System Test Plan

For

Audio Surveillance Project

Team members: Jacob Attia, Caleb Leeb, Zachary Tauscher, Jaclyn Welch, Jorge Garcia

Version/Author	Date
Initial(Fall)/Caleb Leeb - Jaclyn Welch - Zak Tauscher	10 19 2021
Final(Fall)/Caleb Leeb, Jaclyn Welch, Zak Tauscher, Jacob Attia	12 06 2021
Initial (Spring)/Caleb Leeb - Jaclyn Welch - Zak Tauscher - Jacob Attia - Jorge Garcia	02 27 2022
Final (Spring)/Caleb Leeb, Jaclyn Welch, Zak Tauscher, Jacob Attia, Jorge Garcia	04 15 2022

Table of Contents

1. Introduction

1.1 Purpose

This document is a test plan for Audio Surveillance Project System Testing, produced by the System Testing team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

1.2 Objectives

- Meets the requirements, specifications and the Business rules.
- Supports the intended business functions and achieves the required standards.
- Satisfies the Entrance Criteria for User Acceptance Testing.

2. Functional Scope

The Modules in the scope of testing for the Audio Surveillance System Testing are mentioned in the document attached in the following path :

- The System Requirements Specification document:
  System Requirements Specification.docx

3. Overall Strategy and Approach

3.1 Testing Strategy

Audio Surveillance System Testing will include testing of all functionalities that are in the scope (Refer Function Scope Section). System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal & external interfaces.

3.2 System Testing Entrance Criteria

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into usability testing and function testing.

3.3 Testing Types

3.3.1 Usability Testing

User interface attributes, cosmetic presentation and content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application. (e.g., access keys, consistent tab order, readable fonts, etc.)

3.3.2 Functional Testing

The objective of this test is to ensure that each element of the component meets the functional requirements of the business as outlined in the:

- Business / Functional Requirements
- Business rules or conditions
- Other functional documents produced during the course of the project i.e. resolution to issues/change requests/feedback

3.4 Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

3.4.1 Suspension Criteria

If the user interface fails to display the locational data then the algorithm is faulty and testing will be halted.

If the locational data outputted by the algorithm is not within an acceptable range, testing will be halted.

If the program fails to classify the recorded audio then the algorithm is faulty and testing will be halted

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. If testing is halted, and changes are made to the hardware, software or database, it is up to the Testing Manager to determine whether the test plan will be re-executed or part of the plan will be re-executed.

3.4.2 Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully.

4. Execution Plan

4.1 Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their inter-dependency.

Requirement	Test Case Identifier	Input	Expected Behavior	Pass/ Fail
The system will take in audio data with the microphone arrays.	1.1	The system will be placed in a quiet lab room where people are talking and walking.	The computer connected to the microphone array will show an input file in .wav format	
The system will calculate time delay of arrival for each microphone	2.1	The system will be placed in a quiet lab room where people are talking and walking	The computer connected to the microphone array will show the data in a .csv file	
The system will pass audio data to the	3.1	The data is transferred	The Raspberry Pi will show the audio	

Raspberry Pi from the microphone arrays via bluetooth.			data from the separate microphone arrays.	
The system will pass the audio data from the Raspberry Pi to the host computer	4.1	The data is transferred	The host computer will receive output from the microphone array.	
The system will output the locational data of sounds in the room from the microphones	5.1	The algorithm is run on the host computer.	The host computer displays the coordinates of the sound in relation to the microphone array.	
The system will output classification of recorded audio.	6.1	The algorithm is run on the host computer	The host computer displays the classification of the recorded audio.	
The system will visualize the locational data on a grid map as points.	7.1	The algorithm is run on the host computer	User interface updates and displays locational data as coordinate points on a grid map.	
The system will display the classification of the audio source at the location received.	8.1	The algorithm is run on the host computer	The classification is displayed on the grid map.	
The user is authenticated to be an authorized user.	9.1	The user inputs a valid and invalid security ID	The user is granted or denied access.	
The system creates a database containing the time delay of arrival data and sound files that will be used by the algorithm to produce locational data.	10.1	The algorithm is run on the host computer	Audio data is produced	

Table 4.1 This table

5. Traceability Matrix & Defect Tracking

5.1 Traceability Matrix

List of requirement, corresponding test cases

Requirement CRITICAL: “The system will take in audio data with the microphone arrays.”

Test Cases: Check that the microphone arrays are receiving and passing sound.

Requirement CRITICAL: “The system will calculate the time delay of arrival for each microphone.”

Test Cases: Check that the data being passed to the local computer includes the time delay of arrival.

Requirement CRITICAL: “The system will pass audio data to the Raspberry Pi from the microphone arrays via bluetooth.”

Test Cases: Check that the Raspberry Pi is receiving and passing audio data.

Requirement CRITICAL: “The system will pass the audio data from the Raspberry Pi to the host computer.”

Test Cases: Check that the local computer is receiving the audio data from the Raspberry Pi.

Requirement CRITICAL: “The system will output the locational data of sounds in the room from the microphones.”

Test Cases: Check that the system’s output of the locational data reflects the actual location that the sound came from.

Requirement CRITICAL: “The system will output classification of recorded audio.”

Test Cases: Check that the classification the system gives of the sound reflects what sound was actually taken in by the system.

Requirement CRITICAL: “The system will visualize the locational data on a grid map as points.”

Test Cases: Check that the system outputs a map that shows the room and identifies any sounds.

Requirement CRITICAL: “The system will display the classification of the audio source at the location received.”

Test Cases: Check that the system outputs the type of sound on the map and that they match the sound that was taken in.

Requirement MEDIUM: “The user is authenticated to be an authorized user.”

Test Cases: Check the user’s identity and compare it to a list of authorized users.

Requirement CRITICAL: “The system creates a database containing the time delay of arrival data and sound files that will be used by the algorithm to produce locational data.”

Test Cases: Check the database that is output from the system and ensure that it matches the sound that was recorded.

5.2 Defect Severity Definitions

Critical	The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows: ● System abends ● Data cannot flow through a business function/lifecycle ● Data is corrupted or cannot post to the database
Medium	The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows: ● Form navigation is incorrect ● Field labels are not consistent with global terminology
Low	The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows: ● Repositioning of fields on screens ● Text font on reports is incorrect

6. Environment

6.1 Environment

- The System Testing Environment will be used for System Testing.

In order to conduct the testing the tester needs the following

- Access to database of microphone data
- Have Python installed with required libraries
- Access to microphone array and host computer
- A quiet room

7. Assumptions

This section lists assumptions made during testing

- User has valid ID
- The room is void of all other audio sources
- All data collected is accurate

8. Risks and Contingencies

Risk #	Risk	Impact	Contingency Plan
1	Bluetooth incompatibility	Low	Use cables
2	Incompatible versions of Python on user's system	High	Include which version of Python is needed to run system in user documentation
3	Microphone is obstructed by obstacles in the room, interfering with sound transmission	Medium	User documentation suggests that best use case requires that system be set up in an area without obstruction