CS476 Requirements Specification

Revision 2.2

November 12, 2024

StratoSplit

Client:

General Dynamics Mission Systems

Mentor:

Brian Donnelly

Team Members:

Sam Cain

Nolan Newman

Dallon Jarman

Elliot Hull

Accepted as baseline requirements for the project:
For the client: | For the team: Samuel Cain

Overview:

The purpose of this requirement specification document is to establish the set of requirements so that the developed solution will meet the client's needs. This document will provide a thorough understanding of the project's scope, goals, and execution plan.

1.Definitions

1.1 Acronyms

AWS - Amazon Web Services
GDMS - General Dynamics Mission Systems
GPS - Global Positioning System
NIST - National Institute of Standards and Technology
SAR - Search and Rescue
USGS - United States Coast Guard

2. Introduction

Effective communication is paramount in defense, public safety, and intelligence communities. The StatoSplit project aims to address the challenge of inefficient communication with radio modems on mobile devices. Inspired by General Dynamics Mission Systems' innovative solutions, our team is developing a Node.js web application, StatoSplit, to simulate audio generation and transmission to a dashboard for real-time monitoring.

General Dynamics Mission Systems, a leading defense contractor, specializes in mission-critical products and systems. Their notable projects include Rescue 21, a Coast Guard distress location system, and the next-generation Global Positioning System (GPS). GDMS has tasked our team with creating a solution that enhances communication efficiency.

Current communication with radio modems relies on cumbersome web interfaces, hindering efficient data transmission and attachment management on mobile devices. StatoSplit seeks to overcome these limitations by developing a user-friendly web application. StratoSplit seeks to design and develop a simulated audio generator that will simulate the environment of the U.S. Coast Guard communication network between sailors and the Coast Guard. From there, a real-time audio transmission dashboard using Node.js will be implemented for audio monitoring to allow for effective and efficient communications between sailors and operators. The key features are as follows, simulated audio generation, real-time audio transmission, a dashboard for monitoring, and a user-friendly interface. The functional requirements are as follows, the application shall simulate audio generation, the application shall transmit audio to a dashboard in real time, and the application shall provide a user-friendly interface. Lastly, the non-functional requirements are as follows: the application shall ensure high availability and uptime, maintain optimal performance under various network conditions, and adhere to industry-standard security protocols.

StatoSplit aims to revolutionize communication in defense, public safety, and intelligence communities by providing a secure, efficient, and user-friendly web application for simulated audio generation and transmission. Our team is committed to delivering a high-quality solution that meets the needs of GDMS and its clients.

3. Problem Statement

The client currently utilizes a local application approach to control radio transmission through a local application-based approach. Even though the clients have a working system in place, there are multiple issues with it. The current setup restricts control to a single, designated device connected to the radio towers, originally designed for audio transmission. This exclusive dependency creates numerous limitations. **Figure 1** below illustrates how the current system works.

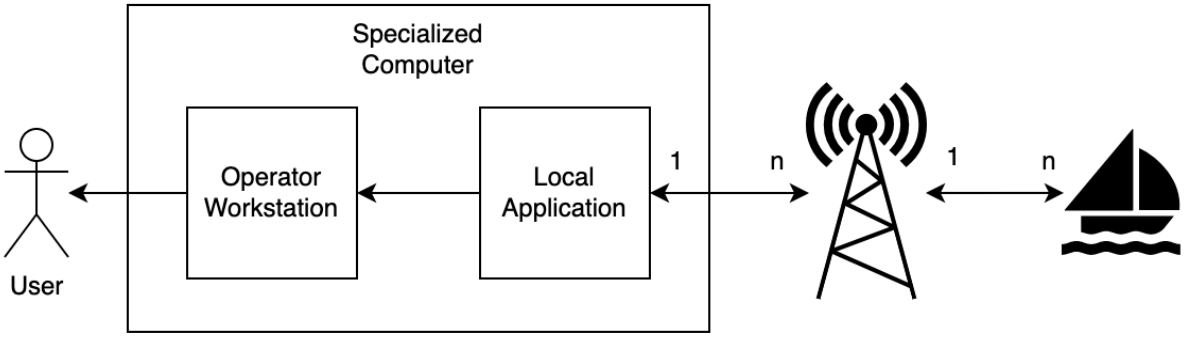

Figure 1: Current GDMS SAR System

Some issues with the current interface identified by GDMS are as follows:

- The current mobile system is not the most user friendly which is an issue since the client wants the system to require minimal training to use.
- Limited device capabilities will hinder the adaptability of the application, restricting its potential for cross-platform compatibility.
- The current system is not future-proof and lacks the ability for the application to be easily updatable.
- The system has very limited user flexibility and does not allow for a diverse array of platforms to access the application.

4. Solution

The following diagrams illustrate the interactions between components, the flow of data, and user access. **Figure 2**, provides an overview of the systems infrastructure, showing how components interact within a cloud-hosted environment. **Figure 3**, offers a detailed view of user interaction with the system, demonstrating how the operator workstation, web application, and audio output devices are integrated to deliver an enhanced audio experience. Together these diagrams depict a comprehensive and modular architecture that ensures high performance, robust security, and ease of scalability. The solution is designed with flexibility in mind, allowing for future expansion, maintainability, and reliability.

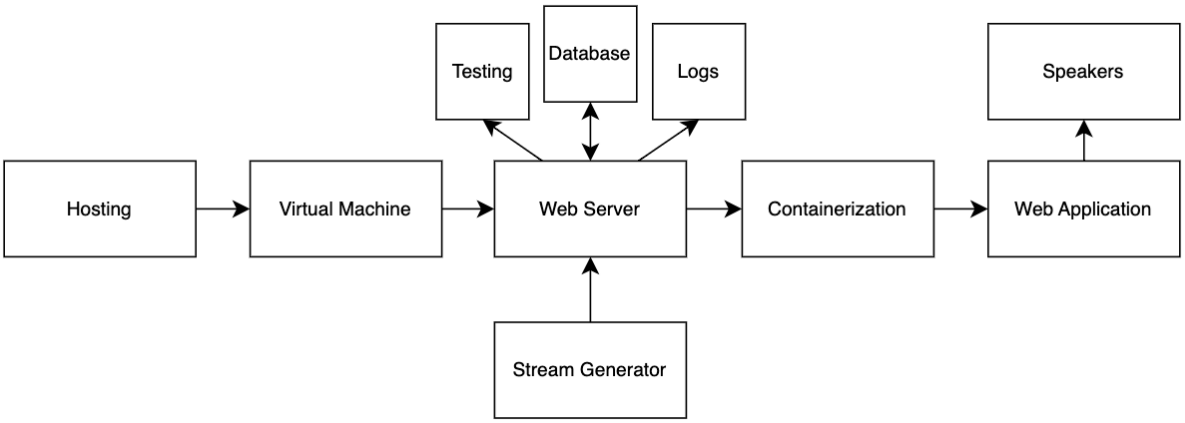

Figure 2: An example workflow of the proposed solution

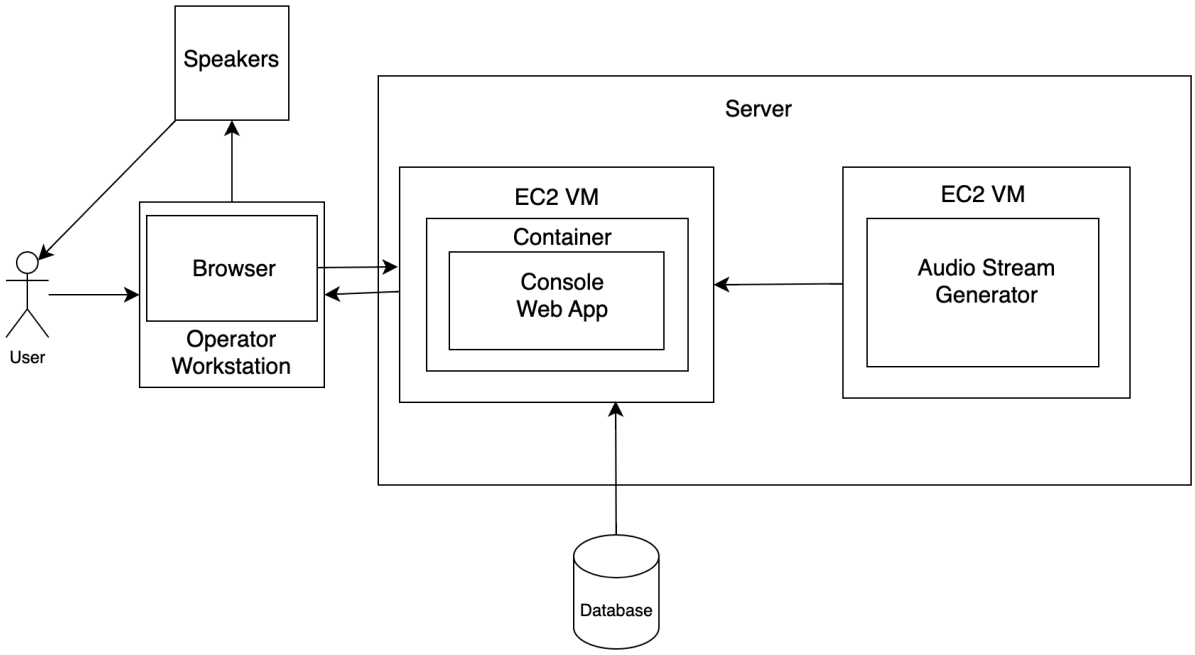

Figure 3: A comprehensive system diagram of the proposed solution

Workflow (Figure 1)

The workflow begins with a cloud-hosted environment providing the infrastructure for deploying virtual machines for the system's core applications. Within this environment a virtual machine hosts a central web server that acts as a hub, coordinating various components and managing interactions between a database, a logging module, and a testing component. The web server receives real-time audio from a stream generator, which simulates various audio inputs. The web server interfaces with a containerization system to deploy isolated instances of the web application, allowing it to manage user interactions and send audio output to connected speakers.

System Diagram (Figure 2)

The system diagram illustrates component interactions. Users access the system via a browser on an Operator Workstation and authenticate via a console web application hosted in a container on an EC2 virtual machine. After authentication, the application communicates with an audio stream generator on a separate EC2 instance, providing users with specific audio streams based on permissions. These authorized streams are mixed into spatial audio and output through speakers connected to the workstation, creating an immersive audio experience. This AWS-hosted setup enables secure, isolated user sessions with protected communication channels.

Summary

This solution provides a secure, scalable, architecture for real-time audio streaming and control, using AWS-hosted virtual machines, containerized applications, and a database for user authentication and data management. The architecture integrates logging, automated testing, and modular containerization to support high performance, security, and scalability. Designed for flexibility, it allows for future expansions like mobile support and advanced audio processing, delivering a secure, reliable, and immersive experience for users.

5. Project Requirements

The requirements section is designed to provide a comprehensive view of the specifications and goals for this project, organized into distinct types – functional, performance, and environmental requirements. This structure allows for a focused examination of each requirement type according to its purpose and scope, ensuring clear guidance for development, testing, and compliance.

Each requirement type is formatted with a table that outlines the requirements, its priority level, its goal metric or priority, and its functional area. These functional areas are organized as follows and where necessary within each table: security, user, administration, usability, audio, logging, and budget. Requirements that are considered essential are left unmarked, while *stretch goals* – additional features that may be pursued if time and resources allow - are italicized in both the table and the requirement's section title. This consistent formatting distinguishes core requirements from aspirational goals, allowing for flexible project management that adapts to available resources and project timeline.

Functional requirements (5.1) focuses on specific features and capabilities of the system, with each requirement structured to include a user story, description, acceptance criteria, assumptions, and error handling. The user story contextualizes the requirement from the perspective of the end-user providing insight into the intended functionality. Acceptance criteria clearly outline the conditions for meeting the requirement, while assumptions specify any underlying conditions necessary for successful implementation. Error handling describes the system's response to any failures, ensuring that each feature is robust and user-friendly.

Performance requirements (5.2) centers on metrics and thresholds that define the system's operational standards. Each performance requirement is detailed in terms of goal metrics, such as response times, latency, and capacity limits, which set quantifiable benchmarks for system performance. These goal metrics ensure that the system meets standards of reliability, efficiency, and responsiveness, critical for maintaining a seamless user experience in high-stakes environments. Performance requirements also include acceptance criteria that outline specific metrics or thresholds ensuring the system meets or exceeds the necessary standards, while assumptions specify underlying conditions necessary for success.

Environmental requirements (5.3) focuses on compliance and infrastructure needs, including security protocols, data residency, and compatibility with specific software or hardware. Environmental requirements use a compliance objective, which defines the specific regulatory or policy goals of each requirement, such as adhering to NIST standards and maintaining secure storage within AWS GovCloud. Each requirement includes implementation criteria, which illustrates actions or configurations needed to meet the compliance objective. This structure ensures that environmental requirements align with regulatory standards and organizational policies. Assumptions for environmental requirements typically include any necessary conditions for compliance.

The requirements section provides a structured, detailed roadmap for the project, ensuring each requirement type is fully addressed and clearly documented. By including *stretch* goals and distinguishing the different needs and criteria of functional, performance, and environmental requirements, this section allows for targeted development and reliable compliance tracking, supporting a high-quality, secure, and user-friendly end product.

5.1 Functional Requirements

Requirement	Priority Level	Functional Area
5.1.1 Encrypt Database Passwords	High	Security
5.1.2 User Login	High	User
5.1.3 Change Password	High	User
5.1.4 Detect User Logout	High	User
5.1.5 Enforce Strong Passwords	Medium	User
5.1.6 Log User Access Incidents	Medium	User
5.1.7 Delete Account	Medium	User
5.1.8 Reconnect Without Re-authentication	Medium	User
5.1.9 Control of Channel Assignment	High	Administrator
5.1.10 Management of User Accounts	High	Administrator
5.1.11 View Channels Available to Users	High	Administrator
5.1.12 Browser Access to Logs	Medium	Administrator
5.1.13 Mix Audio Streams in 3D	High	Audio
5.1.14 Mute channels	High	Audio
5.1.15 Mute all active channels	Medium	Audio
5.1.16 Unmute all muted channels	Medium	Audio
5.1.17 Save channel configurations	Medium	Audio
5.1.18 Load channel configurations	Medium	Audio
5.1.19 Clear current configuration	Medium	Audio
5.1.20 Adjust individual channel volume	Medium	Audio
5.1.21 Adjust virtual speaker location	Medium	Audio
5.1.22 Log errors, access, and activity	High	Logging
5.1.23 Alerts for audio loss	Medium	Logging
5.1.24 *Enable two way communication*	Low	User
5.1.25 *Map with radio tower locations*	Low	Usability

Table 1: Functional Requirements, Priority, and Functional Area

5.1.1 Encrypt Database Passwords

User story: As a SAR system developer, I want passwords in the database to be securely encrypted, so user credentials stay protected even if there is a data breach. This measure ensures that only authenticated users can access their accounts. Password encryption should follow current industry security standards.

Requirement description: The system must encrypt all stored passwords using secure hashing protocols, ensuring protection in case of a database breach. Encryption must meet industry standards and be verified before any credentials are stored. This approach helps ensure that passwords remain secure and unreadable in the event of a database security compromise.

Acceptance criteria:

- All passwords are stored in an encrypted format that meets current security standards
- Password encryption occurs at the point of creation or update and is verified immediately

Assumptions:

- Encryption follows security standards compliant with NIST

Error handling:

- If encryption fails during password storage, prevent password saving and alert the user

5.1.2 User Login

User story: As a SAR system user, I want to securely log into the system with my unique credentials so that I can access critical SAR features quickly and safely. This allows me to start my tasks without delay and with confidence in the security of my data. The system should ensure secure authentication to protect my account.

Requirement description: The system must provide a secure login mechanism that validates user credentials against encrypted records stored in the database. Login attempts, both successful and failed, are recorded in access logs for monitoring and security. This secure authentication process is critical for maintaining access control and protecting sensitive information.

Acceptance criteria:

- Users can log in using unique, registered credentials
- Only authenticated users with valid accounts can access the system
- Each login attempt is recorded in an access log, including successful and failed attempts

Assumptions:

- Each user has a unique identifier (username or email)
- Login attempts are limited to prevent brute-force attacks

Error handling:

- Display a clear message on login failure
- After three failed attempts, prompt the user to reset their password

5.1.3 Change Password

User story: As a SAR system user, I want to be able to change my password whenever I feel it is necessary, like in case of a security concern. This option helps me maintain my account's security and feel assured that my access is protected. The system should store my new password in a securely encrypted format.

Requirement description: Users must have the ability to securely change their passwords whenever needed, with new passwords stored in encrypted format in the database. The system should verify the user's current password before allowing changes and enforce minimum strength requirements on the new password. Each password change is logged, helping ensure accountability and supporting system security.

Acceptance criteria:

- Users can change their password through a secure form after providing their current password
- Password changes are recorded in an access log
- The new password meets enforced security criteria before acceptance

Assumptions:

- Users will be guided on password strength requirements during the change process

Error handling:

- Display error messages if the new password does not meet criteria
- If the current password is incorrect, alert the user without revealing which field was incorrect

5.1.4 Detect User Logout

User story: As a SAR system administrator, I want the system to detect when a user logs out or has been inactive for a specified time to help keep accounts secure. This ensures unauthorized users can't access an unattended session, protecting sensitive data. The system should handle logouts promptly for increased security.

Requirement description: The system must detect both intentional logouts and inactivity-based timeouts to secure unattended sessions and prevent unauthorized access. The system should also warn users before automatic logout due to inactivity, allowing them to extend the session if needed. All logout events are recorded in access logs for tracking and monitoring purposes.

Acceptance criteria:

- Users are logged out after a set period of inactivity.
- Manual logouts are detected and recorded in an access log

Assumptions:

- Inactivity timeouts are predefined based on system security requirements

Error handling:

- Display a warning before session timeout to allow the user to extend the session

5.1.5 Enforce Strong Passwords

User story: As a SAR system user, I want the system to require strong passwords so that my account is difficult to breach. This adds an essential layer of protection to my access and reduces the risk of unauthorized use. Password strength should meet security requirements including special characters and a minimum length.

Requirement description: The system must enforce strong password requirements guiding users to create secure passwords that meet defined criteria, such as minimum length, special characters, and mixed-case letters. This reduces vulnerabilities to unauthorized access

and enhances the overall security of user accounts. Users receive feedback on whether their passwords meet these criteria before they are saved.
Acceptance criteria:

- Users cannot save a password that does not meet the required strength
- Clear feedback is given on password requirements during account creation or password change

Assumptions:

- Password strength requirements align with the United States Coast Guard's security policy

Error handling:

- Display specific error messages if password criteria aren't met

5.1.6 Log User Access Incidents

User story: As a sar system administrator, I need user access incidents logged securely, including logins, password changes, and failed login attempts. This information helps monitor for unusual activities and supports accountability for all access actions. Logs should be stored in a way that allows for easy review when needed.

Requirement description: The system must log all access-related actions (logins, logouts, password changes, and failed attempts) in a secure, centralized location. These logs enable monitoring and analysis of user activity to help detect security issues and ensure compliance with access policies. Regular logging of access incidents supports accountability and helps in identifying potential security threats.
Acceptance criteria:

- Each login, logout, and password change is recorded with timestamps and user ID
- Access logs are securely stored and can be retrieved as needed

Assumptions:

- Logs are regularly reviewed for security monitoring purposes

Error handling:

- If logging fails, the system will alert the administrator and attempt to resolve or retry logging actions

5.1.7 Delete Account

User story: As a SAR system user, I want to delete my account securely if I no longer need to access the platform. I want my data to be removed and my access to be closed. This helps me manage my information and ensure accountability. Account deletion should be confirmed and securely processed.

Requirement description: Users must have the option to delete their accounts permanently, with a confirmation prompt to prevent accidental deletions. Once confirmed, the system securely removes all associated user data from the database, making the account inaccessible. The deletion event is recorded in the logs, supporting privacy compliance and user data management.
Acceptance criteria:

- Users can delete their accounts after confirming their decision
- Deletion is recorded in the access log with a timestamp

- The user's data and credentials are removed from the database making the account inaccessible

Assumptions:

- Account deletion is irreversible

Error handling:

- Display a clear message on login failure
- After three failed attempts, prompt the user to reset their password

5.1.8 Reconnect Without Re-authentication

User story: As a SAR operator, I want to reconnect to the system automatically after a brief network drop, so I can continue my work without needing to log back in. This keeps my workflow seamless during operations where continuity is crucial. The system should maintain my session security during short disconnections.

Requirement description: The system should automatically allow users to reconnect after a brief network interruption without requiring a new login, provided reconnection occurs within a defined time limit. This enables seamless workflow continuity for operators. If the reconnection time exceeds this limit, users are required to re-authenticate to maintain security.

Acceptance criteria:

- Users can reconnect within a set time without re-authentication
- The session timeout countdown resets upon reconnection

Assumptions:

- Session reconnection is allowed within a specific time to maintain security

Error handling:

- If the user cannot reconnect within the time limit, require a fresh login

5.1.9 Control of Channel Assignment

User story: As a system administrator, I want the ability to assign audio channels to specific users so that each operator has access only to the channels relevant to their role. This control allows me to manage channel distribution efficiently, ensuring users focus on the appropriate communications. The system should provide intuitive tools for assigning and updating channel access for each user.

Description: The system must enable administrators to assign specific audio channels to individual users or groups based on operational needs. This feature helps ensure that users access only the channels they require for their tasks, enhancing workflow organization and reducing unnecessary audio clutter. Administrators should be able to modify assignments quickly, adapting to changes in staffing or operational demands.

Acceptance criteria:

- Administrators can view and assign audio channels to individual users or user groups
- Channel assignments can be updated or removed as needed
- Users only see and access channels assigned to them in their dashboard

Assumptions:

- Administrators have a dedicated interface or permissions for assigning channels

Error handling:

- If an assignment fails, notify the administrator and allow them to retry or select a different

5.1.10 Management of User Accounts

User story: As a system administrator, I want to manage user accounts within the audio platform so that I can add, edit, or deactivate users as needed. This helps me keep the user list current and ensures that only authorized personnel have access. The system should provide secure options to manage account information and permissions efficiently.

Description: The system must allow administrators to manage user accounts, including creating new accounts, updating user information, and deactivating accounts as needed. This functionality is essential for maintaining security and access control, especially in dynamic SAR environments where personnel changes may occur frequently. Admins should have access to secure tools that let them configure user roles and permissions in a streamlined way.

Acceptance criteria:

- Administrators can create, edit, or deactivate user accounts within the system.
- Account changes, such as activation or deactivation, are reflected immediately in user access
- User management actions are logged for accountability

Assumptions:

- User accounts are securely stored and comply with role-based access controls

Error handling:

- If account management actions fail, notify the administrator and allow them to retry

5.1.11 View Channels Available To Users

User story: As a system administrator, I want to view all the audio channels available to each user so I can ensure they have the correct permissions and access. This visibility allows me to audio channel assignments and make adjustments as necessary. The system should provide a clear and organized view of channel access for each user.

Description: The system must enable administrators to view the list of audio channels accessible to each user, providing a detailed overview of current permissions. This feature supports auditing and verification, helping administrators confirm that each user has the appropriate channels for their role. The channel view should be easy to access and present information in an organized user-friendly format.

Acceptance criteria:

- Administrators can view a list of channels assigned to each user
- The list is accessible in a centralized location and includes relevant details (e.g., channel name, assignment date).

Assumptions:

- Channel access information is updated in real-time and reflects the latest assignments

Error handling:

- If channel access data fails to load, notify the administrator to refresh or retry

5.1.12 Browser Access To Logs

User story: As a system administrator, I want to access system logs through a browser so that I can monitor user actions and troubleshoot issues remotely. This flexibility allows me to manage and audit the system without needing specialized software. The system should make logs accessible from any secure, authorized browser.

Description: The system must allow administrators to access system logs through a web browser, making it easy to monitor activities, review user actions, and troubleshoot issues from any location. This feature is especially useful for remote administration, providing flexibility for admins to view and manage logs securely. Logs should be organized by date, user actions, and other relevant filters to streamline the audit process.

Acceptance criteria:

- Administrators can access system logs from an authenticated browser
- Logs are searchable by data, user, and type of action
- Log access complies with security standards to protect sensitive data

Assumptions:

- Browser-based access is available through secure login

Error handling:

- If logs fail to load, notify the administrator, log the error, and provide a retry option
- Log to a file and to the database providing an option to evaluate either

5.1.13 Mix Audio Streams in 3D

User story: As a SAR operator, I want the ability to to mix multiple audio streams in a 3D space, so I can spatially differentiate between sources for better situational awareness. The spatial audio setup enables me to locate and prioritize channels quickly and accurately in mission-critical scenarios. The system should support intuitive 3D audio mixing to enhance the clarity of communications.

Description: The system must provide real-time, spatial 3D audio mixing for multiple audio streams, allowing users to differentiate between various communication sources by assigning each stream a unique virtual location. This functionality helps operators interpret audio inputs more naturally, supporting rapid decision making. The spatial audio should be adjustable and responsive to maintain clear audio resolution and directionality

Acceptance criteria:

- Users can assign virtual spatial locations to each active audio stream
- The system processes audio streams in real-time with minimal latency
- Users can clearly distinguish between streams based on assigned spatial locations

Assumptions:

- Spatial audio mixing will be rendered through compatible hardware (e.g., multi-channel speakers or headphones)

Error handling:

- If an audio stream cannot be assigned a spatial location, alert the user and provide a default location

5.1.14 Mute Channels

User story: As a SAR operator, I want to mute specific audio channels to reduce distractions and focus on high-priority communications. This helps me manage auditory input, particularly in high-stress environments with multiple audio sources. The system should allow quick and easy muting of individual channels as needed.

Description: The system must enable users to mute individual audio channels selectively. Muting should be easily accessible to reduce unwanted noise without permanently

altering channel configurations. This feature aids operators in concentration on relevant channels by removing less critical audio sources from their immediate attention.
Acceptance criteria:

- Users can mute and unmute any active audio channel
- The system visually indicates muted channels to distinguish them from active ones.

Assumptions:

- Muted channels can be reactives as needed without resetting other audio settings

Error handling:

- If a mute request fails, notify the user and retry the action

5.1.15 Mute All Active Channels

User story: As a SAR operator, I want the ability to mute all active audio channels simultaneously so that I can eliminate background noise when necessary. This feature helps me focus or address external interruptions without individually muting each channel. The system should allow me to mute all channels quickly and restore them when needed.

Description: The system must provide a feature to mute all active audio channels at once, silencing all sounds without adjusting each channel individually. This feature is useful in situations where operators need immediate silence, allowing them to manage auditory input efficiently. Muting all channels should be easy to activate, and users should be able to unmute channels individually or all at once afterward.
Acceptance criteria:

- Users can mute all active channels with a single action
- A visual indicator shows that ll channels are muted
- Individual channels can still be unmuted if needed after using this function

Assumptions:

- Muted channels can be unmuted either individually or collectively without altering other settings

Error handling:

- If muting all channels fails, notify the user and retain previous audio settings

5.1.16 Unmute All Muted Channels

User story: As a SAR operator, I want the option to unmute all previously muted channels at once so that I can quickly restore all active communications. This feature allows me to resume monitoring multiple channels without individually unmuting each one. The system should enable a seamless return to the original audio setup after muting.

Description: The system must offer a feature to unmute all currently muted channels simultaneously, restoring all audio sources to their active states. This functionality helps operators resume listening to multiple channels at once without individually unmuting each. Unmuting all should be easy to execute and should restore all channels to the volume settings they had before being muted.
Acceptance criteria:

- Users can unmute all muted channels with a single action
- All previously muted channels are restored to their original volume and active states

Assumptions:

- Channels retain their original volume and spatial configurations when unmuted

Error handling:

- If unmuting all channels fails, notify the user and allow retrying the action

5.1.17 Save Channel Configurations

User story: As a SAR operator, I want to save my current audio channel configurations so that I can quickly restore my preferred settings later. This allows me to maintain consistency and efficiency during operations, even if I switch channels frequently. The system should allow me to save configurations easily without disrupting my workflow.

Description: The system should allow users to save their current audio channel settings, including volume levels, spatial locations, and muted channels, as a configuration profile. This feature enables operators to restore their preferred audio setups quickly, enhancing efficiency during shifts or when switching between tasks. Configurations should be saved persistently and accessible upon user request.

Acceptance criteria:

- Users can save the current configuration of all active audio channels
- Saved configurations include volume levels, spatial settings, and mute states
- Configurations can be named and stored for easy retrieval

Assumptions:

- Saved configurations are stored securely and can be accessed across user sessions

Error handling:

- If a configuration fails to save, notify the user and prompt them to retry or save under a different name

5.1.18 Load Channel Configurations

User story: As a SAR operator, I want to load previously saved audio channel configurations so that I can quickly switch to a preset setup during my operations. This lets me adjust the audio environment without manually setting each channel, saving time in critical situations. The system should make loading configurations straightforward and efficient.

Description: The system must allow users to load saved audio channel configurations, restoring volume levels, spatial settings, and mute states as saved in the profile. This feature ensures users can quickly access their preferred setups, streamlining operations and minimizing manual adjustments. The system must make loading configurations a simple process accessible within the audio interface.

Acceptance criteria:

- Users can load any saved configuration with a single action
- Loaded configurations restore all saved settings, including volume, spatial position, and mute status

Assumptions:

- Saved configurations can be accessed across sessions and are retained until deleted

Error handling:

- If a configuration fails to load, notify the user and suggest reloading or selecting an alternative configuration

5.1.19 Clear Current Configuration

User story: As a SAR operator, I want to clear my current audio channel configuration so that I can reset my setup quickly without individually adjusting each channel. This option helps me start fresh or remove unnecessary settings when switching tasks. The system should provide a simple way to clear all adjustments and return to default audio settings.

Description: The system must allow users to clear their current audio configuration, resetting all individual channel settings (volume, spatial position, and mute states) to default values. This feature enables operators to remove personalized adjustments quickly, creating a clean starting point for a new configuration. Clearing the configuration should not affect any saved presets and should be accessible with minimal steps.

Acceptance criteria:

- Users can clear the current configuration with a single action
- Clearing the configuration resets all channels to default volume, spatial position, and unselected unmuted state
- This action does not delete or alter any saved configuration.

Assumptions:

- The system has a default audio configuration for channels to revert to

Error handling:

- If clearing fails, notify the user and allow them to retry the action

5.1.20 Adjust Individual Channel Volume

User story: As a SAR operator, I want to adjust the volume of individual audio channels, so I can prioritize certain streams over others based on their relevance to my tasks. This ability to control volume helps me create an effective listening environment during critical operations. The system should support fine-tuned volume adjustments for each channel.

Description: The system must provide users with the ability to adjust the volumes for each channel individually. This helps operators prioritize audio sources by making critical streams more audible and reducing the volume of less important channels. The volume control should allow for smooth, granular adjustments to suit the user's needs.

Acceptance criteria:

- Users can increase or decrease the volume of each active channel independently
- Volume adjustments are responsive and reflect immediately in the audio output
- Adjustments do not affect saved configuration profiles unless explicitly saved

Assumptions:

- Volume levels are controlled through the user interface and adjust audio in real-time

Error handling:

- If a volume adjustment fails, notify the user and reset the channel volume to its last known setting

5.1.21 Adjust Virtual Speaker Location

User story: As a SAR operator, I want to adjust the virtual location of audio channels in my listening environment, so I can create a spatial representation of different sources. This

spatial arrangement enhances my ability to identify specific channels quickly. The system should provide flexible controls to set each channel's virtual position in 3D space.

Description: The system must allow users to adjust the virtual position of each audio channel within a 3D listening environment. This enables operators to organize channels spatially, helping them differentiate sources based on location. Adjustments to the virtual speaker location should be intuitive and update the audio positioning in real-time.

Acceptance criteria:

- Users can assign and adjust the virtual location of each audio channel in a 3D space with speakers and headphones
- Adjustments are maintained unless reset or saved in a configuration
- Changes in virtual position reflect immediately in the spatial audio output

Assumptions:

- Virtual location adjustments are intuitive and work with compatible audio hardware

Error handling:

- If a location adjustment fails, reset the channel to its previous position and notify the user

5.1.22 Log Errors, Access, and Activity

User story: As a SAR system administrator, I want to log system errors, user access events, and activity to maintain detailed records for monitoring and auditing. This log allows me to identify issues, analyze patterns, and enhance security. The system should ensure logs are accessible, secure, and structured for easy review.

Description: The system must log all relevant errors, user access events, and activity records in a secure, centralized location. This logging capability supports troubleshooting, security analysis, and compliance with operational standards. Logs should be organized by type and timestamp, with options for filtering and search to facilitate quick access.

Acceptance criteria:

- All errors, access events, and activities are recorded with timestamps, user identifiers, and action descriptions
- Logs are securely stored and be filtered by error, access, or activity type

Assumptions:

- Logs are routinely reviewed and used for security monitoring and auditing

Error handling:

- If logging fails, the system alerts the administrator and retries, with an option to store the log temporarily until the issue is resolved

5.1.23 Alerts for Audio Loss

User story: As a SAR operator, I want alerts for any significant audio loss on active channels so that I can take immediate corrective action if needed. These alerts help ensure that no important communications are missed during operations. The system should monitor audio integrity and alert me if audio loss exceeds an acceptable threshold.

Description: The system must monitor audio channels for significant packet loss or interruptions and alert users if audio loss reaches a critical threshold. This feature ensures that operators are promptly notified of any potential communication issues, allowing them to respond

accordingly. Alerts should be clear and actionable, with suggestions for troubleshooting where possible.
Acceptance criteria:

- Users receive alerts when audio loss exceed 5% on any active channel
- Alerts specify the affected channel and provide options to investigate or troubleshoot

Assumptions:

- The system has real-time monitoring for packet loss on each channel

Error handling:

- If audio loss monitoring fails, log the issue, notify the user, and attempt to re-establish monitoring as soon as possible

5.1.24 Enable Two-Way Communication

User story: As a SAR operator, I want the system to support two-way communication on select channels so I can respond directly to ongoing communications. This feature enables me to stay fully engaged with important channels without switching to a separate device. The system should make two-way communication accessible within the main audio interface.

Description: The system must support two-way communication on designated channels, allowing users to respond to audio streams directly from the platform. This feature facilitates real-time communication, enabling operators to engage in essential dialogues. Two-way communication should be seamlessly integrated into the audio management system.
Acceptance criteria:

- Users can initiate and maintain two-way communication on supported channels
- Audio quality remains clear during two-way exchanges

Assumptions:

- Two-way communication channels are clearly designated within the interface

Error handling:

- If two-way communication fails, notify the user and suggest switching to an alternative channel or device

5.2 Performance Requirements

Requirement	Goal Metric	Functional Area
5.2.1 HTTPS Connection Overhead	<= 1 s	Security
5.2.2 Authentication Response Time	<= 500 ms	User
5.2.3 Concurrent User Support	>= 10 users	User
5.2.4 Concurrent Audio Streams	>= 88 sources	Audio
5.2.5 Minimized Packet Loss	<= 5% loss	Audio
5.2.6 Low Latency Transmission	<= 500 ms	Audio
5.2.7 3D Audio Processing Latency	<= 200 ms	Audio
5.2.8 Mute/Unmute Response Time	<= 100 ms	Audio
5.2.9 Volume Adjustment Latency	<= 100 ms	Audio
5.2.10 Log Retrieval	<= 5 s	Logging

Table 2: Performance Requirements, Goal Metric, and Functional Area

5.2.1 HTTPS Connection Overhead

User story: As a system administrator, I want all HTTPS connections to establish quickly so that secure data transfers do not disrupt the user experience. This ensures that the systems meet high security standards while maintaining usability. Quick HTTPS connections are essential to protect data integrity without significant delays.

Description: The system must establish HTTPS connections within 1 second, ensuring secure, encrypted access without introducing significant delays. This requirement protects data integrity and aligns with security standards by securing all communications over HTTPS.

Acceptance criteria:

- HTTPS connections establish within 1 second
- All client-server communications use HTTPS with valid SSL/TLS certificates

Assumptions:

- HTTPS is enforced on all endpoints and includes automatic redirection from HTTP

5.2.2 Authentication Response Time

User story: As a user, I want authentication to be quick so that I can access the platform without experiencing significant delays. This enables me to log in efficiently, which is especially critical during time-sensitive operations. The system should respond to authentication requests within 500 ms to ensure a seamless experience.

Description: The system must respond to authentication requests within 500 ms to provide a fast login experience, which is essential for users needing quick access in high-stakes situations.

Acceptance criteria:

- Authentication completes within 500 ms
- Users are notified immediately if authentication fails

Assumptions:

- Authentication servers and network infrastructure are optimized for low-latency performance

5.2.3 Concurrent User Support

User story: As a system administrator, I want the system to support multiple concurrent users so that our team can collaborate without any performance issues. This allows for uninterrupted access to essential functions during SAR operations. The system should handle at least 10 concurrent users effectively.

Description: The system must handle more than 10 concurrent users, maintaining functionality and responsiveness under load. This requirement ensures the platform remains accessible and efficient for simultaneous users, supporting collaborative efforts in operations.

Acceptance criteria:

- The system supports at least 10 concurrent users without degraded performance or delayed responses
- All essential functions operate as expected with concurrent access

Assumptions:

- Network and server resources are scaled to handle multiple users without compromising performance

5.2.4 Concurrent Audio Streams

User story: As a SAR operator, I want the system to support 88 concurrent audio streams so that I can monitor multiple communication sources simultaneously. This capability is essential for managing real-time operations effectively. The system should maintain clear audio quality across all active streams.

Description: The system must support more than 88 concurrent audio streams with stable performance and minimal latency. This capability is essential for operations where operators need to monitor multiple audio channels in real time.

Acceptance criteria:

- The system can handle 88 concurrent audio streams without performance issues
- Audio quality and latency remain stable when multiple streams are active

Assumptions:

- System performance and audio hardware are optimized to support a high number of streams

5.2.5 Minimized Packet Loss

User story: As a SAR operator, I want minimal packet loss during audio transmissions to ensure clear and uninterrupted communication on all active channels. This prevents audio dropouts that could impact missions success. The system should maintain packet loss below 5%.

Description: The system must maintain packet loss below 5% for all audio streams, preserving audio quality and clarity. Low packet loss is critical for understanding communications in real-time environments, where lost audio data can lead to miscommunication.

Acceptance criteria:

- Packet loss for all audio transmission remains below 5% on average
- The system actively monitors packet loss and adjusts transmission parameters to minimize disruption

Assumptions:

- Network conditions are stable, and system resources are optimized for high-quality audio transmission

5.2.6 Low Latency Transmission

User story: As a SAR operator, I want audio transmissions to have low latency so that I receive communications in real time to make rapid decisions. This supports timely coordination in critical operations. The system should keep transmission latency under 500 ms to maintain synchronized communication.

Description: The system must maintain end-to-end latency of under 500 ms for audio transmissions, ensuring real-time communications remain synchronized. Low latency is essential for timely communication and delayed audio can hinder situational awareness.

Acceptance criteria:

- End-to-end transmission latency does not exceed 500 ms, including processing, network, and transmission delays
- All audio streams arrive in near real-time with synchronized timing across channels

Assumptions:

- The network is configured to prioritize low-latency audio transmission

5.2.7 3D Audio Processing Latency

User story: As a SAR operator, I want 3D audio processing to be fast so I can accurately differentiate and locate audio sources. This helps me quickly identify important communications in a multi-channel environment. The system should process 3D audio positioning adjustments within 200 ms.

Description: The system must process 3D spatial audio adjustments within 100 ms, allowing users to hear spatially distinct audio without delay. This latency limit supports real-time processing for 3D audio, enhancing the operator's ability to respond to spatial cues.

Acceptance criteria:

- 3D audio positioning updates within 200 ms of adjustments
- Spatial audio cues are distinguishable and correctly rendered without delay

Assumptions:

- Audio processing resources are sufficient to maintain spatial adjustments
- Audio hardware is capable of spatial sound placement

5.2.8 Mute/Unmute Response Time

User story: As a SAR operator, I want mute and unmute actions to occur instantly so that I can manage audio channels quickly. This helps me control audio based on priority or silence distractions during operations. The system should process mute/unmute commands within 100 ms.

Description: The system must process mute and unmute commands within 100 ms, providing real-time control over audio channels. This quick response time is essential for managing multiple channels efficiently, enabling operators to focus on high-priority audio.
Acceptance criteria:

- Mute and unmute actions respond within 100 ms of the user command
- Audio feedback immediately reflects the mute/unmute status

Assumptions:

- Audio control functions are optimized for low-latency response

5.2.9 Volume Adjustment Latency

User story: As a SAR operator, I want volume adjustments to apply instantly so that I can control audio levels based on each channel's importance. This lets me prioritize audio channels quickly in dynamic scenarios. The system should apply volume adjustments within 100 ms.

Description: The system must apply volume adjustments to individual channels within 100 ms, ensuring that users can control audio levels in real-time. Quick volume adjustments allow operators to prioritize specific channels by adjusting volume levels without delay.
Acceptance criteria:

- Volume adjustments respond within 100 ms of user input
- Audio feedback reflects the volume change immediately on the affected channel

Assumptions:

- The system infrastructure supports real-time volume adjustments with low latency

5.2.10 Log Retrieval

User story: As a system administrator, I want logs to load quickly so that I can review recent activities and access incidents without delay. This helps me identify and resolve issues efficiently when monitoring system activity. The system should retrieve logs within 5 seconds.

Description: The system must retrieve and display logs within 5 seconds, ensuring that administrators can review recent and historical activities without delay. Quick log retrieval supports timely monitoring and troubleshooting, allowing administrators to identify and address issues effectively.
Acceptance criteria:

- Logs load within 5 seconds of the request for both recent and historical data
- The system allows filtering by date and type to refine log retrieval

Assumptions:

- The logging database is optimized for quick access and retrieval

5.3 Environmental Requirements

Requirement	Priority Level	Functional Area
5.3.1 HTTPS access	High	Security
5.3.2 Use of U.S. Sourced Software	High	Security
5.3.3 Zero Trust Architecture	High	Security
5.3.4 AWS GovCloud Access	High	Security
5.3.5 NIST Compliance	Medium	Security
5.3.6 User-Friendly UI	Medium	Usability
5.3.7 Responsive Web Design	Low	Usability
5.3.8 Cross-Platform Compatibility	Low	Usability
5.3.9 $100 AWS Budget	High	Budget
5.3.10 $2500 Speaker Budget	High	Budget

Table 3: Environmental Requirements, Priority, and Functional Area

5.3.1 HTTPS Access

Compliance Objective: Ensure that all data in transit is encrypted to protect against interception and tampering, meeting security standards for secure communication in web environments.

Description: The system must enforce HTTPS for all network communications, ensuring that data transmitted between clients and servers is encrypted and secure. This requirement protects sensitive information from interception or tampering during transmission, aligning with best practices for secure web access. HTTPS access should be enabled across all interfaces, with HTTP requests automatically redirected to HTTPS to prevent accidental unencrypted access.

Implementation criteria:

- All external connections are secured with HTTPS, using TLS protocols for encryption
- HTTP requests are automatically redirected to HTTPS
- Valid SSL/TLS certificates are implemented and regularly updated

Assumptions:

- HTTPS is enforced at both application and network levels

5.3.2 Use of U.S. Sourced Software

Compliance Objective: Ensure compliance with national security guidelines that restrict software sources to trust U.S. venders, reducing potential risks associated with foreign-sourced software.

Description: The system environment must be built using software sourced exclusively from the U.S., ensuring compliance with national security guidelines and regulations. This requirement reduces the risk of vulnerabilities from foreign-sourced software and aligns with

operational security policies applicable to sensitive government operations. Any software used must have verifiable sourcing and certification that aligns with U.S. standards.
Implementation criteria:

- All software components and dependencies are verified as U.S.-sourced

Assumptions:

- The software team has access to sourcing information and verifiable supplier records

5.3.3 Zero Trust Architecture

Compliance Objective: Implement a security framework where no user or device is trusted by default, reducing the risk of both internal and external threats to meet high standards of access control.

Description: The system must be designed with a Zero Trust architecture, where no entity (internal or external) is inherently trusted, and all access requests are verified continuously. This security model enforces stringent identity verification, authorization, and continuous monitoring, reducing the risk of internal and external threats. Key aspects include least-privilege access, secure authentication mechanisms, and micro-segmentation to minimize attack surfaces
Implementation criteria:

- Access to resources is restricted based on the least privilege principle, and permissions are reviewed regularly
- Network micro-segmentation is implemented, limiting access within the system based on role and operational needs

Assumptions:

- The infrastructure supports Zero Trust Principles and allows for granular access control

5.3.4 AWS GovCloud Access

Compliance Objective: Ensure that all data processing and storage occur within a highly secure, U.S.-based cloud environment ensuring security requirements for sensitive data.

Description: The system must operate within AWS GovCloud to comply with U.S. government data residency, security, and compliance standards, as GovCloud offers enhanced security measures suitable for sensitive information handling. AWS GovCloud provided DoD compliance, meeting the high-security requirements of governmental operations. Using AWS GovCloud ensures that data remains within U.S. borders and that stringent access controls are enforced.
Implementation criteria:

- All infrastructure and services are hosted within AWS GovCloud
- Data storage and process comply with standards specific to GovCloud
- Access to AWS GovCloud is restricted to authorized personnel with verified credentials

Assumptions:

- The organization has a secure AWS GovCloud account with designated user roles

5.3.5 NIST Compliance

Compliance Objective: Adhere to NIST cybersecurity standards to protect data integrity, confidentiality, and availability, ensuring alignment with federal guidelines for data security.

Description: The system must adhere to NIST (National Institute of Standards and Technology) cybersecurity standards, ensuring that data protection, identity management, and security practices meet federal guidelines. Compliance with NIST helps mitigate risks related to data integrity, confidentiality, and availability, enhancing the overall security posture of the system. Key NIST standards should include protocols for incident response, data encryption, access control, and security audits.
Implementation criteria:

- Data encryption, access control, and logging policies align with NIST SP 800-53 and SP 800-171 standards
- Incident response protocols are in place with regular testing to meet NIST standards
- Regular security audits and assessments are conducted to verify ongoing compliance

Assumptions:

- NIST guidelines are accessible

5.3.6 User-Friendly UI

Compliance Objective: Ensure the system interface is accessible and intuitive, allowing users to perform tasks efficiently with minimal navigation.

Description: The user interface must be designed to facilitate ease of use, with intuitive navigation and a clear layout that requires minimal clicks to reach any primary function. This supports operational efficiency by reducing the time spent on interface navigation and focusing attention on core tasks.
Implementation criteria:

- All core functional areas (e.g., audio control, user settings, logs) are accessible within a maximum of 3 clicks from the dashboard
- Key actions should include labels, tooltips, or icons that indicate their purpose

Assumptions:

- Users have access to training resources or documentation, if needed, to familiarize themselves with advanced functionality

5.3.7 Responsive Web Design

Compliance Objective: Ensure that the application is accessible on various screen sizes and resolutions, maintaining usability and accessibility across devices.

Description: The system interface must adapt dynamically to different screen sizes, resolutions, and device types. This ensures that the application remains fully functional and visually consistent on multiple platforms, supporting access from diverse operational environments.
Implementation criteria:

- The interface layout adapts seamlessly to standard screen sizes
- All interactive elements remain accessible and usable on all devices
- No horizontal scrolling is required on any supported screen size

Assumptions:

- Users will access the system primarily on modern browsers

5.3.8 Cross-Platform Compatibility

Compliance Objective: Ensure that the application functions consistently across multiple operating systems and browsers, minimizing compatibility issues.

Description: The system must be compatible with major operating systems and web browsers, ensuring uniform functionality and appearance regardless of the user's platform. This supports widespread accessibility and reduces the risk of interruptions due to platform incompatibility.

Implementation criteria:

- The application operates without functional or visual discrepancies across Chrome, Firefox, Safari, and Edge
- The application supports current and recent versions of Windows, macOS, and Linux
- Testing and debugging cover cross-platform compatibility issues before deployment

Assumptions:

- Users are operating on the latest or penultimate versions of major browsers and operating systems

5.3.9 $100 AWS Budget

Compliance Objective: Maintain operational costs within the budget constraints set by the client to avoid unexpected expenses.

Description: The system must utilize AWS resources within a budget of $100, including storage, data transfer, and compute resources. This ensures sustainable operation by minimizing costs while maintaining essential functionality within AWS.

Implementation criteria:

- Resource usage is monitored regularly to avoid exceeding the budget
- AWS services are optimized for cost efficiency

Assumptions:

- Usage patterns are predictable, and the systems workload aligns with the allocated budget constraints

5.3.10 $2500 Speaker Budget

Compliance Objective: Ensure that the audio hardware setup remains within budget, as specified by the client, while meeting performance and quality standards.

Description: The budget for speaker hardware must not exceed $2500, balancing cost control with high-quality audio output to support the system's audio requirements. This limit includes all associated components and installation costs necessary for the audio output setup.

Implementation criteria:

- Audio hardware selected is cost-effective yet capable of spatial audio
- Quotes for equipment are obtained and verified to ensure total expenses do not exceed $2500

Assumptions:

- The speaker configuration is finalized and does not require replacement or upgrades, aligning with the allocated budget

6. Potential Risks

Risk Table			
Item	**Likelihood / 5**	**Severity / 5**	**Risk / 10**
6.1 Interfacing with Radio Simulator	2	4	6
6.2 Application Appearance	3	2	5
6.3 Hosting Service Downtime	1	5	6
6.4 Incompatibility of Selected Tools	1	4	5
6.5 Extra Noisy or Distorted Channel	3	4	7
6.6 Critical Library Disappears	1	5	6
6.7 Password Corruption	2	5	7
6.8 Fails to detect User Leaving	2	2	4
6.9 Weak Password is Used	2	4	6
6.10 Unauthorized Access	2	3	6
6.11 Accidental Account Deletion	2	4	6

Table 4: Potential Risks of the System rated on likelihood and severity

6.1 Interfacing with Radio Simulator (Moderate Severity - 6)

Main Risk: The radio signals being sent to the web server could be affected by external factors, resulting in distorted or missing playback. For the radio simulator to work properly it needs to be both efficient and reliable to ensure the project's success. The potential risk is that the radio simulator can have distortion that causes audio to not come in clearly or can have issues when playing from the cloud.

Mitigation: The team intends to leverage Amazon Web Services tools to use subnetting to ensure the packets from the radio simulator are being sent to the web application in a secure and stable manner.

6.2 Application Appearance (Moderate Severity - 5)

Main Risk: For the application appearance, it is important to ensure client satisfaction. The user interface must align with their clients wants and must incorporate a clean set of audio stream controls that change color to represent when an audio is being played. It must also contain the ability to easily change the position of audio in 3D space. A potentiation challenge that can be faced is getting the web application’s look and feel down to the exact client needs.

Mitigation: If such issues arise, StratoSplit will communicate with General Dynamics and discuss other possible solutions that will meet their needs. While maintaining an ideal audio

dashboard, it is much more crucial to get the features of the dashboard rather than the look and feel of the dashboard.

6.3 Hosting Service Downtime (Moderate Severity - 6)

Main Risk: It is possible that AWS could go down at some point, which would also make our application unusable. However, AWS is a very reliable service, being used by many large companies around the world.

Mitigation: One way to possibly mitigate this risk is to split our application between multiple remote servers, so that in the event one is lost, the app can continue to function. However, our client has specified which servers we will be able to use, so we may not be given the option to spread our app across multiple servers, making us vulnerable to one data center going offline.

6.4 Incompatibility of Selected Tools (Moderate Severity - 5)

Main Risk: Through our research of tech feasibility, we have decided upon many different tools to complete our project. Many of these tools, however, are still being supported and developed. This is good, from a security standpoint, to improve how they work. However, a future update could also result in some functionality being changed or removed, breaking the app in the process.

Mitigation: This risk can be mitigated through addition of automated testing, and the thorough research and testing of the application. By creating more modular code, we can help the process of transferring technologies easier in the future, if that becomes necessary.

6.5 Extra Noisy or Distorted Channel (High Severity - 7)

Main Risk: If one channel is constantly giving feedback, or is otherwise distorted, then it could make other channels difficult to listen to. This could impact the effectiveness of the radio operators, and delay communication.

Mitigation: To mitigate this risk, it would be advantageous to be able to mute that particular channel, while listening to the rest, or at least lower its volume.

6.6 Critical Library Disappears (Moderate Severity - 6)

Main Risk: There are several different libraries that our application relies on. If any of these libraries are removed from a machine that the application is running on, then we risk not being able to run the application at wall.

Mitigation: This can be mitigated by loading your libraries locally onto the project. With library information remaining local, any changes upstream won't affect us. However, this will increase the size of the application.

6.7 Password Corruption (High Severity - 7)

Main Risk: There are many ways for a password to become lost in transmission. The two main ways are that the hash is being altered to something the hackers can abuse. The second main way is the salt getting corrupted or leaked. The salt ensures the security of the password. It ensures that our database cannot be cracked through methods such as rainbow table attacks. This is of high severity as if our passwords get corrupted then our users will not be able to access the service.

Mitigation: To mitigate this risk, it would be of best interest to confirm the hash after it is inserted into the database to ensure that no data was lost in transmission.

6.8 Fails to detect User Leaving (Low Severity - 4)

Main Risk: If the software fails to detect a user leaving the session, then resources may be unnecessarily taken up within the host. A container running for an unnecessary amount of time may eat into our budget, as we would be paying for more uptime with AWS. This is unlikely to happen with well written code for detecting if the user is still present, and would only likely be noticeable if it occurred many times, leaving many containers running.

Mitigation: There will be a system in place for detecting when a user is still actively using a container, and if the user disconnects, the container will be automatically shut down.

6.9 Weak Password is Used (Moderate Severity - 6)

Main Risk: If the user is able to set a weak password for themselves, it may open the system up to being breached by simple attacks that try many common passwords. This could compromise the security of all conversations that are being held on communications accessible by the role that has been compromised.

Mitigation: There will be a system in place that will guarantee the minimum security requirements of a password. These could include special characters, numbers, and a combination of lowercase and uppercase letters.

6.10 Unauthorized Access (Moderate Severity - 6)

Main Risk: As part of the Zero Trust methodology we will be implementing is to never trust the user. This means that at any moment a threat actor can be impersonating the user. This can lead to unauthorized password changes, listening in on radio frequencies that are forbidden and much more.

Mitigation: To mitigate unauthorized access through a legitimate user's account, we will implement reauthentication anytime the user plans on changing something about their account. This means making the user re-login if they want to change their password or change any of their settings. This ensures that only the user owner is making changes to their account and nobody else.

6.11 Accidental Account Deletion (Moderate Severity - 6)

Main Risk: If an action is taken to delete an account, when a user did not intend for their account to be deleted, then the user will lose all of their access. Even if their access is

reinstated, all of their preset settings would be lost, resulting in lost time and reduced efficiency until the settings can be manually restored.

Mitigation: This could be mitigated by making the interface to delete an account clear, and to have a confirmation pop up before the actual account deletion is executed.

7. Project Plan

Name	Start Date	End date	Days	September	Octover	November	December	January	February	March	April	May
Documentation	9/2/2024	5/9/2025	180									
Audio Generator	10/14/2024	12/31/2024	57									
Database	11/8/2024	1/30/2025	60									
User Interface	11/8/2024	3/31/2025	102									
Spatial Audio	11/8/2024	2/27/2025	80									
Testing	12/2/2024	4/29/2025	107									
Zero Trust	12/16/2024	4/16/2025	88									

Figure 4: Gantt Chart of Project Timeline

Our initial priority has been proving the feasibility of the project to the clients. It is important to provide the ability to simulate an audio generator and being able to multicast that audio from the generator to a cloud based web application with spatial audio. From there, our next biggest priority is documentation. As only a prototype and proof of concept is being created. It is important that when this project gets passed off, documentation is up to spec showing how the project was developed and how the project works.

Our primary priority is ensuring communication and documentation. These two ensure that the clients are getting the project they desire. Through documentation, this allows the clients to see where we are in the development process and gives them the opportunity to ask better questions and to provide better feedback.

Second, our priority is to get a feasible stream generator that will send multicast audio from an external source into our web application. The plan for this is to use FFMPEG to generate the audio streams and multicast it. This will create the foundation of the project. Allowing the project to be built off of its core functionality will make it easy to scale to more audio streams and require less back tracking.

Next, the user interface needs to be developed according to the provided requirements from our clients. The plan is to create an easy to use interface that allows the operators to create custom configurations that enable them to quickly save and load different dashboards that suit their needs for the day. Next we will create a simple and intuitive 3D audio board that allows the operators to place any stream where they would like whether it be above them, behind them, or next to them.

Testing is a big part of ensuring the stability of the code. The testing suite that will be used during the phase is called Jest. It allows for simple test cases to be created and allows the code to be scanned for code coverage. This allows most major bugs to be removed from the code and ensures stability in the program. Tests will be created throughout the entire development process to ensure that no lines of code are not being tested.

From there, we will implement our Zero Trust methodology, such as patching any security vulnerabilities being open ports, backdoors, and setting up a secure firewall that only allows outbound traffic from the web application. Next, we will implement Two Factor Authentication to create a second layer of security if a password gets leaked. This will ensure that only our users can access the application. Lastly, we will ask for the user to reauthenticate

any time they want to make a change to their account to ensure Zero Trust is the core of the app. This makes sure that the user is who they say they are and not a threat actor.

8. Conclusions

The StratoSplit project, developed in collaboration with General Dynamics and the U.S. Coast Guard is designed to enhance audio communication systems for operators. With the existing communication interface not being modern, We will create a secure and efficient platform that integrates advanced features such as real-time audio monitoring, spatial 3D audio, and robust user management. Hosted on AWS GovCloud and designed with a Zero Trust architecture so the system ensures high security. With different types of requirements being used to solve different problems. Such as functional requirements, which focus on specific features and capabilities of the system. performance requirements which centers on metrics and thresholds that define the system's operational standards. And Environmental requirements which focus on compliance and infrastructure needs, including security protocols. With all of these requirements satisfied we will create a product that is easy to use, keeps zero trust security, and is quick to run.

With potential risks such as radio simulator interfacing challenges, hosting service downtimes, and tool incompatibilities. These risks each have associated severities which can range from small issues with stream breakage to high issues that can provide security vulnerabilities. Addressing these issues we came up with mitigation strategies to ensure reliability. These strategies, such as AWS subnetting, automated testing, and fallback mechanisms strengthen the system's resilience. After implementing these mitigation strategies the chance of website downtime or security breaches will be reduced. With these requirements we will be able to create a product that the client will be satisfied with while reducing risks. Following our plan above we will be able to complete these requirements in the specified timeframe.