Software Requirements Specification for Audio360

Team #6, Six Sense
Omar Alam
Sathurshan Arulmohan
Nirmal Chaudhari
Kalp Shah
Jay Sharma

November 14, 2025

Contents

List of Figures

List of Tables

Revision History

Date	Version	Notes
2025-10-06	1.0	Initial Write-up
2025-10-22	2.0	Technical terms to glossary
2025-10-22	2.1	Context diagram included
2025-10-26	2.2	Update sampling rate requirement
2025-11-05	3.0	Addressing TA comments
2025-11-05	3.1	Symbolic Constants section
2025-11-05	3.2	Providing additional rationale and clarification for functional and non-functional requirements.
2025-11-05	3.3	Verification and acceptabce criteria abstracted away from defining specific metrics.
2025-11-07	4.0	Added requirements for software layer abstraction (FR4.5)
2025-11-12	4.1	Added a symbols, abbreviations, and acronyms table

1 Goal

1.1 G.1 Context and overall objective

With around 4 million Canadians affected by hearing loss [1], there is a significant need for assistive technologies that can improve situational awareness and safety. Many safety cues and general sound alerts such as the sound of a car approaching, a kettle whistling, or a phone ringing may be missed, leading to increased risk of injury and miscommunication.
Many existing solutions focus on speech transcription, but lack the ability to provide directional information about sound sources or classify non-speech sounds. This project aims to address this gap by developing an assistive device that provides real-time visual indications of sound source locations and classifications.
The objective of this project is to develop an assistive device that aids individuals who are deaf or hard of hearing by providing real-time visual indications of sound source locations and classifications (ex. 'car on your left').
Some of the high-level goals of the project are:

Goal 1. Capture real-time audio data from a microphone array with synchronized sampling to enable accurate situational analysis of sound sources.

Goal 2. Analyze captured audio to determine the direction of arrival (DoA) of sound sources with minimal error and with minimal latency nearing real-time.

Goal 3. Analyze captured audio to classify the sound sources with their English label (ex. 'car', 'phone', 'kettle', 'alarm', 'speech').

Goal 4. Display audio classification and sound direction on smart glasses in real-time without obstructing the user's field of view.

Goal 5. Provide a user-friendly interaction with the smart glasses, allowing the user to easily set up, use, and understand visual indicators.

Goal 6. Ensure that the system is comfortable to wear for extended periods of time, with minimal discomfort or fatigue.

Goal 7. Ensure that the system provides additional safety aids to the user by providing information to the user to make cautious decisions. Also ensure user privacy and security when using the system to avoid external malicious actors.

1.2 G.2 Current situation

Currently, individuals who are deaf or hard of hearing face significant challenges in maintaining situational awareness due to missed audio cues. Existing assistive technologies address some aspects of this problem, but leave critical gaps:

- **Smart glasses with transcription capabilities:** Some devices can listen to live human audio and transcribe it to text (multilingual) in real-time, displaying the transcript on a smartphone display. However, these solutions focus solely on speech transcription and do not provide directional information about sound sources or classify non-speech sounds.
- **Hearing aids:** Traditional hearing aids amplify ambient sounds to improve awareness of audio sources at various volumes [2]. While this helps individuals with partial hearing loss, it does not assist those who are profoundly deaf, nor does it provide visual cues about sound direction or classification.
- **Notification systems:** Some home automation systems can send visual alerts (e.g., flashing lights) when specific sounds are detected, such as doorbells or smoke alarms. However, these systems are limited to fixed locations and predetermined sound types, lacking portability and real-time directional awareness.

The current solutions fail to address the critical need for real-time, portable, directional awareness of environmental sounds, leaving individuals vulnerable to missing important safety cues such as approaching vehicles, warning beeps from machinery, or emergency alerts.

1.3 G.3 Expected benefits

The proposed system will deliver significant improvements to the daily lives and safety of individuals who are deaf or hard of hearing:

- **Real-time spatial awareness:** Enable identification of sound source locations on a 2D plane in real-time, allowing users to quickly orient themselves toward important sounds such as someone calling their name, an approaching vehicle, or an emergency alarm.
- **Enhanced safety:** Reduce the risk of injury by alerting users to critical safety cues that are typically communicated through sound, such as warning beeps from forklifts, tea kettles whistling, car engines or emergency sirens (from emergency vehicles) approaching from behind.
- **Improved situational awareness:** Provide continuous awareness of the acoustic environment without requiring the user to constantly scan their surroundings, reducing cognitive load and enabling more natural interactions with their environment.
- **Sound classification:** Differentiate between various types of sounds (e.g., speech, alarms, vehicles, household appliances) to help users prioritize their attention and responses appropriately.
- **Reduced frustration and miscommunication:** Minimize instances of missed phone calls, doorbell rings, or verbal attempts to gain the user's attention, leading to smoother social interactions and reduced social isolation.

- **Portable and wearable solution:** Unlike fixed home automation systems, the smart glasses form factor provides continuous protection and awareness regardless of location, whether at home, work, or in public spaces.

- **Independence and confidence:** Empower users to navigate their environment more independently without relying on others to alert them to important sounds, fostering greater autonomy in daily activities.

1.4 G.4 Functionality overview

The system will provide the following principal functions:

- **Real-time audio capture:** Continuously capture audio signals from a synchronized microphone array mounted on smart glasses, ensuring precise temporal alignment for accurate spatial analysis.

- **Direction of arrival (DoA) estimation:** Process captured audio to determine the angular direction of sound sources on a 2D plane relative to the user's position, with a target accuracy of θ_e for single sound sources.

- **Sound source classification:** Analyze audio characteristics to classify detected sounds into meaningful categories (e.g., speech, vehicle sounds, alarms, household appliances) using audio fingerprinting techniques, with a target accuracy of at least $A_{classification}\%$.

- **Visual feedback generation:** Generate intuitive visual representations of detected sound sources, including their direction and classification, displayed on the smart glasses interface with minimal latency ($\leq t_{lat}$ second).

- **Multi-source handling:** Detect and track multiple simultaneous sound sources when feasible, prioritizing the most relevant or critical sounds based on classification and proximity.

- **Real-time processing:** Execute all signal processing, direction estimation and classification algorithms real-time, with consistent performance and low latency.

- **Noise cancellation or audio filtering:** The system will modify or filter sound waveforms received from the microphones to reduce background noise and improve the accuracy of the DoA estimation and classification algorithms.

1.5 G.5 High-level usage scenarios

The following scenarios illustrate fundamental usage paths through the system:

1.5.1 Scenario 1: Pedestrian crossing detection

A user is walking in an urban environment and approaches a street intersection. As they prepare to cross, a car approaches from their left side. The system detects the engine sound, estimates its direction (e.g., 90° to the left), classifies it as a vehicle, and displays a visual indicator on the smart glasses showing the direction and classification. The user recognizes the alert and waits for the vehicle to pass before crossing safely.

1.5.2 Scenario 2: Kitchen safety alert

A user is cooking in their kitchen when a tea kettle on the stove begins to whistle. The system captures the high-pitched sound through the microphone array, determines that it is coming from behind and to the right (e.g., 135°), classifies it as a kettle or alarm sound, and displays a directional indicator. The user turns toward the alert and removes the kettle from heat, preventing a potential hazard.

1.5.3 Scenario 3: Social interaction

A user is in a crowded room when someone calls their name from across the space. The system detects the speech sound, estimates the direction (e.g., 30° to the right), classifies it as speech or a human voice, and displays the information on the glasses. The user turns in the indicated direction to make eye contact and engage in conversation, reducing social friction and missed interactions.

1.5.4 Scenario 4: Workplace awareness

A user is working in an industrial setting when a forklift begins reversing nearby, emitting a warning beep. The system detects the beeping pattern, determines its direction (e.g., directly behind at 180°), classifies it as a warning signal, and alerts the user with a prominent visual indicator. The user steps aside to maintain a safe distance from the moving equipment.

1.6 G.6 Limitations and exclusions

The following aspects are explicitly outside the scope of this project:

- **Autonomous danger assessment:** The system will not independently evaluate whether a detected sound represents an immediate danger or automatically alert the user of hazardous situations. It will present directional and classification information, leaving interpretation and response decisions to the user.

- **Augmented reality overlay:** The system will not provide full augmented reality capabilities with spatial overlays showing sound locations directly mapped onto the user's field of view. Visual feedback will be presented through a simpler display interface on the smart glasses.

- **User response monitoring:** The system will not track whether the user has noticed, acknowledged, or responded to presented alerts. There is no feedback loop to ensure user reaction or to escalate notifications.
- **3D spatial localization:** Direction estimation will be constrained to a 2D horizontal plane around the user. Elevation angle determination (above or below the user's head level) is excluded from the core functionality.
- **Sound source distance estimation:** While direction will be provided, the system will not attempt to estimate the absolute distance to sound sources.
- **Continuous recording or data storage:** The system will not record or store audio data beyond what is necessary for real-time processing. No historical logs of detected sounds will be maintained.
- **Network connectivity:** All processing will occur locally on the embedded hardware. The system will not require internet connectivity or cloud-based services for core functionality.

1.7 G.7 Stakeholders and requirements sources

1.7.1 Primary stakeholders

- **Individuals who are deaf or hard of hearing:** The primary end-users of the system, who will directly benefit from improved situational awareness and safety. This group is quite large in population, with approximately 4 million people who experience hearing loss in Canada alone (1 in 10) [1]. The McMaster Sign Language club is a group at McMaster University that falls in this stakeholder group.

1.7.2 Secondary stakeholders

- **Family members and caregivers:** Individuals who support people with hearing loss and will benefit from improved communication and reduced safety concerns.
- **Employers and workplace safety officers:** Organizations that employ individuals with hearing loss and are responsible for maintaining safe working environments.
- **Accessibility advocates and organizations:** Groups focused on improving quality of life and independence for individuals with disabilities.
- **Healthcare providers and audiologists:** Professionals who may recommend or integrate such assistive technologies into patient care plans.
- **Future developers and researchers:** The broader engineering and scientific community who may build upon this work or apply similar techniques to related problems.

1.7.3 Requirements sources

- **Academic literature:** Research on hearing loss impact, assistive technologies, direction of arrival algorithms, and audio classification techniques.
- **Domain experts:** Consultation with domain experts, such as Dr. Mohrenschildt, for technical feasibility and requirements validation.
- **Existing assistive technologies:** Analysis of current solutions such as hearing aids, transcription glasses, and home alert systems to identify gaps and opportunities.
- **Hardware and software documentation:** Technical specifications for microcontroller, source code libraries, smart glasses hardware, and microphone array components.
- **Standards and best practices:** IEEE standards for embedded systems, accessibility guidelines, and real-time system design principles.
- **Proof of concept testing:** Empirical results from prototyping and laboratory testing to validate technical approaches and refine requirements.

2 Environment

2.1 E.1 Glossary

- **ASL:** American Sign Language used by deaf and hard of hearing individuals to communicate using hand gestures and facial expressions.
- **Euclidean Quadrants:** The four regions of a 2D Cartesian plane, divided by the x and y axes, numbered 1 to 4 in a counterclockwise direction starting from the top-right. The ranges of the quadrants are (0°–90°), (90°–180°), (180°–270°), and (270°–360°) degrees.
- **Frequency Domain:** A way to represent periodic or sinusoidal signals based on their frequency content, showing how much of the signal lies within each given frequency range. Frequency units is Hz.
- **Hardware Acceleration:** The use of specialized hardware modules within a microcontroller to perform specific computational tasks that are more computationally efficient than running equaicalent software on the CPU. For example, signal processing and vector operations.
- **Microcontroller:** Compact embedded hardware system with a CPU, memory, input/output peripherals, designed to process real-time data directly on the device.

- **Microphone:** Input sensor to a device that converts audio soundwaves to digital bytestream.
- **Microphone Array:** A collection of microphones that are synchronized to capture audio from the environment to create a single multi-channel audio signal.
- **Normal Operating Condition:** The state in which the system, including hardware and softwre, operate within expected environmental and usage parameters. This includes audio input sampled at the correct sample rate, input audio signals are within the human hearing most audible range (250 Hz – 8 kHz) [3], and audio input levels below 90 dB.
- **Real Time:** Computational processing of the tasks within the system that adhere to specific timing constraints. All processing is guranteed to complete before its deadline.
- **Spectral Leakage:** The spreading of a signal's energy across multiple frequency bins in a frequency spectrum due to finite time windowing. Essentially, noise that is caused by sharp signal cutting when processing audio signals as windows.

2.1.1 E.1.1 Symbols, Abbreviations, and Acronyms

Symbol	Value	Description
θ_e	$\pm 45°$	Angle error threshold. 45 threshold is selected to indicate the sound source direction within half-quadrant resolution.
$A_{classification}$	90%	Classification accuracy. 90% is chosen as it an achievable metric with the time given while providing high confidence in audio classification for stakeholder's use cases.
t_{lat}	$1s$	Latency time of visual feedback. 1 second was chosen arbitrarily chosen. It will be tuned to better fit stakeholder's reaction time in critical situations. This will be tuned during the validation stage.

Table 1: Table of symbolic constants used in Audio360.

symbol	description
AC	Anticipated Change
Audio360	360 Audio analysis system on smart glasses
DAG	Directed Acyclic Graph
FFT	Fast Fourier Transform
FR	Functional Requirement

M	Module
MG	Module Guide
MIS	Module Interface Specification
NFR	Non-Functional Requirement
OS	Operating System
R	Requirement
SC	Scientific Computing
SPI	Serial Peripheral Interface
SRS	Software Requirements Specification
T	Test
UC	Unlikely Change
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
VnV	Verification and Validation

Table 2: Table of abbreviations and acronyms used in Audio360.

2.2 E.2 Components

Below are the components that will affect the system.

1. Physical Environment
   - Ambient noise sources that the system needs to detect and classify (ex. car passing by, people talking, dog barking)
   - Sounds that the classification module does not have knowledge about.
   - Weather conditions that may distort the microphone input (ex. wind, rain, damping from humidity)
   - Room acoustics that will differ based on the space the user is in (ex. inside a building vs. outside)
2. User interaction
   - The system will be mounted on the user's head, which is considered an external component affected by the system.
   - The user's head movement will also affect the system, since it may affect the microphone orientation and sound localization.
3. Legal Components
   - Privacy standards related to continuous audio capture in public spaces.

2.3 E.3 Constraints

1. Legal constraints

   - Criminal Code - Private Communications (Section 184) [4]. Under this provision, it's an offense to listen, record or acquire a private communication without consent. Although audio retrieved from the environment won't be stored, this provision may limit where the user is able to use this device. Most likely, they may only be able to use this device in public places.
   - Privacy & Personal Information Laws (PIPEDA) [5]. Under this act, if data is stored, transmitted or analyzed, this act triggers obligations that must follow under the privacy law. These obligations include consent, secure handling and limited handling. These obligations limit what can be done with the audio sources.

2. User interaction constraints

   - Since the device must fit on the user's head, the hardware is limited in size, resulting in chips with slower CPUs and less onboard memory which limits onboard processing power.
   - Since the user will be using this system throughout the day, the system must operate on a portable power source.

3. Funding constraints

   - With a maximum budget of $500, the hardware capabilities are limited. (ex. CPU speed, memory space) As a result, the system functionality will be impacted based on the limitations of the hardware. As a result, hardware acceleration techniques shall be used (FR3.4).

4. Physical environment constraints

   - The system may be used in outdoor environment. Thus the system shall be functional in all types of weather.

2.4 E.4 Assumptions

1. Physical environment assumptions

   - The environment will not exceed normal outdoor temperature ranges (0-35ºC).
   - The device will be used in moderate environments, not in places with overwhelming amount of background noises (ex. concerts, factories).
   - The environment will always be dry, eliminating the need for the system to be waterproof.

2. User interaction Assumptions

- The user's head orientation remains stable, with minimal rapid head motion, when the device is collecting audio data from the environment.

3. Legal Assumptions

    - The user will always use the device in public, where there is no expectation of privacy.
    - No audio data is permanently stored, transmitted or analyzed outside of the system.

2.5 E.5 Effects

Since the system is mainly just processing input from the environment, it won't largely affect many of the environmental components mentioned in E.4. The main environmental component that the system will affect are the user's interacting with the device. Listed below are two ways the system will affect users.

- The system will generate a small amount of heat, which may make the user feel more hot than usual.
- The system will raise awareness of the user's surroudings, reducing the likelihood of accidents when in public.

2.6 E.6 Invariants

1. User Safety & Comfort

    - The system must not significantly reduce the user's field of vision.
    - The system must not emit heat at levels that cause harm to the user.

2. Legal Boundaries

    - The system must maintain the privacy integrity of sounds it collects. This means, it doesn't permanently store any sounds collected from the environment.

2.7 E.7 Context

Figure 1 is a context diagram that provides a high-level overview of the Audio360 system and its interactions with external entities. Audio360 receives audio input from a microphone array, processes the data to analysis the user's environment, and outputs visualizations to a display.

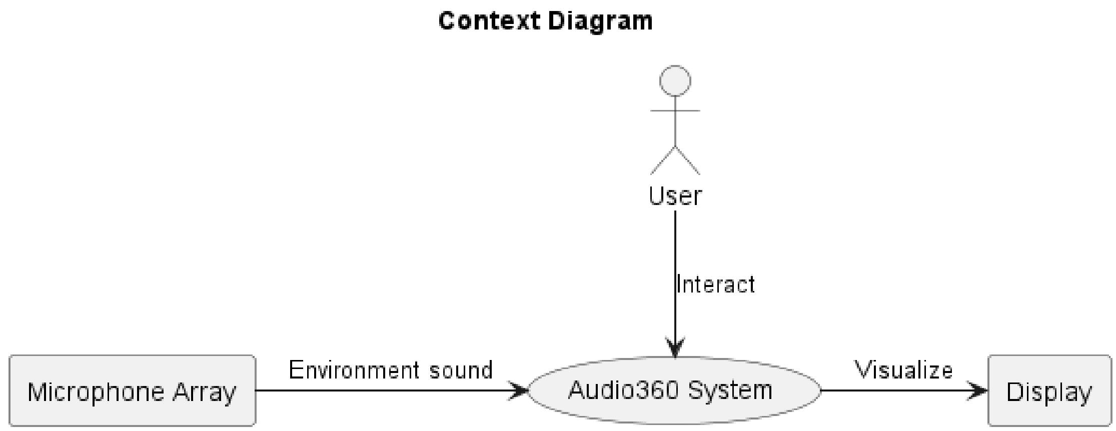

Figure 1: Context diagram on how the system interacts with the environment.

3 System

3.1 S.1 Components

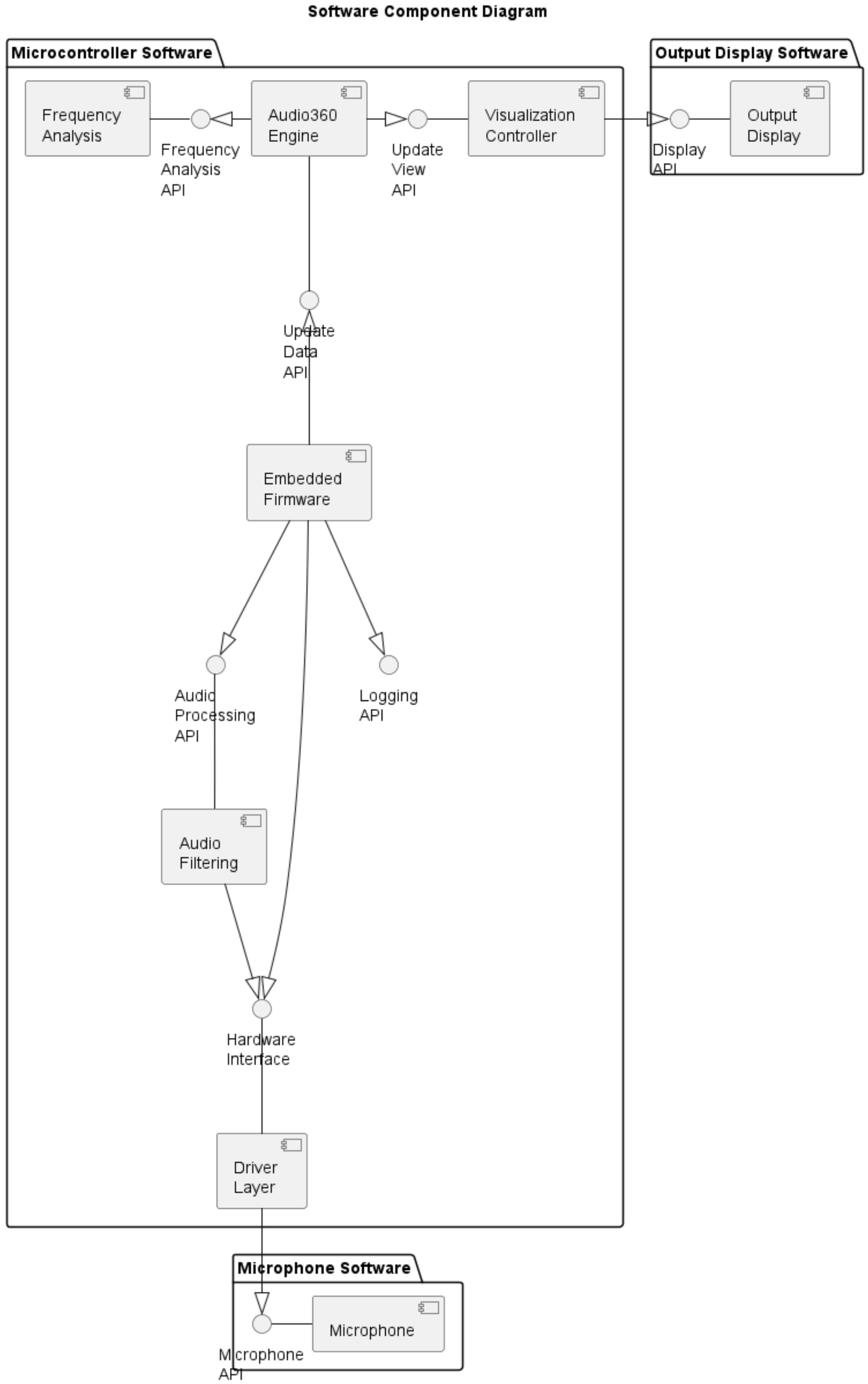

Figure 2: Software component diagram.

3.1.1 Software Components

- **Embedded Firmware:** Main component responsible for managing all embedded software on the Microcontroller.
- **Driver Layer:** Component responsible for providing drivers on the microcontroller. It provides interfaces for high level applications to interact with hardware components external to the microcontroller
- **Audio Filtering:** Component responsible for processing raw audio signals in real-time sent by the microphones.
- **Audio360 Engine:** Main component responsible for running Audio360 features. It serves as the primary interface and controller for the Audio360 system. The component is designed to be hardware agnostic, allowing seamless integration with any firmware running on other hardware.
- **Frequency Analysis:** Component responsible for analyzing properties of processed frequency signals. Sub components of this includes audio classification and direction of arrival estimation.
- **Visualization Controller:** Component reposonsible for creating and sending visualization output to the output display.

3.1.2 Hardware Components

- **Microphone:** Component responsible for collecting audio signals from the environment.
- **Output display:** Component responsible for displaying visuals to the user.
- **Microcontroller:** Component responsible for executing the main software and interfacing with input and output peripheral devices.

3.2 S.2 Functionality

3.2.1 Embedded Firmware

Functional Requirements

- **FR1.1:** The firmware shall schedule tasks based on its priority. Priority ensures that critical tasks are ran and completed before its deadline, avoiding functionality loss.
- **FR1.2:** The firmware shall process and synchronize audio signals from all microphones connected to the microcontroller. Audio synchronization is required to determine direction of audio source as it depends on frequency phase shift.

- **FR1.3:** The firmware shall handle memory errors to prevent the microcontroller from crashing. This ensures that the system will never enter an unrecoverable state.
- **FR1.4:** The system shall perform continuous diagnostics on all hardware components to monitor hardware errors in real-time. This ensures the system can react to failures as soon as they occur.

Non-Functional Requirements

- **NFR1.1:** The firmware shall process audio signals in real-time with monotonic frame sequences. Earliest frames shall have higher priority. For example, if frame F_t arrives at time t, then the priority, $P(F_t)$, must satisfy $P(F_t) > P(F_{t+1})$.
- **NFR1.2:** The firmware shall operate faster than the connected microphone sample rate 16 kHz, referring to NFR7.1. This ensures that all input data from the microphone can be processed.
- **NFR1.3:** The firmware shall permanently discard microphone audio data that is greater than the sampling window size. This ensures that audio data is retained only temporarily for processing, supporting Goal 7, protecting user privacy.

3.2.2 Driver Layer

Functional Requirements

- **FR2.1:** The driver layer shall provide an interface for higher level software to interact with the hardware on the microcontroller.
- **FR2.2:** The driver layer shall process hardware interface requests based on system permissions of the requester.
- **FR2.3:** The driver layer shall return error codes upon return to higher-level software to support error handling.
- **FR2.4:** The driver shall maintain data integrity on memory slots that is actively being used.

Non-Functional Requirements

- **NFR2.1:** The driver shall immediately propagate any errors to the firmware layer for proper error handling. This is to ensure that the processor can take the neccessary steps to recover out of the error.

3.2.3 Audio Filtering

Functional Requirements

- **FR3.1:** The audio filtering component shall convert digital audio waveform to frequency domain.
- **FR3.2:** The audio filtering component shall normalize the amplitude of incoming and outgoing signals. This is to simplify analysis when comparing different signals.
- **FR3.3:** The audio filtering component shall filter incoming audio signals to reduce frequency spectral leakage.
- **FR3.4:** The audio filtering component shall use available hardware acceleration for efficient computations. The system needs to run as fast as possible with limited compute power, and using hardware acceleration is the most efficient method available.
- **FR3.5:** The audio filtering component shall detect and flag audio anomalies such as clipping, lost signal and silence. This enables identification of microphone or signal path faults.

Non-Functional Requirements

- **NFR3.1:** The audio filtering component shall represent audio signals in the frequency domain with less than 10% error from the true value. This is maximum tolerable error to process audio for classification and directional analysis.
- **NFR3.2:** The audio filtering component shall be scalable to handle different input signal sizes without missing timing constraints defined in NFR1.2 for input signal that are powers of 2 and size up to 4096 samples. Powers of 2 since hardware acceleration, from FR3.4, for FFT are most optimized for this [6]. 4096 samples is the maximum standard size.
- **NFR3.3:** The audio filtering component shall have an audio processing success rate end to end of atleast 80%. Due to variations in input audio quality, some errors are expected. While state of the art audio classification systems can reach up to 90% accuracy, they typically require significantly more computational resources that this project does not have access to.

3.2.4 Audio360 Engine

Functional Requirements

- **FR4.1:** The Audio360 Engine shall retrieve data, such as frequency domain and errors, from the microcontroller via the driver layer for further processing.
- **FR4.2:** The Audio360 Engine shall notify dependent components when new frequency data is available. The latest data shall always be used.

- **FR4.3:** The Audio360 Engine shall control the flow of processed audio data from audio frequency analysis to visualization. Only processed data should be visualized to the user.
- **FR4.4:** The Audio360 Engine shall disable audio classification and directional analysis features until microphone faults addressed in FR3.5 are resolved. This prevents the generation of unreliable or unsafe outputs.
- **FR4.5:** The Audio360 Engine shall be portable. That is the system can be deployed on different hardware. This is to ensure that the there is flexibility to upgrade hardware to meet higher processing demands when needed.

Non-Functional Requirements

- **NFR4.1:** The Audio360 Engine shall poll data without conflicting with ongoing microcontroller memory writes, ensuring no data loss.
- **NFR4.2:** The Audio360 Engine shall retrieve frequency data within the timing constraints of the embedded firmware defined in NFR1.2.
- **NFR4.3:** The Audio360 Engine shall permanently discard microphone audio data immediately after completion of audio analysis. This adheres to Goal 7, protecting user privacy.

3.2.5 Frequency Analysis

Functional Requirements

- **FR5.1:** The frequency analysis component shall classify sound sources based on features extracted from the frequency domain representation of the audio input signal.
- **FR5.2:** The frequency analysis component shall estimate the direction of arrival of the audio source using frequency domain representation of the audio input signal.
- **FR5.3:** The frequency analysis component shall represent the direction of arrival as an angle (θ) in radians. θ is measured relative to the forward axis of the glasses frame to the sound source, Figure 3.
- **FR5.4:** The frequency analysis component shall notify users when a sound classification result has low confidence (predicted accuracy of classification prediction $<$ treshold) or is unrecognized to prevent misleading contextual feedback.

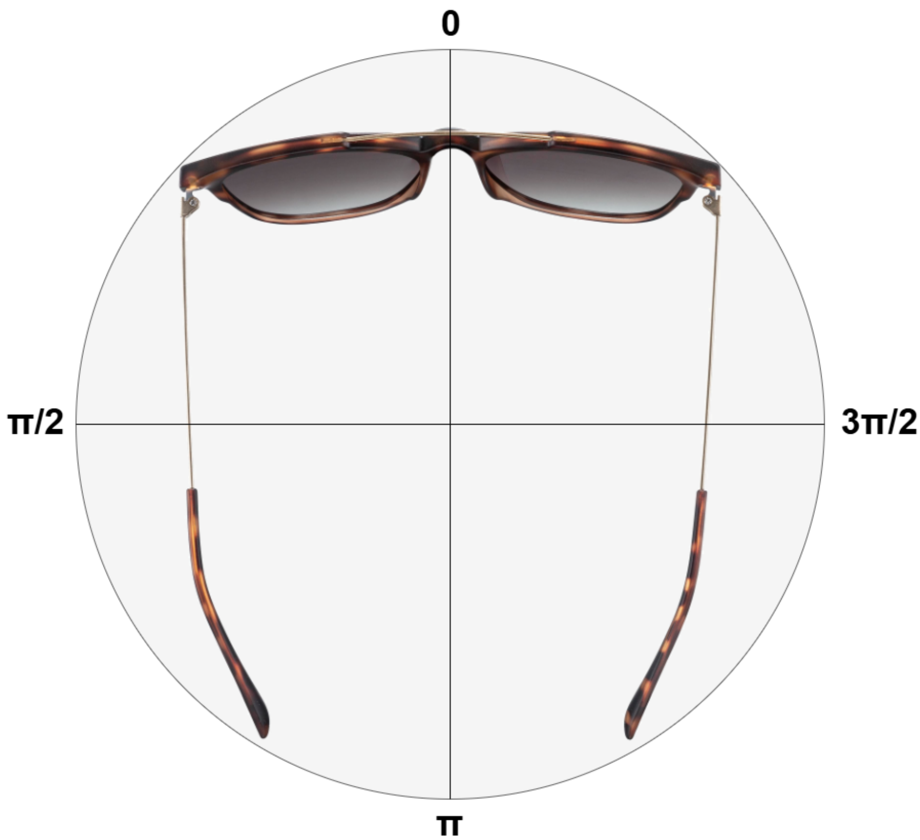

Figure 3: Coordinate System.

Non-Functional Requirements

- **NFR5.1:** The frequency analysis shall classify atleast 3 distinct sound sources relevant to people who are hard of hearing. Three was chosen to provide a minimal base that is useful for the stakeholders.
- **NFR5.2:** The frequency analysis component shall achieve a minimum classification accuracy of $A_{classification}$ under normal operating conditions.
- **NFR5.3:** The component shall estimate the direction of arrival of an audio source with a maximum error of θ_e degrees. This requirement ensures that the system can correctly notify users of sound sources within their respective Euclidean quadrants.

3.2.6 Visualization Controller

Functional Requirements

- **FR6.1:** The visualization controller component shall notify users of the direction of an audio source whenever it is detected by the system.

- **FR6.2:** The visualization controller component shall alert users when the core safety features such as direction determination or classification fail. This ensures that users are aware of degraded safety functions.

Non-Functional Requirements

- **NFR6.1:** The visualization controller component shall prioritize to display the most safety-critical information at the time because display space is limited.

- **NFR6.2:** The visualization controller component shall present information in a non-intrusive manner, minimizing visual obstruction so users can safely perform external activities. This allows the user to wear the glasses and perform daily activities without distraction.

3.2.7 Microphone

Functional Requirements

- **FR7.1:** The microphone shall collect soundwaves from its environment and translate them to digital representation.

- **FR7.2:** The microphone shall collect soundwaves within the human hearing most audible range (250 Hz - 8 kHz) [3].

Non-Functional Requirements

- **NFR7.1:** The microphone shall collect soundwaves at 16 kHz. According to Nyquist's theorem, audio sampling should be greater than two times the highest frequency to avoid aliasing [7]. The maximum frequency most audible to humans is approximately 8 kHz. [3].

3.2.8 Output Display

Functional Requirements

- **FR8.1:** The output display shall notify users of the classified sound source.

- **FR8.2:** The output display shall notify users of the direction of the sound source.

Non-Functional Requirements

- **NFR8.1:** The output display shall operate at a minimum of 30 display updates per second to ensure low latency and provide information to the user without perceptible delay.

3.2.9 Microcontroller

Functional Requirements

- **FR9.1:** The microcontroller shall run the main Audio360 software in a closed environment. No external devices other than the microphones and output display shall be able to connect to it. This ensures protection against unauthorized access adhering to Goal 7: security and privacy.

Non-Functional Requirements

- **NFR9.1:** The microcontroller shall have a clock speed of atleast 100 MHz to support audio sampling at 16 kHz, derived from NFR7.1. This allows approximately 5000 computational steps to process each audio sample, which is enough to meet core features functionality.

- **NFR9.2:** The microcontroller shall provide atleast four microphone inputs to support directional analysis of audio. Directional analysis requires a minimum of two microphones. However, since the microphones are mounted on glasses and sound damping occurs across the user's head, at least four microphones, one in each corner, are required to achieve reliable directional analysis.

3.3 S.3 Interfaces

Audio360 is designed as a closed embedded system due to its safety-critical application domain, refering to functional requirement FR9.1. External access is intentionally restricted to ensure reliability, security, and user safety. The only user interface is the integrated visual display. The display provides real-time information by notifying users of detected sounds and their corresponding direction. No external programmatic interfaces (APIs) will be provided. This design decision ensures that system integrity is maintained and that no external software can interfere with the safety-critical operation of the device.

3.4 S.4 Detailed usage scenarios

This section provides detailed versions of the high-level scenarios from G.5, with specific technical details and interaction flows.

3.4.1 S.4.1 Pedestrian Crossing Detection

User Story: As a deaf individual, I want to be alerted when vehicles are approaching so I can cross streets safely.
Scenario: A user is walking in an urban environment and approaches a street intersection. As they prepare to cross, a car approaches from their left side.
Interaction Flow:

1. System detects engine sound pattern characteristic of a vehicle

2. DoA algorithm estimates direction
3. Audio classification identifies the sound as a vehicle
4. Smart glasses display visual indicator showing direction and notifies the user of a vehicle approaching
5. User recognizes the alert and waits for the vehicle to pass before crossing safely

3.4.2 S.4.2 Kitchen Safety Alert

User Story: As a deaf individual, I want to be notified of household safety sounds so I can respond to potential hazards.
Scenario: A user is cooking in their kitchen when a tea kettle on the stove begins to whistle.
Interaction Flow:

1. Microphone array captures the high-pitched whistle sound
2. System processes the audio and identifies the characteristic pattern
3. DoA algorithm estimates the direction
4. Audio classification identifies it as a kettle or alarm sound (90% accuracy target)
5. Smart glasses display directional indicator notifying the user of a kettle whistling ($\leq$ 1s latency)
6. User turns toward the alert and removes the kettle from heat, preventing potential hazard

3.4.3 S.4.3 Social Interaction

User Story: As a deaf individual, I want to know when someone is trying to get my attention so I don't miss social interactions.
Scenario: A user is in a crowded room when someone calls their name from across the space.
Interaction Flow:

1. System captures audio from all directions through the microphone array
2. Voice activity detection identifies human speech patterns
3. DoA algorithm estimates the direction
4. Audio classification identifies the sound as speech or human voice
5. Smart glasses display the information with directional arrow
6. User turns in the indicated direction to make eye contact and engage in conversation

3.4.4 S.4.4 Workplace Awareness

User Story: As a deaf worker, I want to be aware of industrial safety warnings so I can maintain workplace safety.
Scenario: A user is working in an industrial setting when a forklift begins reversing nearby, emitting a warning beep.
Interaction Flow:

1. System detects the characteristic beeping pattern through microphone array
2. DoA algorithm estimates the direction
3. Audio classification identifies it as a warning signal or industrial alarm
4. Smart glasses display prominent visual indicator notifying the user of a forklift backing up and directional arrow
5. User steps aside to maintain a safe distance from the moving equipment

3.5 S.5 Prioritization

This section classifies all system behaviors, interfaces, and scenarios by their degree of criticality to ensure proper development prioritization and resource allocation. Requirements are organized into three criticality levels: Critical, Important, and Desirable, with corresponding development stages.

3.5.1 S.5.1 Critical Requirements (Stage 1 - Foundation)

Critical requirements form the essential foundation of the system and must be implemented first. These requirements are necessary for basic system operation and safety.

Estimated deadline: December 9th, 2025

Core Hardware and Infrastructure:

- FR7.1 - Microphone soundwave collection and digital conversion
- FR7.2 - Human most audible range coverage (250 Hz - 8 kHz)
- NFR7.1 - 16 kHz sampling rate
- FR9.1 - Closed microcontroller environment
- NFR9.1 - Minimum 100 MHz clock speed
- FR9.2 - Four microphone inputs for directional analysis

Sound Classification:

- FR5.1 - Sound source classification
- NFR5.1 - Classification of 3+ distinct sound sources
- NFR5.2 - 90% classification accuracy

Basic Audio Processing:

- FR1.2 - Audio signal processing and synchronization
- FR3.1 - Digital to frequency domain conversion
- FR3.2 - Signal amplitude normalization
- NFR3.1 - Frequency domain accuracy ($\leq$ 10% error)
- NFR1.2 - Processing faster than 16 kHz sample rate

Error Handling and Reliability:

- FR1.3 - Memory error handling
- FR1.4 - Continuous hardware diagnostics
- FR2.3 - Driver error code returns
- NFR2.1 - Immediate error propagation

Basic Display Functionality:

- FR8.1 - Sound source classification display
- FR8.2 - Sound direction display
- NFR8.1 - 30 display updates per second

3.5.2 S.5.2 Important Requirements (Stage 2 - Core Features)

Important requirements implement the primary functionality that delivers the system's main value proposition. These build upon the critical foundation.

Estimated deadline: February 9th, 2026

Direction of Arrival (DoA) Analysis:

- FR5.2 - Direction of arrival estimation

- FR5.3 - Angular representation in radians
- NFR5.3 - Maximum 45 degrees DoA error

Advanced Audio Processing:

- FR3.3 - Spectral leakage filtering
- FR3.4 - Hardware acceleration utilization
- FR3.5 - Audio anomaly detection
- NFR3.2 - Scalability to 4096 frames
- NFR3.3 - 90% processing success rate

System Integration:

- FR2.1 - Driver layer hardware interface
- FR2.2 - Permission-based hardware access
- FR2.4 - Data integrity maintenance
- FR4.1 - Audio360 Engine data retrieval
- FR4.2 - Component notification system
- FR4.3 - Data flow control

Visualization and User Interface:

- FR6.1 - Direction notification display
- FR6.2 - Safety feature failure alerts
- NFR6.1 - Safety-critical information prioritization
- NFR6.2 - Non-intrusive display design

3.5.3 S.5.3 Desirable Requirements (Stage 3 - Enhancement)

Desirable requirements enhance system performance and user experience but are not essential for basic operation.

Estimated deadline: April 9th, 2026

Advanced System Features:

- FR1.1 - Priority-based task scheduling
- NFR1.1 - Monotonic frame sequence processing
- FR4.4 - Feature disabling during microphone faults
- FR5.4 - Low confidence classification notifications

Performance Optimization:

- NFR4.1 - Conflict-free data polling
- NFR4.2 - Timing constraint compliance
- NFR4.3 - Immediate data disposal after analysis

3.5.4 S.5.4 Scenario Prioritization

Critical Scenarios (Stage 1):

- S.4.2 - Kitchen Safety Alert (highest safety impact)
- S.4.4 - Workplace Awareness (industrial safety)

Important Scenarios (Stage 2):

- S.4.1 - Pedestrian Crossing Detection (public safety)
- S.4.3 - Social Interaction (quality of life)

3.5.5 S.5.5 Development Priority Rationale

The prioritization follows a safety-first approach:

1. **Stage 1 (Critical):** Establishes reliable hardware foundation and basic audio processing capabilities. Focuses on system stability and error handling to prevent failures that could compromise user safety.
2. **Stage 2 (Important):** Implements core DoA and classification features that deliver the primary value proposition. Kitchen and workplace safety scenarios are prioritized due to their immediate safety implications.
3. **Stage 3 (Desirable):** Adds performance optimizations and advanced features that enhance user experience but are not essential for basic safety functionality.

This staged approach ensures that the most safety-critical features are implemented first, providing immediate value to users while building toward a complete system. The prioritization also considers technical dependencies, with lower-level components (hardware, drivers) being implemented before higher-level features (classification, visualization).

3.6 S.6 Verification and acceptance criteria

This section specifies the conditions under which the implementation will be deemed satisfactory. The verification criteria are organized by the major functional areas of the system and are directly traceable to the goals defined in G.1 Context and overall objective and the functional/non-functional requirements defined in S.2 Functionality.

3.6.1 S.6.1 Audio capture verification

Related to Goal 1, FR1.2, FR7.1, FR7.2, and NFR7.1.

- **VC-1.1 Synchronized sampling:** The system shall demonstrate that all four microphones in the array capture audio samples with synchronously, that is all microphone data are captured within 4 timesteps (one timestep per microphone). This will be verified through oscilloscope measurements or timestamp analysis of captured data.
- **VC-1.2 Sampling rate:** The system shall maintain a minimum sampling rate of 16 kHz across all microphone channels, verified through analysis of captured audio buffer timestamps.
- **VC-1.3 Continuous operation:** The system shall demonstrate continuous audio capture for the duration the power lifecycle of the system (single charge on a battery) without buffer overruns, dropped samples, or system crashes, verified through stress testing and log file analysis.

3.6.2 S.6.2 Direction of arrival (DoA) verification

Related to Goal 2, FR5.2, FR5.3, and NFR5.3.

- **VC-2.1 Single source accuracy:** For a single sound source in a controlled environment, the system shall estimate the direction of arrival with an angular error of less than or equal to θ_e in all test cases, verified through testing with known sound source positions at various angles, θ, that are multiples of θ_e and satisfies $0 \leq \theta \leq 360°$.
- **VC-2.2 Processing latency:** The DoA estimation algorithm shall complete processing and output a direction estimate within t_{lat} of sound detection, verified through timestamp logging of detection and output events.
- **VC-2.4 Range coverage:** The system shall detect and provide directional estimates for sounds originating from any direction in the full 360° horizontal plane around the user, verified through systematic testing at multiple angles.
- **VC-2.5 Consistency:** For a stationary sound source, the system shall provide consistent direction estimates within θ_e, verified through repeated testing.

3.6.3 S.6.3 Sound classification verification

Related to Goal 3, FR5.1, FR5.4, NFR5.1, and NFR5.2.

- **VC-3.1 Classification accuracy:** The audio classification system shall achieve at least $A_{classification}\%$ accuracy in identifying sound categories from a predefined set, verified through testing with a labeled dataset.
- **VC-3.2 Processing latency:** The classification algorithm shall complete processing and output a classification label before the next microphone audio sampling, verified through timestamp logging.
- **VC-3.3 Category coverage:** The system shall support classification of at least 3 distinct sound categories relevant to safety and daily life, verified through documentation of the classification model and testing with representative samples from each category.
- **VC-3.4 Handling unknown sounds:** The system shall gracefully handle sounds that do not match any trained category by either assigning a low-confidence score or providing a generic label (e.g., "unknown sound"), verified through testing with sounds outside the training set.

3.6.4 S.6.4 Visual display verification

Related to Goal 4, FR6.1, FR8.1, FR8.2, NFR6.2, and NFR8.1.

- **VC-4.1 Display latency:** The complete system pipeline from sound detection to visual display on the smart glasses shall complete within t_{lat}, verified through synchronized video recording of sound source activation and display appearance.
- **VC-4.2 Display visibility:** Visual indicators on the smart glasses shall be clearly visible and distinguishable under typical indoor and outdoor lighting conditions, verified through user testing and subjective evaluation by test participants.

3.6.5 S.6.5 User interaction verification

Related to Goal 5, FR6.2, and NFR6.1.

- **VC-5.1 Setup simplicity:** A new user shall be able to power on the device and begin receiving audio alerts without requiring external documentation, verified through observing participants unfamiliar with the system using it for the first time.

3.6.6 S.6.6 System reliability verification

Related to FR1.3, FR1.4, FR2.3, FR3.5, FR4.4, NFR2.1, NFR3.3, and NFR4.1.

- **VC-6.1 Startup reliability:** The system shall successfully complete initialization and enter operational mode in all power-on attempts, verified through repeated startup testing. This ensures the system is working all the time as it is a safety critical system.

4 Project

4.1 P.1 Roles and personnel

All project staff have responsibilities they must carry out in addition to the individual work required for deliverables. Admin related roles, and general technical subgroups have been outlined under Team Member Roles section of the Development Plan.

In addition to these responsibilities, each project member has been designated as an **expert** in a specific area of focus essential to the development of this system. The designated focus areas are outlined below:

- Research in using hardware acceleration methods and other code optimization methods for embedded development, since the system is deployed on microcontroller with limited CPU and memory space. **(Sathurshan Arulmohan)**.
- Hardware integration, including implementation of low-level hardware interface for high-level software. Features should be abstracted so that it can integrate with other hardware devices. Since the hardware requires an API integration for this, this focus area is needed. **(Omar Alam)**.
- Analysis of methods to extract directionality from a microphone array, since multple microphones needs to be processed. and run some algorithm to determine which direction it is coming from. This algorithm needs to be researched. **(Kalp Shah)**.
- Exploration of techniques for separating multiple audio sources from a single microphone input using Independent Component Analysis (ICA). This focus area is required since the system's environment will contain a mixture of sounds that need to analyzed and processed independently. **(Jay Sharma)**.
- Development of audio-fingerprinting for classifying sound sources based on the frequency domain representation of the audio signal. This feature needs to comply with the processing and memory limits of the compute unit.**(Nirmal Chaudhari)**.

All experts share the following responsibilities for their area of focus.

- Look through research articles, and technical evaluations to come up with feasible approaches for proposed methods.
- Collaborate with other team members to discuss findings.
- Maintain clear and organized documentation of sources and proposed methods.
- Complete implementation of the focus area in the system.
- Testing to validate and verify the feature.

4.2 P.2 Imposed technical choices

Microcontroller: The system shall use a dedicated microcontroller with a minimum clock speed of 100 MHz. This constraint is imposed to ensure that the audio processing tasks defined in functional requirement NFR9.1 can be completed within timing constraints. A microcontroller is also chosen because it can be physically embedded within the glasses, ensuring low power consumption and compact integration. For this capstone project, a larger developer microcontroller will be used for ease of development, regardless of if it can be embedded into the glasses. Future iterations must use a microcontroller that is small enough to fit inside the glasses and has the required computer power.

Microphones: The system shall use between two and four omnidirectional microphones to collect audio data from the environment. This range of microphones enables directional detection while minimizing hardware complexity, cost, and power consumption.

Programming Languages: The embedded firmware shall be implemented in C/C++. This choice is imposed because compiled C/C++ binaries can be directly executed on any microcontroller, and the language provides low level access to system resources, memory, and hardware acceleration functions necessary for efficient audio processing.

4.3 P.3 Schedule and milestones

To better understand the needs of the stakeholders, there are various milestones that need to be completed within the realm of the requirements process. These milestones can be generalized as a list of tasks, with their deadlines outlined below.

1. Initial Supervisor Consultation (2025-09-15)
2. Extraction of project soft and hard goals (2025-09-22)
3. Technical analysis for feasibility of project goals (2025-09-29)
4. Extraction of requirements from goals and technical assessment (2025-10-2)
5. Requirements Documentation (2025-10-10)
6. Validation Plan (2025-10-27)
7. System Design Plan (2025-10-27)
8. System Integration (2025-01-19)
9. Verification & Validation of system (2026-02-02)
10. Consultation with primary stakeholder (TBD)

4.4 P.4 Tasks and deliverables

1. **Initial Supervisor Consultation (2025-09-15)**

   Tasks:

   - Meet with the supervisor to clarify the project vision.
   - Discuss key goals, available resources and expected POC outcomes.
   - Identify initial research domains.

   Expected Outcomes:

   - Team has a shared understanding of project objectives and potential risks associated with the project scope and goals.
   - Defined list of research priorities outlined in section P.1.

2. **Extraction of project soft and hard goals (2025-09-22)**

   Tasks:

   - Based on team elicitation, brainstorm and document list of soft goals.
   - Deconstruct soft goals into a list of hard goals using definition, contribution, decomposition techniques.
   - Discuss trade-offs of defined hard goals.

   Expected Outcomes:

   - Comprehensive list of soft and hard goals, clearly defining the project direction.
   - Consensus among team members and supervisor on achievable project objectives.
   - Prioritized list of goals for verification testing.
   - Validated list of goals with primary stakeholders.

3. **Technical analysis for feasibility of project goals (2025-09-29)**

   Tasks:

   - Review research articles relevant to goals extracted during the elicitation stage.
   - Conduct feasibility assessments for key goals set in the project.
   - Run small-scale prototype, or test to confirm technical viability.

   Expected Outcomes:

   - Confirmation of which project goals are technically achievable within the scope.

- Revised goals based on hardware and software capabilities.
- Early identification of design constraints.

4. **Extraction of requirements from goals and technical assessment (2025-10-2)**

   Tasks:

   - Translate validated soft and hard goals into specific system requirements.
   - Identify dependencies and relationships between requirements.
   - Ensure each requirement is unambiguous, measurable and achievable within the scope.

   Expected Outcomes:

   - Drafted list of functional and non-functional requirements derived from goals.
   - Clear mapping between project goals and corresponding requirements.

5. **Requirements Documentation (2025-10-10)**

   Tasks:

   - Iteratively updating sections in SRS document based on the results of different tasks mentioned in P.3.

   Expected Outcomes:

   - Completed Rev 0 SRS document
   - Software architecture and design can commence with a clear path using SRS as a guidance.

6. **Validation Plan (2025-10-27)**

   Tasks:

   - Come up with a plan for validating the system meets the requirement specification. This plan should be exhaustive enough to confirm that all of the stakeholders' goals have been met.

   Expected Outcomes:

   - The V&V deliverable has been completed, and clearly outlines the plan for validation and verification.

7. **System Design Plan (2025-10-27)**

   Tasks:

   - Each focus area expert works on completing the design plan for the implementation of that feature.
   - The design plan will be communicated to the rest of the team and the project advisor for validation on the design and any feedback will be addressed.
   - Each focus area expert works on prototyping the feature in their focus area, and validate that it works correctly.

   Expected Outcomes:

   - Each focus area has a fully completed design plan based on the investigation completed by the expert.
   - The plan has been written in Rev 0 of the Design Document.
   - Prototyping for features in focus areas has been completed, and validated by the rest of the team.

8. **System Integration (2025-01-19)**

   Tasks:

   - Integrate hardware and software modules into a single functional system.
   - Connect and configure communication interfaces between subsystems (sensor, microcontroller, output display).
   - Perform initial integration testing to verify data exchange between the various components.
   - Debug any errors that arise in this process.

   Expected Outcomes:

   - All system components function together correctly, and system works as expected.
   - Interfaces (drivers, APIs, protocols) are correctly configured and stable.
   - Identified and resolved all integration errors, with documentation updated based on fixes.

9. **Verification & Validation of System (2026-02-02)**

   Tasks:

   - Using the Rev 0 V&V report, team goes through plan for validation and verification of the system against requirement specifications.

- For any unmet requirements based on the validation plan, the team re-assess the requirements to see it's priority in the given timeline. Previous requirement documents are updated accordingly.

Expected Outcomes:

- All documentation has been updated based on the current implementation of the system.
- Team has a clear list of unmet requirements that need to be addressed.

10. **Consultation with primary stakeholder (TBD)**

    Tasks:

    - Meet with the McMaster Sign Language club to conduct a semi-structured interview.
    - Document all findings from interview specific to the questions asked.
    - Update stakeholder goals and requirements based on the results of the interview.

    Expected Outcomes:

    - SRS document has been updated to be inline with McMaster Sign Language club needs.

4.5 P.5 Required technology elements

Microphones: Capable of sampling audio at 16 kHz in order to capture the full audible spectrum without aliasing. This complies with non function requirement NFR7.1.
Analog to Digital Converter: An analog to digital converter is required between each of the microphone and the microcontroller to convert analog signals into digital form suitable for processing.
Microcontroller: The microcontroller shall have sufficient processing speed and I/O bandwidth to synchronously acquire data from multiple microphones without delays or sampling misalignment.
Output Display: A visual display module to show audio classification and detection to users who are hard of hearing.

4.6 P.6 Risks and mitigation analysis

The project has several risks to meeting the schedule outlined in section P.4. The following table summarizes the identified risks, their potential impact, and possible adaption measures:

Risk	Impact	Adaptation Measures
Unable to create feasible design plan for focus area.	Delays in design phase and deliverable submission.	Consult with project advisor for guidance. Reassess and narrow scope of focus area.
Unable to engage with primary stakeholder: McMaster ASL.	Misalignment of project goals	Try to find another stakeholder that can provide similar feedback. Seek advice from project supervisor.
System is unable to meet SRS requirements.	Unable to validate and verify system functionality as per plan.	Reassess requirements and project scope. Prioritize critical requirements. Seek advice from project supervisor
Team members unable to meet deadlines due to personal reasons.	Delays in deliverable completion. Additional workload on other team members.	Reallocate tasks amongst other team members. Refer to the team charter.
Hardware components are damaged during development.	Delays in development and testing of system.	Order spare components in advance. Budget for potential replacements.
Excessive hardship in integrating hardware with software.	Delays in testing system integration.	Write modular code to isolate hardware-software interface. Rigorously defined module interfaces.
Insufficient audio algorithm performance.	System fails to meet accuracy and latency requirements.	Explore alternative algorithms. Simplify problem scope. Seek advice from project supervisor.
Team conflicts and communication breakdowns.	Reduced productivity and morale.	Address conflicts early through open communication. Regular check-ins as described in team charter. If necessary, involve course instructor.

Table 3: Risk and mitigation analysis.

4.7 P.7 Requirements process and report

The development of this project requires iterating over various stages of the requirements process. These stages include:

1. **Elicitation**

   (a) Conduct stakeholder interviews and surveys to gather users' needs.

   (b) Review background documents and research articles describing how core features have been implemented in the past.

(c) Consult with the technical supervisor of this application (MVM) to gain insight into what approaches can be used.

2. **Analysis**

   (a) Based on information retrieved from the elicitation process, derive a list of soft and hard goals for the application.

   (b) Using the goals defined previously, derive a list of key requirements. This will be influenced by the goals of the stakeholders, but also input on potential limitations based on discussion with the supervisor.

   (c) Group requirements into functional vs. non-functional categories.

   (d) Prioritize requirements using the MoSCoW framework (Must, Should, Could, Won't).

   (e) Iteratively evaluate defined requirements based on new constraints that arise during implementation.

3. **Documentation**

   (a) Write requirements in a structured format that is clear, testable, and unambiguous.

   (b) Use labels for different requirements and goals for traceability.

4. **Specification**

   (a) Using the `docs/` folder in the main repository for this project, update the various reports with the latest information based on what was discussed or finalized in other stages of the elicitation process.

5. **Validation**

   (a) Share draft requirements with stakeholders for confirmation.

   (b) Share requirements with the project supervisor to get expert opinion on feasibility of requirements.

   (c) Within the team, ensure the requirements are feasible, measurable, and aligned with the project scope.

As this project follows the V-model methodology, Figure 4, the team will aim to make the various stages of the requirements process mentioned above linear. Since although this model allows the team to re-visit previous parts, it will cost a lot to change in later stages of the project. This is why this team is heavily prioritizing the initial stages of the requirements process.

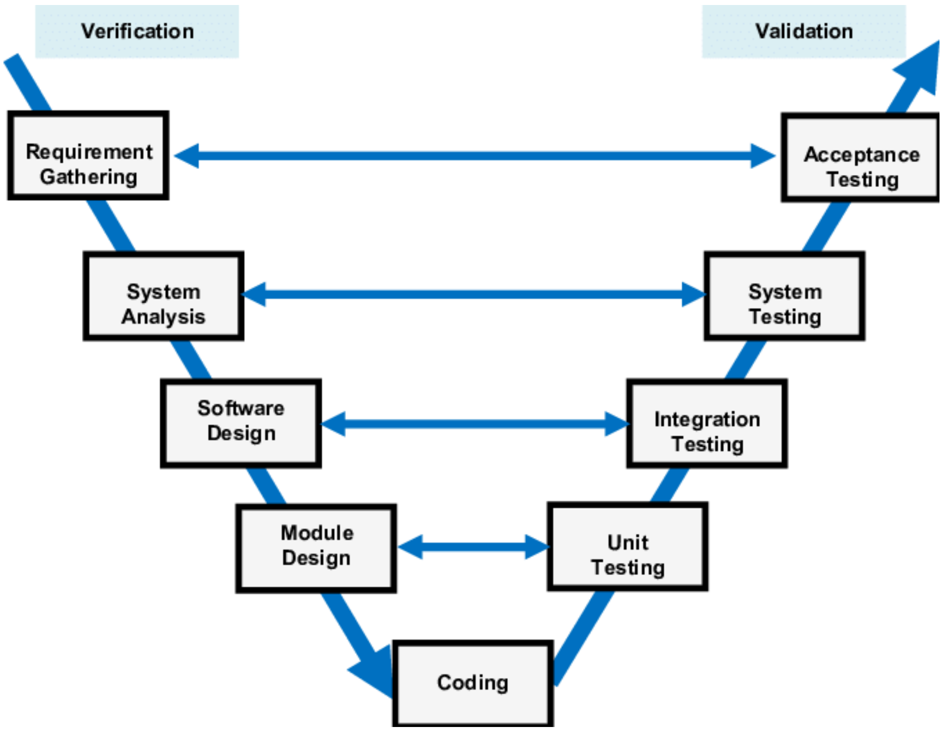

Figure 4: Illustration of the v-model process.

References

[1] Healthing.ca, “Hearing loss in canada: Stats, impact and resources,” *Healthing.ca*, 2025.

[2] N. I. on Deafness and O. C. Disorders, “Hearing aids,” *National Institute Deafness and Other Communication Disorders*, 2022, nIH Publication No. 99-4340.

[3] T. Tedeschi, *Human hearing frequency and audible range*. Miracle-Ear, 2025. [Online]. Available: https://www.miracle-ear.com/blog-news/human-hearing-range

[4] Government of Canada, “Criminal code, rsc 1985, c c-46, s 184,” *Justice Laws Website*, 1985.

[5] ——, “Personal information protection and electronic documents act, s.c. 2000, c. 5,” *Justice Laws Website*, 2000.

[6] CMSIS-DSP, “Complex fft functions,” 2025.

[7] DataForth, “Sampling laws,” *DataForth*, 2025. [Online]. Available: https://www.dataforth.com/data-acquisition-and-control-sampling-law

Appendix — Reflection

Questions 3 and 5 are answered as a team.

1. What went well while writing this deliverable?

   Kalp: I think what went really well during the writing of the SRS document was the frequent in-person work sessions. We were all able to sit together in a room, quickly review each other's work, make changes, discuss unclear assignments, and so on. This process, compared to the individual writeup and review process we used earlier, was much more efficient and effective, especially since this document was very interdependet (goals section for example having content related to the requirements section).

   Nirmal Chaudhari: Coming up with the schedule and milestone for this project was relatively easy for the elicitation and documentation stages since it was basically just a recap and coherent with what we just went through. Moreover, since our team had come up with the list of focus areas in advance, coming up with team roles and added responsibilities was relatively easy as well.

   Sathurshan: In this deliverable, the team effectively collaborated on specific sections, allowing us to provide constructive feedback and iteratively refine our requirements during the initial write-up. This process helped align our goals for the system and provided a clearer understanding of what needed to be achieved.

   Omar: Throughout the deliverable writing process, the team communicated effectively and responded to feedback on the pull requests in a timely manner. This allowed us to rapidly iterate on the document. Since everyone brings an equal level of commitment and enthusiasm to the project, it made the collaboration process smooth and enjoyable.

   Jay: Building on our earlier deliverables made the writing process much smoother. Having already defined the problem scope and development plan gave us a solid foundation to work from, so translating those ideas into detailed requirements felt natural. The structured template also helped keep everything organized, making it easier to ensure completeness across all sections.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   Kalp: I think the main pain point was the way that we divided up the work on the document since many of the sections were dependent on each other. This often lead to some people on the team waiting for others to finish their section before they could start their own so that there wasn't conflicting information or text in the document. This process was slighly improved with the frequent in-person work sessions, but it was still a pain point.

   Nirmal Chaudhari: While working on the environment section, intially coming up with the list of components in the environment the system will have to interact with as difficult. This is because of the ambuguity that initially existed with what we can consider the "environment" for this system. This unclearity was resolved by coming up with a very high level use-case scenario of what a typical person would be interacting with when using the device. **Sathurshan:** Many sections were dependent on others being completed first, which blocked some of the writing process. With the granted extension, several sections were delayed, leading to a time crunch toward the end. To address this, the team organized multiple collaborative work sessions to work on the SRS document together. This allowed us to exchange ideas in real time, resolve blockers quickly, and progress in parallel. Moving forward, during the project planning stage, the team should prioritize dependency related issues and set internal deadlines to ensure smoother progress.

Omar: There was some friction when it came to sections that relied on other sections being completed first. However, the team was able to work through these issues by holding several in-person and virtual meetings to discuss the document and make progress together. This allowed us to quickly resolve any blockers and ensure that everyone was on the same page.

Jay: Striking the right balance between technical precision and readability was challenging. Some sections needed enough detail to be actionable, but too much made them hard to follow. We addressed this by reviewing each other's sections and providing feedback on clarity, which helped us converge on a consistent level of detail throughout the document.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

   A lot of the requirements related to focus areas defined in our SRS document were inspired by our project supervisor. He gave great insight into what major components need to be researched for this system to work. For example, he went over the requirement of needing 4 ADC converters in our microprocessor to retrieve synchronized audio input across all 4 microphones. If he didn't give this insight early on, the team would have been stuck in the later stages on the project with a microprocessor that will not work well for this system. Furthermore, he gave good information of how the team can go about using Independent Component Analysis to seperate audio into sources. He also mentioned we shouldn't use deep machine learning models for audio classification, since they won't be able to run on the microprocessor well.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

   Nirmal Chaudhari: For this project the three courses I think will enable us to be the most successful are: Signals & Systems, Requirements Engineering and Software Design 2. Signals & Systems was important since the entire project revolves around processing and analyzing audio signals in the frequency domain. Requirements Engineering is important in helping us figure out our requirements and ensuring throughout the entire process that what we are building is the right thing. And Software Design 2 is useful in helping us implement best practices into the project, thus making it sustainable in the future.

Sathurshan: 3MX3: signal processing, 2GA3: computer hardware architecture. 3RA3: software requirements, 2DA4: software design, 3A04: software architecture, 3S03: software testing.

Omar: The courses that will help us the most with this project are: Signals and Systems (3MX3) - This course provides a solid foundation in signal processing, which is crucial for our project focused on audio signal analysis. Concurrent Programming (3BB4) - This course teaches us the core principles of concurrent programming which will be essential for implementing real-time processing on our microcontroller.

Kalp: The courses that will probably help us the most for this project are 3MX3 (Signals & Systems with Dr. Mohrenschildt). This course taught us many of the signal processing algorithms for that we will likely need to apply during our audio analysis for the system. Another good course would be 3A04 (Software Architecture) that taught us how to design and implement large scale software systems which will be important for our project as we we will be designing many modular components that work together. The planning techniques from the course, specifically, will be very useful.

Jay: 3MX3 (Signals & Systems) is definitely critical for understanding audio processing and frequency analysis. Beyond that, 3BB4 (Concurrent Systems) will be valuable for managing real-time constraints on the microcontroller, and 2GA3 (Computer Architecture) will help us optimize performance on embedded hardware. The software engineering courses like 3RA3 and 3S03 are also important for ensuring we build a reliable, well-tested system.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

   Each memmber of the team requires the following qualfiications as contributing developers to the team.

   - Embedded software development.
   - Strong software design for features being implemented on any microcontroller platform.
   - Strong testing skills.
   - Strong debugging skills.

In addition to this, since each team member is a focus area expert, they require the following skills and competencies to carry out that role.

- Look through research articles, and technical evaluations to come up with feasibile approaches for proposed methods.
- Collaborate with other team members to discuss findings.
- Maintain clear and organized documentation of sources and proposed methods.
- Ensure that all research and implementation choices align with project objectives, timeline and budgeting costs.
- Based on the confirmed approach, complete the full implementation of that focus in the system.
- After implementation, create test cases that cover's the main functionality of the feature in that focus area.
- Configure github pipeline to run those tests on every PR and merge into a feature branch.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

   Kalp: I think the main focus for knowledge areas to explore has to be around embedded software development and hardware integration. Since I've only done software development in industry before, I have experience developing, testing, and debugging software, but have never explored the hardware side of the systems.

   Nirmal Chaudhari: For embedded software development and strong software design, two approaches are available: (1) reviewing processor documentation and (2) practicing and test building small programs on the board to see how it targets key peripherals work. I will focus on the second approach since practical experience seems more important. For testing and debugging skills, the team can (1) study or use existing knowledge of frameworks and (2), conduct peer reviews with other members on the team. For this I would prefer the second approach since working with the team on real bugs would help me grow as a developer to see how others resolve bugs. For research and technical evaluation, the knowledge can be gained by (1) reading research papers existing in the focus area and (2) contacting domain experts like mvm for insights. I plan to start with the first option, since academic resources provides structure that can be referenced to later on.

   Sathurshan: I plan to focus on gaining knowledge in embedded software development, as it is something I want to specialize in the next few years.

   Omar: I believe the best way to learn any skill is by doing it and struggling through problems. Each microcontroller platform has its own quirks and tools. I plan to gain practical experience by developing smaller projects on the STM32 platform, which will help me understand its architecture. Each problem in our project can be subdivided into smaller projects, which will help me learn as I go. Additionally, I will refer to the STM32 documentation and online tutorials to supplement my learning.

Jay: I'll be focusing on Independent Component Analysis (ICA) for audio source separation. Two main approaches are: (1) implementing ICA algorithms from research papers and testing them with sample audio data, and (2) prototyping directly on the hardware with real microphone input. I'll start with the first approach since it lets me validate the core algorithms quickly without hardware dependencies, then transition to hardware testing once we have a stable foundation. I'll also lean on our supervisor's expertise to guide algorithm selection and avoid overly complex solutions.