



Boston University
**Electrical & Computer
Engineering**

Boston University
Electrical & Computer Engineering
EC 463 Senior Design Project

First 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
 - arecord (Linux binary/command)
 - g++
 - Python Virtual Environment
- Yobe SDK (GrandE)
- Audio Generation
 - IDListener_demo.cpp
 - normalize_wav.cpp
- Google Speech-To-Text API
 - googleTabulate.py
- OpenAI API
 - OpenAItesting.py

Prototype/MVP Goal: has backend pipeline integration of functioning noise immunity, personalized voice recognition, and gives relevant/ accurate 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 upwards. The pipeline is driven by a Bash script that automates the processes of audio capture, processing, and response generation. As depicted in Figure 1, the backend workflow captures audio, normalizes it, and processes it using Yobe's SDK. The pipeline then performs speech-to-text transcription, saving the resulting text to a .txt file. Subsequently, the OpenAI-powered LLM generates a response based on the public information for general questions and 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.

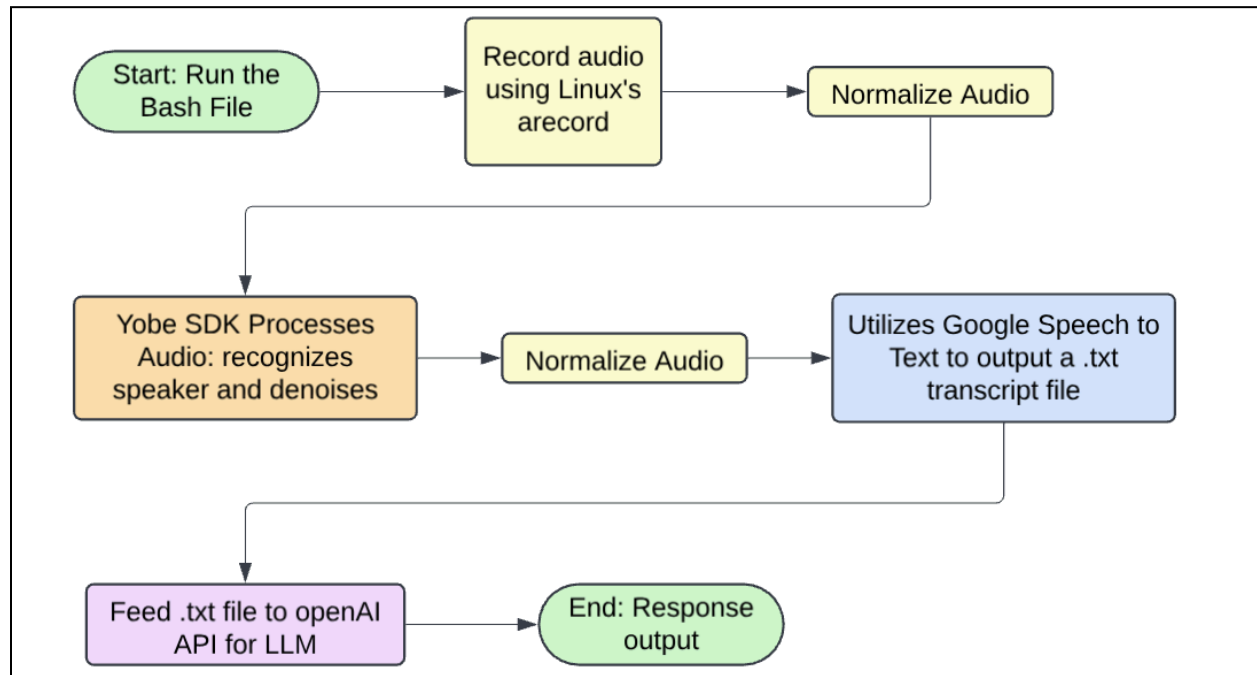


Figure 1: Illustration of Backend System Integration

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 the Bash script generate_wav.sh 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**

Running Bash Script

1. Execute permissions to the script generate_wav.sh: **chmod +x generate_wav.sh.**
2. Execute the script to begin recording: **./generate_wav.sh.**

Testing Procedure:

There are 8 specific tests that 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 12 seconds from the end of audio recording to transcript generation, and produce a transcript that accurately conveys the intended message.

1. Personalized voice recognition
 - a. Personalized Recognition Test 1: Andrew main speaker, Noa background
 - b. Personalized Recognition Test 2: Suhani main speaker, Jackie background
 - c. Play the post-processed audio file to verify this
2. Works with noise
 - a. Noise Environment Test 1: Crowd chatter at 3 volumes and play the output
 - b. Noise Environment Test 2: Beeping at 3 volumes and play the output
3. Relevant/accurate responses to questions that can be answered from the internet or specific BU questions i.e. our BU-specific database with classes taught.
 - a. First, ask general internet question:
 - i. Relevance Response General Test 1: About Boston
 - b. BU-specific question – has to be information accessible in the database
 - i. Relevance Response Use-Case Test 1: Classes taught
 - ii. Relevance Response Use-Case Test 2: Find me the office hours
4. Can connect to UI
 - a. Plays on D-iD, digital human accurately speaks LLM-generated response.

Measurable Criteria:

Specific Test Case Requirements:

- I. Personalized voice recognition**
 - A. Two speakers, one authorized and one not
 - B. Results will be verified by replaying the audio to confirm correctness.
- II. Works with noise**
 - A. The transcript after Speech-To-Text will be checked for conveying the correct message
- III. Relevant/ accurate responses** to questions that can be answered from the internet or specific BU questions i.e. classes taught and **Can connect to UI**
 - A. General and case-specific questions will be asked
 - B. The LLM response will be checked for conveying an accurate message
 - C. D-iD video speaks out the LLM response

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 12 seconds.
- **Message Accuracy:** The transcript must accurately convey the intended message query.

Score Sheet:

Requirement	Transcript is correct (Y/N)	Latency (<12s)	Pass/Fail
Personalized Recognition Test 1			
Personalized Recognition Test 2			
Noise Environment Test 1			
Noise Environment Test 2			
Relevance Response General Test 1			
Relevance Response Use-Case Test 1			
Relevance Response Use Case Test 2			
Can connect to UI (D-ID)			
Result →		/8	