Boston University
Electrical & Computer
Engineering

Boston University
Electrical & Computer Engineering
EC 463 Senior Design Project

Final Project 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
- Macbook Pro (Interface)

Software:

- 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* (database with information about course logistics — professors, time, location, etc.)
- Google ASR Speech-To-Text API, LLM API
    - *main.py*
        - Call the files below
    - *voiceAssistant.py*
        - Processes streamed audio and prepares for LLM processing
        - Handles user experience edge cases
        - Performs API calls
    - *txtToLLM.py, callLlm.py, manualCheck.py*
        - The txtToLLM.py script reads a user's question, corrects professor name spellings using lastNames() from callLlm.py, and corrects any other words calling using obviousMispellings() from manualCheck.py.
        - Using OpenAI, generateSql() (using corrected professor names version) creates a SQL query based on the school.db course table.
        - Output is turned into a natural sentence in respondToUser()
- Django Framework (Notable Files)
    - *consumers.py*
        - Text-to-speech (TTS) and socket handling
    - *butlar_interface.html*
        - Creates web interface

Final Project Testing Goal: supports a full conversation, employing live audio processing and low-latency LLM-generated relevant responses. Our focus does not lie in our user interface, but rather in the back-end functionality.

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 in real-time (Figure 1). The pipeline then performs speech-to-text transcription in real-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 speaker, enabling seamless and interactive UI engagement through a web interface.

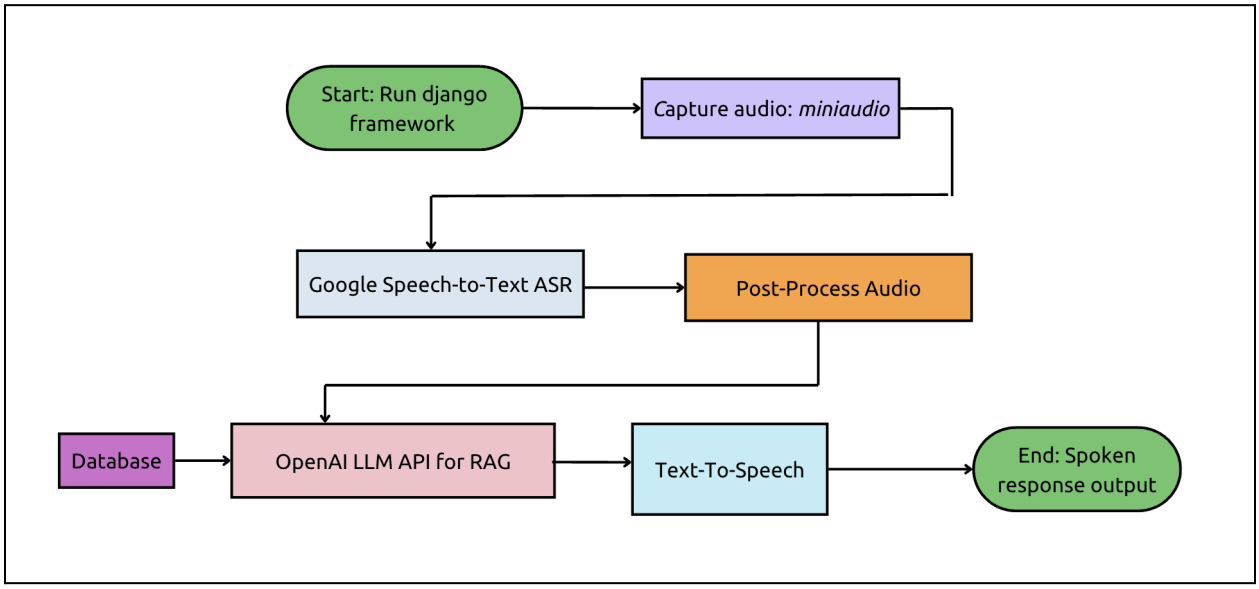

Figure 1: Illustration of Backend System Integration

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/django_top`

Running the Session

1. Start a GCloud virtual environment:
    a. `source ~/gcloudenv/bin/activate`
2. Start running server
    a. `daphne -b 127.0.0.1 -p 8000 django_top.asgi:application`
3. Access the server here and press
    a. `http://127.0.0.1:8000/butlar/interface/`

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. Specific (Location-based Query, Class Type-based) Queries
    a. Prompt BUtLAR using a Class ID & name
    b. Prompt BUtLAR using a course type (lab/discussion/lecture)
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. Multiple Queries (Miscellaneous Prompts)
    a. Users can ask multiple questions until they terminate the session
    b. Prompting questions with a less apparent/straightforward answer

Measurable Criteria:

Specific Test Case Requirements:

I. **Specific (Location-based Query, Class Type-based) Queries**
   A. Prompt BUtLAR using a Class ID & name
      1. "When is EC414?"
      2. "When is machine learning?"
   B. Prompt BUtLAR using a course type (lab/discussion/lecture)
      1. "When is EC414 discussion?"
      2. "Where is EC 311 lab?"

II. **Name Detection**
   A. The transcript after Speech-To-Text will be checked for conveying the Professor's correct last name.
   B. We will say, "What does Professor Eagle Teach?" → Correct Name: Egele
   C. We will say, "What does Tally Moret teach?" → Correct Name: Tali Moreshet

III. **Multiple Queries (Miscellaneous Prompts)**
   A. We will first ask, "How many discussion sections does EC414 have?"
   B. We will then ask, "What class is on Tuesdays and Thursdays from 1:30 to 3:15 PM in CAS 227?"

   *Conversation will terminate when the user says, "Goodbye, BUtLAR."

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.
- **Conversation Termination:** Conversation terminates when the user says, "Goodbye, BUtLAR."
- **Timeout:** A session will timeout if no user speaking is detected for 45 seconds.

Score Sheet:

Requirement	Transcript is correct (Y/N)	TTS output is correct (Y/N)	Latency	Pass/Fail
Specific Query Test 1	N/A		< 4 s:	PASS
Specific Query Test 2	N/A		< 4 s:	PASS
Name Correction Test 1	(Print both wrong and corrected)		< 4 s:	PASS
Name Correction Test 2	(Print both wrong and corrected)		< 4 s:	PASS
Asking Multiple Queries Test 1	N/A		N/A	PASS
Asking Multiple Queries Test 2	N/A		N/A	PASS
Result →			6/6	