, we utilize a BU-specific database with information about certain professors' classes taught. Finally, the LLM-generated response is conveyed through a digital human, enabling seamless and interactive UI engagement.

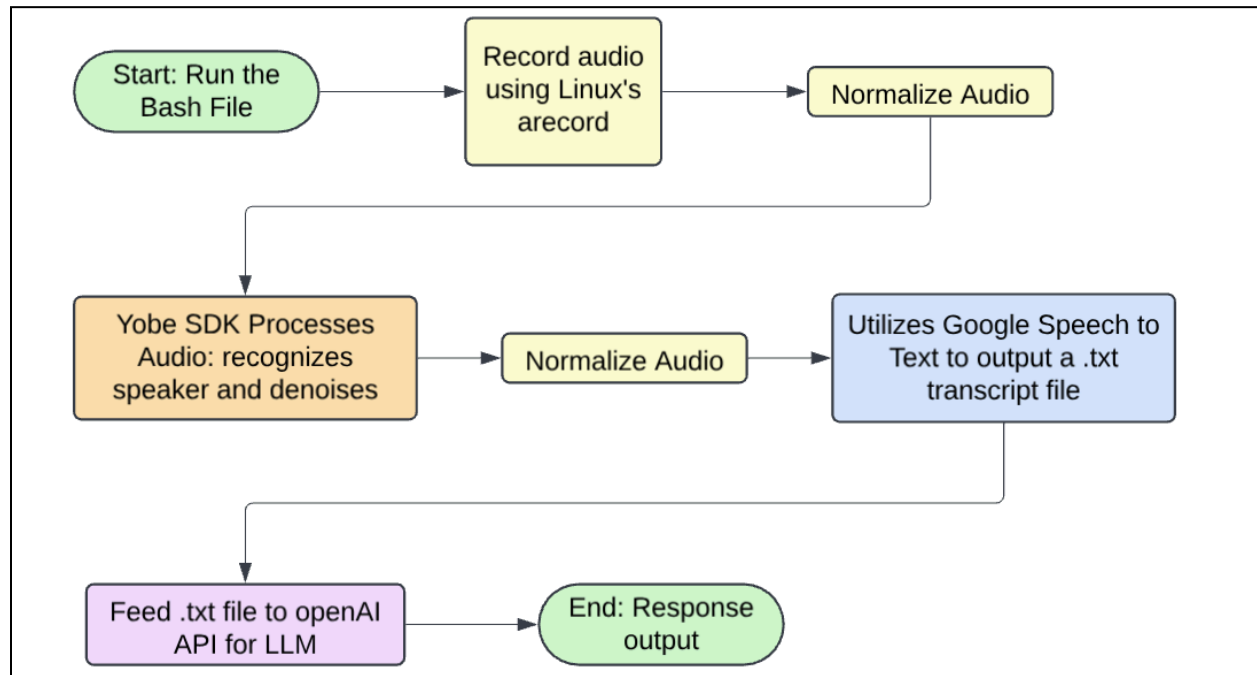


Figure 1: Illustration of Backend System Integration

Pre-Testing Setup Procedure:

Raspberry Pi Connection:

1. 2 AI-Micro Rode Dual Speakers are connected to Raspberry Pi.
2. Raspberry Pi is connected to the network via Ethernet.
3. Run and pipe the live audio streaming files on the Raspberry Pi.

Server-Side Connection:

1. Establish SSH connectivity with the Raspberry Pi (remote access) using the following command: **ssh yobe@128.197.180.176**
2. Navigate to the appropriate directory:
 - a. **cd BUtLAR_Voice-Powered-Digital_Human_Assistant/Audio/testing_audio**

Running the Session

1. Start a GCloud virtual environment:
 - a. **source ~/gcloudenv/bin/activate**
2. Begin live-streaming audio
 - a. **./miniaudio_stream | sox -t raw -r 16000 -e signed -b 16 -c 2 - -t raw -r 16000 -e signed -b 16 -c 1 - | python3 main.py**

Testing Procedure:

4 specific tests must be evaluated as either “Pass” or “Fail.” To achieve a “Pass,” each test must meet its unique criteria, ensure a latency of less than 5 seconds from the end of the audio recording to transcript generation, and produce a transcript that accurately conveys the intended message.

1. Live Audio Processing
 - a. Intakes audio in live time
2. Name Correction
 - a. Mispronounced names with inaccurate transcriptions are matched and output the correct last name
 - b. Last name is properly detected from the database list
3. Conversation Termination
 - a. Query thread ends when the user says, “Goodbye, BUtLAR.”
4. Multiple Queries
 - a. Users can ask multiple questions until they terminate the session.

Measurable Criteria:

Specific Test Case Requirements:

- I. Live Audio**
 - A. Speaker can ask questions in real-time.
 - B. We will say the sentence “Where is Professor Pisano’s office room?” and check the latency.
- II. Name Detection**
 - A. The transcript after Speech-To-Text will be checked for conveying a Professor’s correct last name.
 - B. We will say, “Where is Professor Eagle’s office room?” → Correct Name: Egele
- III. Loop Termination**
 - A. The conversation ends with the user saying, “Goodbye, BUtLAR.”
- IV. Multiple Queries**
 - A. Users can ask multiple questions in the same conversation thread.
 - B. We will first ask, “Who teaches Control Systems?”
 - C. We will then ask, “Where is the course EC 471 located?”

General Requirements:

In addition to satisfying the criteria above, the system must meet the following overarching requirements for every test case:

- **Latency:** The time from the end of the audio recording to the generation of the LLM-generated response must be less than 5 seconds.
- **Message Accuracy:** The transcript must accurately convey the intended message query.
- **Response Relevancy:** All answers provided must be accurate and relevant to the user’s question.

Score Sheet:

Requirement	Transcript is correct (Y/N)	Latency	Pass/Fail
Live Audio Processing Test 1	N/A	< 2 s:	
Live Audio Processing Test 2	N/A	< 2 s:	
Name Correction Test 1	(Print both wrong and corrected)	< 2 s:	
Name Correction Test 2	(Print both wrong and corrected)	< 2 s:	
Conversation Terminates Test 1	N/A	N/A	
Conversation Terminates Test 2	N/A	N/A	
Asking Multiple Queries Test 1	N/A	N/A	
Asking Multiple Queries Test 2	N/A	N/A	
Result →		/8	