Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Courtney Thurston	11/8/2018	Version 1.0 Approved	1.0
Courtney Thurston	11/23/2018	Version 1.1 Approved; changed database provider	1.1
Jose Polo	12/08/2018	Version 1.2 Approved; changed web server provider	1.2
Albert DiCroce	12/28/2018	Version 1.3 Approved; defining dB and other minor hardware comments resolved	1.3
Daniel Silverio	01/03/2019	Version 1.4 Approved; change of Gbps and performance requirements comments	1.4
Jonathan Betancourt	01/11/2019	Version 1.5 Approved; complete first pass at resolving SRS hardware comments	1.5
Courtney Thurston	01/19/2019	Version 1.6 Approved; complete first pass at resolving SRS software comments	1.6
Daniel Silverio	01/24/2019	Version 1.7 Approved; second pass section 3 revisions	1.7
Albert DiCroce	01/27/2019	Version 1.8 Approved; second pass section 4 revisions	1.8
Daniel Silverio	02/01/2019	Version 1.9 Approved; new parts spec included	1.9
TEAM	02/12/2019	Version 2.0 approved; revisions of SRS v1.0 comments complete	2.0

1. Introduction

1.1 Purpose

The goal of this document is to provide support information on the Binaural Audio project (v1.0). It will attempt to explain the functionality of the program and the features it provides. Note: it will not fully describe how the program works or how the user should use it. For that purpose, one should read the user manual written by the creators of the project.

1.2 Document Conventions

Every requirement statement is to have its own priority, assigned to the left of the project requirement. The lowest numerical value assigned indicates the highest available priority level. For instance, a priority of P0 is more critical than an assigned priority of P1.

- **P0** – A requirement critical to core product function. Must be achieved.
- **P1** – A requirement likely needed for desired product function or ease-of-use. Should be achieved.
- **P2** – A requirement that may be pursued to add additional functionality to the product; a stretch goal.

Where provided, references within this document will use the MLA 8 format.
Especially important text, or titles of relevant sections, are **bolded**.

1.3 Intended Audience and Reading Suggestions

- **Developers** may find the System Requirements Specifications (SRS) document helpful in defining desired functionality and the hardware protocols and software methods used to achieve this result.
- **Project managers** may use the SRS to aid in understanding the scope of the project and the hardware and software components herein.
- **Marketing staff**, where applicable, may find the SRS helpful in identifying key selling points of the product.
- **Users** may find the SRS helpful for understanding the full swath of features available within the product, and the specific mechanisms by which those features were achieved.
- **Testers** may find the SRS helpful in identifying requirements to be tested, which should match testing plan items on a minimum of a 1:1 basis; best practice calls for at least a 3:1 basis (testing item to requirement item) for major requirement items, or requirements with critical priorities.
- **Documentation writers** may find the SRS helpful in identifying areas which may need additional documentation support for end user ease-of-use.
- **Customers** will find the SRS helpful in understanding the requirements that drove product design, and thus how to use the product to its fullest intended use.

1.4 Product Scope

The goal of the Binaural Audio capstone project is to create a lightweight integrated device consisting of a small camera, binaural microphones, and a head orientation tracker mounted onto a pair of glasses. An additional GPS unit will be contained within an Android Device. The device will allow the user to capture and record audio and video in the same manner that it was experienced organically by the user.

There are many applications and uses for such a device. For instance, it could be used in military settings for both training and combat scenarios. In training situations, realistic audio and video captured from a first-person perspective can be used to create the best training simulations where the recorded data could be played back during training. This methodology would provide the most realistic scenarios as each person would hear and see the video exactly as if they experienced it firsthand. In a combat scenario, the recorded audio could be transmitted between soldiers so that each soldier knows the relative position and environment of every other soldier. This would assist a team in identifying threats and ensuring that everyone remained coordinated and safe. Further, the device can assist users that struggle with short-term memory loss, a common side-effect of a large swath of medical conditions. If the user is having difficulties remembering a situation that happened, they can use their android device to easily playback the recorded environment to assist them in remembering the details.

1.5 References

"830-1998 - IEEE Recommended Practice for Software Requirements Specifications – IEEE Standard." IEEE Standard, IEEE, ieeexplore.ieee.org/document/720574/.

Baumgarte, F., and C. Faller. "Binaural Cue Coding-Part I: Psychoacoustic Fundamentals and Design Principles." *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, 2003, pp. 509–519., doi:10.1109/tsa.2003.818109.

"Binaural Cues for a Single Sound Source." *Spatial Audio Processing*, pp. 127–138., doi:10.1002/9780470723494.ch7.

"Binaural Cues for Multiple Sound Sources." *Spatial Audio Processing*, pp. 139–154., doi:10.1002/9780470723494.ch8.

Brown, C.p., and R.o. Duda. "A Structural Model for Binaural Sound Synthesis." *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, 1998, pp. 476–488., doi:10.1109/89.709673.

Choueiri, Edgar. "Binaural Audio Through Loudspeakers." *Immersive Sound*, 2017, pp. 124–179., doi:10.4324/9781315707525-6.

Ericson, Mark A. "Binaural Audio Symbology for Urgency Displays." *PsycEXTRA Dataset*, 2006, doi:10.1037/e577702012-011.

Flessner, Jen-Hendrik, et al. "Assessment and Prediction of Binaural Aspects of Audio Quality." *Journal of the Audio Engineering Society*, vol. 65, no. 11, 2017, pp. 929–942., doi:10.17743/jaes.2017.0037.

Geravanchizadeh, Masoud, et al. "Binaural Speech Intelligibility Prediction in the Presence of Multiple Babble Interferers Based on Mutual Information." *Journal of the Audio Engineering Society*, vol. 65, no. 4, 2017, pp. 285–292., doi:10.17743/jaes.2017.0004.

Hu, Jwu-Sheng, and Wei-Han Liu. "Location Classification of Nonstationary Sound Sources Using Binaural Room Distribution Patterns." *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, 2009, pp. 682–692., doi:10.1109/tasl.2008.2011528.

Klein, Florian, and Stephan Werner. "Auditory Adaptation to Non-Individual HRTF Cues in Binaural Audio Reproduction." *Journal of the Audio Engineering Society*, vol. 64, no. 1/2, 2016, pp. 45–54., doi:10.17743/jaes.2015.0092.

Laitinen, Mikko-Ville, and Ville Pulkki. "Binaural Reproduction for Directional Audio Coding." *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, doi:10.1109/aspaa.2009.5346545.

Litchfield, Peter. "How Our Special Music Works." *How Do Binaural Beats Work?*, www.binauralbeatsmeditation.com/the-science/.

Marquardt, Daniel, et al. "Theoretical Analysis of Linearly Constrained Multi-Channel Wiener Filtering Algorithms for Combined Noise Reduction and Binaural Cue Preservation in Binaural Hearing Aids." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, 2015, pp. 2384–2397., doi:10.1109/taslp.2015.2479940.

M. Tikander, A. Harma and M. Karjalainen, "Binaural positioning system for wearable augmented reality audio," 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684), New Paltz, NY, USA, 2003, pp. 153-156. doi: 10.1109/ASPAA.2003.1285854

Nandy, D., and J. Ben-Arie. "Estimating the Azimuth of a Sound Source from the Binaural Spectral Amplitude." *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, 1996, p. 45., doi:10.1109/tsa.1996.481451.

Powell, T. "Binaural Speaker Listener Tests." *IRE Transactions on Audio*, AU-6, no. 3, 1958, pp. 76–76., doi:10.1109/tau.1958.1166140.

Roginska, Agnieszka. "Binaural Audio Through Headphones." *Immersive Sound*, 2017, pp. 88–123., doi:10.4324/9781315707525-5.

Shabtai, Noam R., and Boaz Rafaely. "Generalized Spherical Array Beamforming for Binaural Speech Reproduction." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, 2014, pp. 238–247., doi:10.1109/taslp.2013.2290499.

Szurley, Joseph, et al. "Binaural Noise Cue Preservation in a Binaural Noise Reduction System With a Remote Microphone Signal." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, 2016, pp. 952–966., doi:10.1109/taslp.2016.2535199.

Takanen, M., et al. "Binaural Assessment of Parametrically Coded Spatial Audio Signals." *The Technology of Binaural Listening*, 2013, pp. 333–358., doi:10.1007/978-3-642-37762-4_13.

Wendt, Torben, et al. "A Computationally-Efficient and Perceptually-Plausible Algorithm for Binaural Room Impulse Response Simulation." *Journal of the Audio Engineering Society*, vol. 62, no. 11, 2014, pp. 748–766., doi:10.17743/jaes.2014.0042.

Zheng, Jianwen, et al. "A Linear Robust Binaural Sound Reproduction System with Optimal Source Distribution Strategy." *Journal of the Audio Engineering Society*, vol. 63, no. 9, 2015, pp. 725–735., doi:10.17743/jaes.2015.0066.

2. Overall Description

2.1 Product Perspective

The Binaural Audio project is a new, self-contained product. Though glasses-mounted camera products that capture audio already exist on the market, these products typically lack in two main areas: (1) they fail to provide binaural audio, and (2) they fail to provide head-tracking capabilities. Both are critical requirements to the Binaural Audio product, as the goal is to provide the end user with realistic playback of scenarios (e.g. a conversation) as they were experienced by the user; this is more accurately approximated with binaural audio, head tracking, and a video feed than simply by a standard camera and microphone combination.

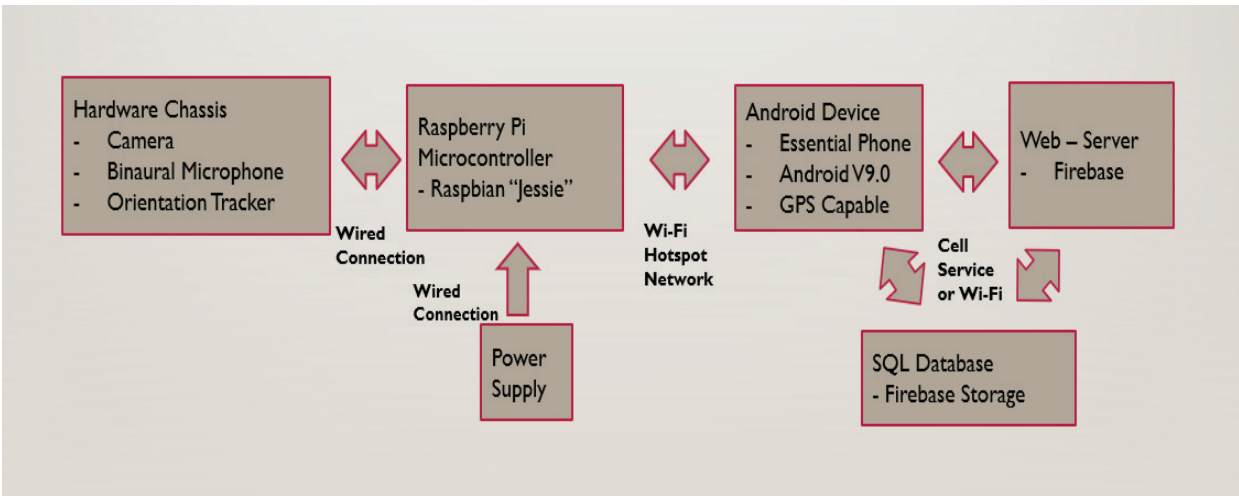

Figure 1: Architecture of Binaural Audio, V2.0

2.2 Product Functions

As **Figure 1** shows, the Binaural Audio product is comprised of several hardware and software components that are meshed together to provide the desired capabilities and performance. There are three core functionalities required for this product: the ability to capture video, the ability to capture binaural audio, and the ability to stream this audio and video concurrently. This product is distinct from existing products on the market due to (1) the binaural audio capability, which most market competitors lack, and (2) the product's compatibility with and support for various tech stacks. Most products on the market—The Extreme Go Camera or the Immortal Mics suite of products, for instance—lack integration with commonly used hardware and software suites. Meanwhile, the Binaural Project is built upon open source hardware and software: The Raspberry Pi platform, for instance, is developed for globally. Additionally, the platform itself is relatively cheap. This makes the product extensible and can be easily modified by curious customers. Similarly, the Binaural Project uses the tech stack of Firebase for its software needs. Firebase offers an extensive developer API which can be used to easily understand and change the source code, should a customer wish to further modify the project to fit their own needs. This further simplifies the maintenance load of the Binaural Project, as Firebase calls are simple to understand and used where needed throughout the project to make it modular, extensible, and portable with many other software platforms and libraries worldwide.

Furthermore, the audio is distinct from regular audio in that binaural is captured near to the ear drum itself, making the captured audio highly correlated to the sound heard by the user. By combining binaural audio recording with head orientation tracking, realistic multidimensional audio playback can be achieved. For instance, an end user replaying captured audio should be able to tell that the sound source originated to the right of the user who captured the audio.

The device will allow the user to capture and record audio and video in the same manner that it was experienced organically by the user. This is enabled by the glasses-mounted camera and the small profile binaural microphones, which sit inside the ear without blocking the eardrum. The head inertial measurement unit (IMU) is attached to the glasses and records head orientation of the user for realistic playback. A Raspberry Pi operating on the Jessie Version of Raspbian will be used as the microcontroller for data retrieval and processing. Connections will be made between the Raspberry Pi and the camera, microphones, and orientation tracker to compile and synchronize the data. Further, an Android device, web server, and SQL database will be used for storage and playback. A GPS unit integrated into the Android Device will also record GPS positioning of the user and this data will be synchronized with the data coming from the Raspberry Pi before being stored.

This product function requires an extremely specific subset of hardware requirements that are not required to capture standard audio. Similarly, only a tight band of specification ranges are compatible with the needs of the video capture portion of the product, as the video capture is also designed to mimic the end user's real field-of-view as the source event (e.g. a person talking to them) occurred. For this reason, the technical requirements herein are very specifically tailored to the unique needs of these product functions.

To summarize, the core product functions are as follows:

- Capture front-facing video.
- Capture binaural audio.
- Enable near-real-time (defined as 30 seconds maximum delay) streaming.
    - …On an Android device.
    - …On a web server.
- Enable later playback (non-real-time) of the recorded video.
- Enable later playback (non-real-time) of the recorded audio.

For product functions as provided to the customer, refer to **Artifact 1: Customer Agreement**, **Section "Overall Functionality"**.
For technical requirements needed to achieve these functions, refer to **Section 3.**
For technical requirements and stimulus sequences, refer to **Section 4**.

2.3 User Class

- **Military users** are the most important user class, as military standards are the most stringent when compared to consumer or academic requirements. While formal product certifications for military use are beyond the scope of this Capstone product, the product should be designed to meet appropriate military standards where applicable.
- **Consumer users** are the broadest targeted user group for this product. As consumer needs and wants are inherently varied, the product should be designed to meet the functionality promised by the core requirements, which enables application of the product to many consumer scenarios.
- **Academic users** may also take an interest in the product, which could be utilized in binaural audio research groups and similar projects. This user group is likely to have an interest in the experimental applications of the product, therefore requirements imposed by this user group are the most flexible of the user classes.
- **Hobby users** may be interested in using the Binaural Project for miscellaneous uses, such as for fun, as part of an interacting gaming experience, and more. The requirements of this subclass of

users are similarly flexible, as use cases and user stories are more general and less safety-critical than those of the military or business consumer.

- **Recording users** are defined as the person or persons wearing the Binaural device and its associated platform. For these users' ease-of-use and comfort, the product is designed to meet many regulations and industry-standards as detailed more extensively below, in **2.4 Operating Environment**. For instance, the product should be able to fit comfortably upon the user's head, as the camera is eyeglass-mounted. The Raspberry Pi platform and accompanying wires & phone should be able to fit inside a standard pocket. For full details, please refer to **2.4**.
- **Playback users** are defined as the person or persons who monitor streaming data and who view the streamed data in playback. These users may or may not be the same users as the recording users. For the sake of clearly outlining needed specifications, these user groups have been uniquely defined as two separate groups within this document. However, these user groups are not mutually exclusive. The playback users impose many software requirements upon the project. For instance, the app should be easy to log into; if the username and password do not work, a display message should be shown. Tiles on the side of the app should allow users to navigate through the app options. Streamed video should appear on the main page of the app and should be able to be easily saved to Firebase Storage with the press of a button. Details such as these are critical to the playback user group. For full detail of the requirements affecting this set of users, please also refer to **2.4 Operating Environment** for context.

2.4 Operating Environment

Physical Components:

- A glasses-mounted camera.
- A set of Binaural audio microphones
- A Raspberry Pi 3.
- An Android Essential Phone
- A Global Positioning System (GPS) tracker, which resides within the Android phone.
- An IMU orientation tracking device.

The potential operating environments for the product are many and varied—for instance, the rough terrain of war zone, versus the everyday monotony of the average American consumer. The Binaural Project should work for both customers. Therefore, it makes the most sense to produce the product as a standalone device with only a small number of additional contingencies. This aims to maximize potential use cases for the product, and to provide a uniform experience of comfortable wear and ease-of-use.

With variance in terrain in mind, the product is designed to be IP4 sweat and water resistance and IP5 dust resistant. The project is designed to withstand water, sweat, and dust levels as specified by those standards (IP4 and IP5 are specific levels).

The project is, as mentioned, designed to be worn. The camera is mounted on eye glasses; the binaural microphone connects to the hardware chassis and curls in the outwardly-exposed region of the ear (not in the inner ear). The hardware chassis—which contains the Raspberry Pi 3, the mounting board, the IMU unit, and interconnections—can be held. Further, it is expected that the Android phone can fit within a standard-sized pocket or phone case, as there is already-developed support for such devices (and such devices are typically developed by manufacturers to meet these standards).

As mentioned, the standard operating environment for the cell phone is the Android platform version 8.0 and above; support for iOS has not been developed, nor is it a goal of the project. The standard operating environment on the Raspberry Pi 3 platform is Raspbian Jessie, the Raspberry Pi operating system which is based on Linux.

App-side, Android Studio is used to develop the source code; Firebase Authentication, Storage, and Database are used as a pipeline for securely handling, processing, and using user data in a way that the information packets themselves are not exposed to the developers. The web server will be Unix-based and highly available. In its current iteration, the Binaural Project uses xampp and wamp localhost server tools to deploy over Apache.

For offline playback, the goal is to provide playback abilities regardless of user platform. Playback is offered in two ways; these ways are not mutually exclusive. Video can be viewed through one or both methods. First, playback can be achieved within the Android app. As mentioned, the project supports Android 8.0 and higher, though some features may incidentally work on earlier versions (support for earlier versions is a non-goal). Second, playback may also be achieved via the website. As websites work across operating systems—assuming operating system support for internet browsing—the website playback feature is platform-agnostic as it presents to users.

Regarding the development platforms, Windows 10 and a Linux virtual machine are used to develop the source code for the app and the website.

See **Section 2.7** for further assumptions and dependencies.

2.5 Design and Implementation Constraints

Military users may be subject to additional rules and regulations not applicable to the civilian community.

In keeping in mind, the varied needs of distinct customers, the Binaural Audio product is designed with a minimalistic approach; in this context, minimalistic is designed to mean that the hardware-software tandem is intentionally designed with maintainability, modularity, extensibility, and portability in mind. Complete test coverage is achieved through a combination of many user tests and integration tests; this helps to catch development bugs and other errors while the product is still in development, rather than post-release, when fixing bugs and other errors becomes much more expensive and time-intensive. Where possible, already-existing APIs are used. Using these methods ensures that work is not duplicated; further, using APIs helps to ensure the software is functional across many different platforms. Though this does add an inherent contingency, as addressed in **2.7 Assumptions and Contingencies**, the benefit-to-risk ratio is high: there is a lot of added portability and extensibility added compared to a relatively small amount of risk. Where APIs and open source software were used, they were used from highly reliable sources. For example, the Firebase tech stack has extremely high reliability and availability statistics. This helps to add the maximum benefit to the product while minimizing overall risk and complexity.

All users are constrained to the Android mobile platform as of V1.0 of the Binaural Audio project.

The goal of the project does not include open-sourcing the platform, therefore fundamental design decisions including the web server, database, language of development, and more are potentially limiting factors for the end user to consider.

2.6 User Documentation

A Test Plan, Design Document, User Manual, and Installation Manual will be produced in accordance with IEEE 829, 1016, and 1063 standards.

2.7 Assumptions and Dependencies

Full product use assumes the user:

- Possesses and can use a working Android device.
  - Assumes the Android device is compatible with Android 8.0+.

Most dependencies arise from the software side of the project, as the product is contingent on several third-party APIs and products.

Full product use is contingent upon:

- The use of web server and web hosting platforms; the project is inherently affected by their uptime/downtime, reliability, and other statistics.
- The use of the database service. In its present iteration, the Binaural Project is using Firebase Storage, Database, and Authentication.
- The real-time API PubNub and its support for the methods used.
  - Versioning must remain compatible with the requirements of the device.

The above contingencies are subject to some variations beyond the control of the creators of the Binaural Audio project, including but not limited to third-party downtime, latency, and security policies.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 [P0] Shall support streaming on an Android device in near-real-time.

3.1.1.1 [P0] Shall provide a login page for the app using Firebase Authentication.

3.1.1.2 [P0] Shall provide a button within the Android app to begin streaming in near-real-time.

3.1.1.3 [P0] Shall provide a button within the Android app to end streaming in near-real-time.

3.1.2 [P0] Shall support playback on an Android device (non-real-time).

3.1.2.1 [P0] Shall provide a login page for the app using Firebase Authentication.

3.1.2.2 [P0] Shall provide a button within the Android app to access the end user's stored files.

3.1.2.3 [P0] Shall provide a button within the Android app to begin playback.

3.1.2.4 [P0] Shall provide a button within the Android app to end playback.

3.1.3 [P0] Shall support streaming on a web server in near-real-time.

3.1.3.1 [P0] Shall provide a login page for the web server using Firebase Authentication.

3.1.3.2 [P0] Shall provide a button within the web server to begin streaming in near-real-time.

3.1.3.3 [P0] Shall provide a button within the web server to end streaming in near-real-time.

3.1.4 [P0] Shall support playback on a web server (non-real-time).

3.1.4.1 [P0] Shall provide a login page for the web server using Firebase Authentication.

3.1.4.2 [P0] Shall provide a button within the web server to access the end user's stored files.

3.1.4.3 [P0] Shall provide a button within the web server to begin playback.

3.1.4.4 [P0] Shall provide a button within the web server to end playback.

3.2 Hardware Interfaces

3.2.1 [P0] Shall Support streaming of captured data on an Android Device.

3.2.1.1 [P0] Shall support USB Type-C Connector port.

3.2.1.2 [P0] Shall support 802.11a/b/g/n/ac WiFi Standards.

3.2.1.3 [P0] Shall support MIMO WiFi.

3.2.1.4 [P0] Shall support a wireless connection.

3.2.1.5 [P0] Shall include volume and power button controls.

3.2.2 [P0] Shall capture binaural audio.

3.2.2.1 [P0] Shall have an omnidirectional polar pattern.

3.2.2.2 [P0] Shall be compatible with Linux operating system.

3.2.2.3 [P0] Shall support a USB output connection from sound card.

3.2.2.4 [P0] Shall support a 3.5 mm stereo input to the sound card.

3.2.2.5 [P0] Shall use RTMP Server

3.2.2.6 [P0] Shall support a 3.5 mm stereo auxiliary output from microphone.

3.2.3 [P0] Shall capture front-facing video.

3.2.3.1 [P0] Shall support the latest version of Raspbian operating system. At V2.0 this was Jessie.

3.2.3.2 [P0] Shall support dedicated Csi interface.

3.2.3.3 [P0] Shall support ribbon cable connection.

3.2.3.4 [P2] Shall possess a light-emitting diode (LED) indicator of whether the camera is recording.

3.2.4 [P2] Shall capture orientation data.

3.2.4.1 [P2] Shall support motion-triggered interrupt-signal generation for accelerometer.

3.2.4.2 [P2] Shall support motion-triggered interrupt-signal generation for gyroscope.

3.2.4.3 [P2] Shall support use of CircuitPython libraries.

3.2.4.4 [P2] Shall support Linux OS.

3.2.4.5 [P2] Shall support digital bidirectional I2C interface.

3.2.4.6 [P2] Shall support digital bidirectional UART interface.

3.2.5 [P0] Shall have power supply.

3.2.5.1 [P0] Shall be compatible with the Raspberry Pi 3.

3.2.5.2 [P0] Shall enable the Raspberry Pi to work for up to at least 8 hours.

3.2.5.3 [P0] Shall be controlled by dip-switch.

3.2.5.4 [P0] Shall have dual-Universal Serial Bus (USB) output.

3.2.6 [P0] Shall interface with the hardware chassis components via a Raspberry Pi microcontroller.

3.2.6.1 [P0] Shall support a 1.2GHz 64bit Central Processing Unit (CPU).

3.2.6.2 [P0] Shall possess wireless Local Access Network (LAN) on the board.

3.2.6.3 [P0] Shall possess 40-pin extended General Purpose Input Output (GPIO).

3.2.6.4 [P0] Shall possess 4 USB 2 ports.

3.2.6.5 [P0] Shall possess a composite video port.

3.2.6.6 [P0] Shall support full-size High Definition Multimedia Interface (HDMI).

3.2.6.7 [P0] Shall possess a Micro Secure Digital (micro SD) port.

3.2.6.8 [P0] Shall support upgraded switched Micro USB power source up to 2.5A

3.3 Software Interfaces

3.3.1 [P0] Shall support storage of data in a relational database.

3.3.1.1 [P1] Should not require the installation of database software.

3.3.1.2 [P1] Should support offloading of read traffic.

3.3.1.3 [P0] Shall be compatible with the web server platform.

3.3.2 [P0] Shall enable streaming on a web server.

3.3.2.1 [P1] Should support the customization of an IP address range.

3.3.2.2 [P1] Should support the creation of subnets.

3.3.2.3 [P1] Should support the configuration of route tables.

3.3.2.4 [P1] Should support the configuration of network gateways.

3.3.2.5 [P0] Shall be compatible with Firebase.

3.3.3 [P0] Shall enable streaming on an Android device.

3.3.3.1 [P0] Shall support Android versions as recent as 8.0.

3.3.3.2 [P1] Should support the most recent Android version (9.0 as of Binaural Audio V1.0)

3.3.3.3 [P0] Shall be compatible with Firebase.

3.3.3.4 [P0] Shall be capable of sending data to the web host.

3.4 Communications Interfaces

3.4.1 [P0] Shall communicate between stack layers.

3.4.1.1 [P0] Shall support HyperText Transfer Protocol (HTTP).

3.4.1.2 [P0] Shall support WebSockets.

3.4.1.3 [P2] Shall support Message Queueing Telemetry Transport (MQTT).

3.4.1.4 [P0] Shall be secured using Transport Layer Security (TLS) protocols.

3.4.1.5 [P0] Shall be secured using Secure Sockets Layer (SSL) protocols.

3.4.1.6 [P0] Shall possess network firewalls.

4. System Features

4.1 Shall capture near-real-time video

4.1.1 Description and Priority

The Binaural Audio device must be capable of capturing front-facing video in near-real-time. As noted previously, this is defined as a maximum of 30 seconds of delay. The priority for this feature is 0, the highest priority rating. The benefit of this feature is that a user may stream, in near-real-time, the video that the user wearing the device is experiencing. With regards to specific user stories, this feature proves valuable in military applications; high-level leaders can stream in near-real-time the in-the-field operations of military personnel and experience video like how it is being perceived by the device wearer themselves.

This undertaking is significant; specific technical requirements and specifications must be met to enable this feature, as it is a challenging problem to stream video in near-real-time.

Privacy and security concerns must be considered accordingly, and user data secured. For this reason, the Binaural Audio product team does not plan to collect personal information from users beyond that which is autonomously provided by Google Play Services.

4.1.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
   a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streaming inputs are sent to the Raspberry Pi 3 via direct connection.

4.1.3 Functional Requirements

REQ-1: [P0] Shall support the Audio Video Interleave (AVI) video format.
REQ-2: [P0] Shall support streaming in real-time (defined as a maximum delay of 30 seconds).
REQ-3: [P1] Should support playback of recorded stream after recording event (non-real-time).

4.2 Shall process the video in near-real-time.

4.2.1 Description and Priority

The Raspberry Pi 3 will receive digital signals transmitted from the camera in near-real time. The signals are transmitted via a wired connection between the camera and the Pi. The signals shall be processed according to the Raspberry Pi 3 board architecture. The Pi shall then transmit the stream to the Android device via a Local area or Hotspot network (and from the Android device, transmission to and from the web server and database is possible).

4.2.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
2. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
   a. Streaming inputs are captured by the hardware chassis.
3. Streaming inputs are sent to the Raspberry Pi 3 via direct connection.

4.2.3 Functional Requirements

REQ-1: [P0] Shall support the Audio Video Interleave (AVI) video format.
REQ-2: [P0] Shall support streaming in real-time (defined as a maximum delay of 30 seconds).
REQ-3: [P1] Should support playback of recorded stream after recording event (non-real-time).
REQ-4: [P0] Shall support full HD 1920×720p video.
REQ-5: [P0] Shall support framerate of at least 25 fps.
REQ-6: [P0] Shall combine the captured audio and video into a single stream.

4.3 Shall Capture near-real-time binaural audio.

4.3.1 Description and Priority

The Binaural Audio device must be capable of capturing binaural audio in near-real-time. As noted previously, this is defined as a maximum of 30 seconds of delay. The priority for this feature is 0, the highest priority rating. The benefit of this feature is that a user may stream, in near-real-time, the binaural audio that the user wearing the device is hearing. With regards to specific user stories, this feature proves valuable in military applications; high-level leaders can stream in near-real-time the in-the-field operations of military personnel and experience the audio like how it is being perceived by the device wearer themselves (the point of binaural audio).

This undertaking is significant; specific technical requirements and specifications must be met to enable this feature, as it is a challenging technical problem to stream audio in near-real-time.

Privacy and security concerns must be considered accordingly, and user data secured. For this reason, the Binaural Audio product team does not plan to collect personal information from users beyond that which is autonomously provided by Google Play Services.

4.3.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streaming inputs are sent to the Raspberry Pi 3 via direct connection.

4.3.3 Functional Requirements

REQ-1: [P0] Shall have an omnidirectional polar pattern when both microphones are functioning.
REQ-2: [P0] Shall not occlude the ear drum.

4.4 Shall process the near-real-time binaural audio

4.4.1 Description and Priority

The Raspberry Pi 3 will receive digital sound signals transmitted from the binaural microphone in near-real time. The signals are transmitted via a wired connection between the camera and the Pi. The digital sound signals shall be processed according to the Raspberry Pi 3 board architecture. The Pi shall then transmit the stream to the Android device via a Local area or Hotspot network (and from the Android device, transmission to and from the web server and database is possible).

4.4.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streaming inputs are sent to the Raspberry Pi 3 via direct connection.

4.4.3 Functional Requirements

REQ-1: [P0] Shall sample the audio at a rate of 44.1 kHz.

REQ-2: [P0] Shall combine the captured audio and video into a single stream.

4.5 Shall capture orientation data in near-real-time

4.5.1 Description and Priority

The Raspberry Pi 3 will receive digital signals transmitted from the IMU in near-real time. The signals are transmitted via a wired connection between the IMU board and the Pi. The signals shall be processed according to the Raspberry Pi 3 board architecture using CircuitPython. The Pi shall then transmit the data to the Android device via a Local area or Hotspot network (and from the Android device, transmission to and from the web server and database is possible).

4.5.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streaming inputs are sent to the Raspberry Pi 3 via direct connection.

4.5.3 Functional Requirements

REQ-1: [P2] Shall contain a 3-axis magnetometer

REQ-2: [P2] Shall contain a 3-axis gyroscope

REQ-3: [P2] Shall contain a 3-axis accelerometer

REQ-4: [P2] Shall output fused sensor data.

a. [P2] Shall output Euler Angles.
b. [P2] Shall output Rotation vector.
c. [P2] Shall output Linear acceleration.
d. [P2] Shall output Heading.

4.6 Shall time synchronize all received data.

4.6.1 Description and Priority

The Raspberry Pi 3 will be receiving at least three streams of data via wired or direct pin connection: video, audio, and orientation data streams. The data can be synchronized according to UTC time. A program shall be constructed in Raspbian to ensure this synchronization of data streams before further transmission to other downstream pipelines.

4.6.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3 according to UTC timestamps.
4. Synchronized streams are transmitted to the Android device.

4.6.3 Functional Requirements

REQ-1: [P0] Shall process information as it is received.
REQ-2: [P0] Shall process all incoming data stream concurrently.

4.7 Shall transfer data from microcontroller to Android Device.

4.7.1 Description and Priority

Data transfer between the Raspberry Pi 3 and the Android device shall be completed over a local area or Hotspot Network. WiFi transfer is another available option depending on stream sizes and the preference of the end user. Once the streams are synchronized according to **4.6**, data transfer to the Android device enables further transfer from the Android device to the database and/or to the web server.

4.7.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3.
4. Synchronized streams are transmitted to the Android device over a local area network.

4.7.3 Functional Requirements

REQ-1: [P0] Shall ensure wireless connection has a data rate of at least 11 Mbps.
REQ-2: [P0] Shall stream the audio at 40 Kbps.
REQ-3: [P0] Shall have video bitrate ranging from 300 Kbps to 3 Mbps.
REQ-4: [P0] Shall support RTMP server
REQ-5: [P0] Shall support synchronous replication of data.

4.8 Shall transfer data from Android Device to Web Server.

4.8.1 Description and Priority

Data transfer between the Android device and the web server is enabled over WiFi. The transfer shall be protected according to Firebase security protocols and firewalls. Transmission between the Android device, web server, and database is available on a many-to-many basis.

4.8.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3.
4. Synchronized streams are transmitted to the Android device.
5. Android device transmits streams to the web server over WiFi.

4.8.3 Functional Requirements

REQ-1: [P0] Shall support synchronous replication of data.
REQ-2: [P0] Shall not require infrastructure provisioning.
REQ-3: [P0] Shall include automated backups.
REQ-4: [P0] Shall include database snapshots.
REQ-5: [P0] Shall include automatic host replacement.
REQ-6: [P0] Shall not require the maintenance of database software.

4.9 Shall enable streaming on an Android device in near-real-time

4.9.1 Description and Priority

The Android device will be capable of streaming audio and/or video in near-real-time. The data is transmitted in accordance to **4.7**. After transmission of the stream(s) is complete, the Android device will be capable of playing the stream(s) within the Binaural Audio application. The app will feature buttons for beginning and ending the stream, as well as accessing the user's stored streaming data for later playback.

4.9.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3.
4. Synchronized streams are transmitted to the Android device.
5. End user presses button within the Android app to begin streaming in near-real-time.

4.9.3 Functional Requirements

REQ-1: [P0] Shall ensure wireless connection has a data rate of at least 11 Mbps.
REQ-2: [P0] Shall stream the audio at 40 Kbps.
REQ-3: [P0] Shall have video bitrate ranging from 300 Kbps to 3 Mbps.

REQ-4: [P0] Shall support RTMP server
REQ-5: [P0] Shall support synchronous replication of data.
REQ-6: [P0] Shall not require infrastructure provisioning.
REQ-7: [P0] Shall include automated backups.
REQ-8: [P0] Shall include database snapshots.
REQ-9: [P0] Shall include automatic host replacement.
REQ-10: [P0] Shall not require the maintenance of database software.

4.10 Shall enable playback on a web server in near-real-time.

4.10.1 Description and Priority

The web server will be capable of streaming audio and/or video in near-real-time. The data is transmitted in accordance to **4.9**. After transmission of the stream(s) is complete, the web server will be capable of playing the stream(s) within the Binaural Audio application. The web server user interface will feature buttons for beginning and ending the stream, as well as accessing the user's stored streaming data for later playback.

4.10.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
   a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3.
4. Synchronized streams are transmitted to the Android device.
5. Android device transmits streams to the web server over WiFi.
6. End user presses button within the user interface to begin streaming in near-real-time.

4.10.3 Functional Requirements

REQ-1: [P0] Shall support synchronous replication of data.
REQ-2: [P0] Shall not require infrastructure provisioning.
REQ-3: [P0] Shall include automated backups.
REQ-4: [P0] Shall include database snapshots.
REQ-5: [P0] Shall include automatic host replacement.
REQ-6: [P0] Shall not require the maintenance of database software.

4.11 Shall enable later playback on an Android device.

4.11.1 Description and Priority

The Android device will be capable of retrieving stored audio and video streams from the relational database. Firebase was chosen as the database provider due to the product's low cost, high reliability, and high availability. This data fetch instruction shall be completed over WiFi, and in accordance with the AWS transfer security protocols. After access of the stream(s) is complete, the Android device will be capable of playing the stream(s) within the Binaural Audio application. The app will feature buttons for beginning and ending the stream, as well as accessing the user's stored streaming data.

4.11.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3.
4. Synchronized streams are transmitted to the Android device.
5. End user presses button within the Android app to search for files.
6. End user selects the saved files they want to stream as playback.
7. End user presses button to begin streaming playback.

4.11.3 Functional Requirements

REQ-1: [P0] Shall support synchronous replication of data.
REQ-2: [P0] Shall not require infrastructure provisioning.
REQ-3: [P0] Shall include automated backups.
REQ-4: [P0] Shall include database snapshots.
REQ-5: [P0] Shall include automatic host replacement.
REQ-6: [P0] Shall not require the maintenance of database software.

4.12 Shall enable later playback on a web server.

4.12.1 Description and Priority

The Web Server will be capable of retrieving stored audio and video streams from the relational database. The web server provider must be chosen as the web server provider due to the product's low cost, high reliability, and high availability. This data fetch instruction shall be completed over WiFi, and in accordance with standard transfer security protocols. After access of the stream(s) is complete, the web server will be capable of playing the stream(s). The website will feature buttons for beginning and ending the stream, as well as accessing the user's stored streaming data.

4.12.2 Stimulus/Response Sequences

1. End user flips the physical switch on the side of the Raspberry Pi chassis to begin streaming capture.
    a. Flipping the physical switch shall also initiate the capture of binaural audio via the binaural microphone.
2. Streaming inputs are captured by the hardware chassis.
3. Streams are synchronized by the Raspberry Pi 3.
4. Synchronized streams are transmitted to the Android device.
5. Android device saves streams to the database.
6. End user presses button within the user interface to search for files.
7. End user selects the saved files they want to stream as playback.
8. End user presses button to begin streaming playback.

4.12.3 Functional Requirements

REQ-1: [P0] Shall support synchronous replication of data.
REQ-2: [P0] Shall not require infrastructure provisioning.
REQ-3: [P0] Shall include automated backups.
REQ-4: [P0] Shall include database snapshots.
REQ-5: [P0] Shall include automatic host replacement.
REQ-6: [P0] Shall not require the maintenance of database software.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 IMU with 9 Degree of Freedom (DOF)

5.1.1.1 [P2] Shall have an accelerometer accuracy of less than 12 mg/LSB.

5.1.1.2 [P2] Shall have an accelerometer range of ±16 multiples of gravity.

5.1.1.3 [P2] Shall have a gyroscopic measurement range of ±2000 degrees per second (DPS).

5.1.1.4 [P2] Shall have a gyroscopic sensitivity of less than ±3 percent.

5.1.1.5 P2] Shall have an IO voltage of range from 2.4 V – 3.6 V

5.1.1.6 [P2] Shall have accelerometer low-pass filter bandwidth from 1 kHz - < 8 Hz.

5.1.1.7 [P2] Shall support acceleration ranges from ±2g to ±16g.

5.1.1.8 [P2] Shall support switchable gyroscope ranges from ±125˚/s to ±2000 ˚/s.

5.1.1.9 [P2] Shall have an x-, y-axis magnetic field range of ±1300 µT.

5.1.1.10 [P2] Shall have a z-axis magnetic field range of ±2500 µT.

5.1.1.11 [P2] Shall have a magnetic field resolution of ~0.3 µT.

5.1.1.12 [P2] Shall support streaming in real-time (defined as a maximum delay of 30 seconds).

5.1.2 GPS Module in Phone

5.1.2.1 [P0] Shall operate inside with no clear line of sight (LOS) using A-GPS.

5.1.2.2 [P0] Shall operate outside with LOS using GPS.

5.1.2.3 [P0] Shall operate outside with LOS using GLONASS

5.1.2.4 [P0] Shall have an accuracy of less than 6 meters.

5.1.3 Android Device

5.1.3.1 [P0] Shall include a battery of 2770 mAh or greater.

5.1.3.2 [P0] Shall include internal storage of 64 GB or greater.

5.1.4 Binaural Microphones

5.1.4.1 [P0] Shall have a microphone audio input range of 0 – 1.25 Vrms.

5.1.4.2 [P0] Shall have an open circuit sensitivity of -32 dB SPL (Sound Pressure Level Referenced to 20 μPa).

5.1.4.3 [P0] Shall have a Dynamic Range of 95 dB.

5.1.4.4 [P0] Shall have a Frequency Range from 20 – 20,000 Hz.

5.1.4.5 [P0] Shall produce 40 dB SPL maximum self-noise.

5.1.4.6 [P0] Shall have an A/D conversion resolution of 16 bits.

5.1.4.7 [P0] Shall have a sample rate of 44.1 kHz.

5.1.4.8 [P0] Shall have a signal to noise ratio of 75 dB.

5.1.4.9 [P0] Shall have an open circuit sensitivity of -32 dB.

5.1.4.10 [P0] Shall have a maximum input level of 130 dB.

5.1.5 Facing Video

5.1.5.1 [P0] Shall support 720p or greater High Definition (HD) Video.

5.1.5.2 [P0] Shall enable video streaming with a maximum delay of 30 seconds.

5.1.5.3 [P0] Shall possess a frame rate of at least 30 frames per second (fps) minimum.

5.1.6 Power Supply

5.1.6.1 [P0] Shall have a battery capacity of at least 3660mAh

5.1.6.2 [P0] Shall support output currents up to 1.8A.

5.1.6.3 [P0] Shall support output voltages of 5.1 ± 0.1V.

5.1.6.4 [P0] Shall support standard charging current of 1.0A.

5.1.6.5 [P0] Shall support standard charging voltage of 5.0V.

5.1.6.6 [P0] Shall support cut-off voltage of fully charged at 4.18V – 4.2V.

5.1.7 Web Server

5.1.7.1 [P0] Shall support IPv4.

5.1.7.2 [P0] Shall support IPv6.

5.2 Safety Requirements

5.2.1.1 [P0] Hardware chassis shall comply with classification code HQY, title 21 CFR 886.5850.

5.2.1.2 [P0] Shall comply with AdaFruit safety recommendations.

5.3 Security Requirements

5.3.1.1 [P0] Shall store collected video separate from any identifying information.

5.3.1.2 [P0] Shall store collected audio separate from any identifying personal information.

5.3.1.3 [P0] Shall not make poll end user for identifying personal information.

5.3.1.4 [P0] Shall abide by the Google Play Store Privacy Policy for any personal information provided by Google as a result of Android downloads of the application.

5.3.1.5 [P0] Shall ensure web server is secured using Transport Layer Security (TLS) protocols.

5.3.1.6 [P0] Shall ensure web server is secured using Secure Sockets Layer (SSL) protocols.

5.3.1.7 [P0] Shall ensure web server possess network firewalls.

5.3.1.8 [P0] Shall ensure database web access is secured using TLS protocols.

5.3.1.9 [P0] Shall ensure database web access is secured using SSL protocols.

5.3.1.10 [P0] Shall ensure database web access possess network firewalls.

5.4 Hardware Quality Attributes

5.4.1.1 [P0] Shall collectively weight less than 500 grams for sensors on hardware chassis.

5.5 Software Quality Attributes

5.5.1.1 [P0] Shall use portable software libraries.

5.5.1.2 [P0] Shall use open source software libraries.

5.5.1.3 [P0] Shall use Application Programming Interfaces (APIs) where possible, increasing many key quality measures such as portability and maintainability.

5.5.1.4 [P0] Shall match Firebase availability metrics.

5.5.1.5 [P0] Shall document software classes 1:1 for maintainability.

5.5.1.6 [P0] Shall test requirements 1:1 at a minimum.

5.5.1.7 [P1] Shall be water resistant according to IP4.

5.5.1.8 [P1] Shall be sweat resistant according to IP4.

5.6 Business Rules

Differences in end user characteristics do not make any impact on features available to end users as of version 1.0 of Binaural Audio.

6. Other Requirements

The system must comply with the requirement of the Florida Accessibility of Information and Technology Act (see sections 282.601-282.606, Florida Statutes) and the Standards Applicable to Electronic and Information Technology as set forth in Rule 60EE-1.002, Florida

Appendix A: Glossary

Application Programming Interface (API).
Audio Video Interleave (AVI).
Bluetooth Low Energy (BLE).
Camera Serial Interface (CSI).
Central Processing Unit (CPU).
Display Serial Interface (DSI).
Degrees of Freedom (DOF)
Degrees per Second (DPS)
Frames Per Second (FPS
General Purpose Input Output (GPIO).
Global Positioning System (GPS).
High Definition (HD).
High Definition Multimedia Interface (HDMI).
HyperText Transfer Protocol (HTTP).
Inter-Integrated Circuit (I^2C)
Land Grid Array (LGA)
Light-Emitting Diode (LED).
Line of Sight (LOS)
Local Access Network (LAN).
Message Queueing Telemetry Transport (MQTT).
Micro Secure Digital (micro SD).
Pulse-Per-Second (PPS).
Random Access Memory (RAM).
Real Time Clock (RTC).
Secure Sockets Layer (SSL).
Sound Pressure Level (SPL).
Transport Layer Security (TLS).
Universal Serial Bus (USB).
Universal Asynchronous Receiver/Transmitter (UART).

Appendix B: Analysis Models

Refer to **Figure 1**.