Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2023

Team Majestic
RFID/Smart Stethoscope

Michael Allen
Jaime Barajas
Efrain Morales Loya
Chau Nguyen
Khoi Tran

Revision History

Revision	Date	Author(s)	Description
0.1	07.21.2023	KT, EM	document creation
0.2	07.25.2023	KT, CN, MA, EM	complete draft
0.3	07.25.2023	CN, KT, MA	Review draft

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The RFID Smart Stethoscope is a novel educational tool designed to enhance diagnostic learning in healthcare. Using programmable RFID chips embedded in a wearable shirt, it simulates various auscultation sounds. The product is designed for medical students, healthcare professionals, and educators to improve their diagnostic skills. Key requirements include a user-friendly interface, a compatible device for reading RFID chips, and a specialized shirt to hold the RFID chips. The RFID Smart Stethoscope is an innovative blend of technology and education, serving the evolving needs of healthcare training.

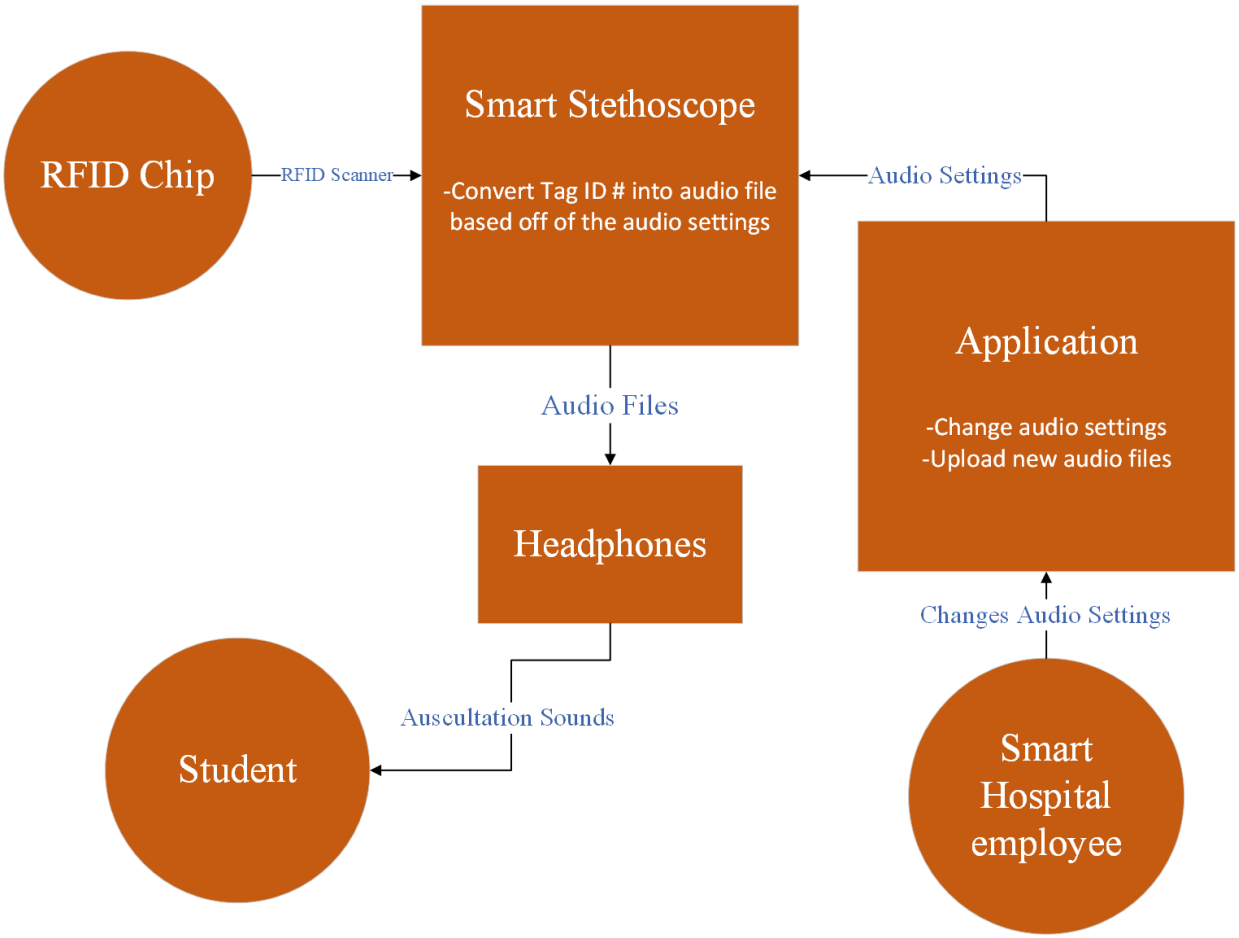

Figure 1: RFID Smart Stethoscope Conceptual Drawing

2 System Overview

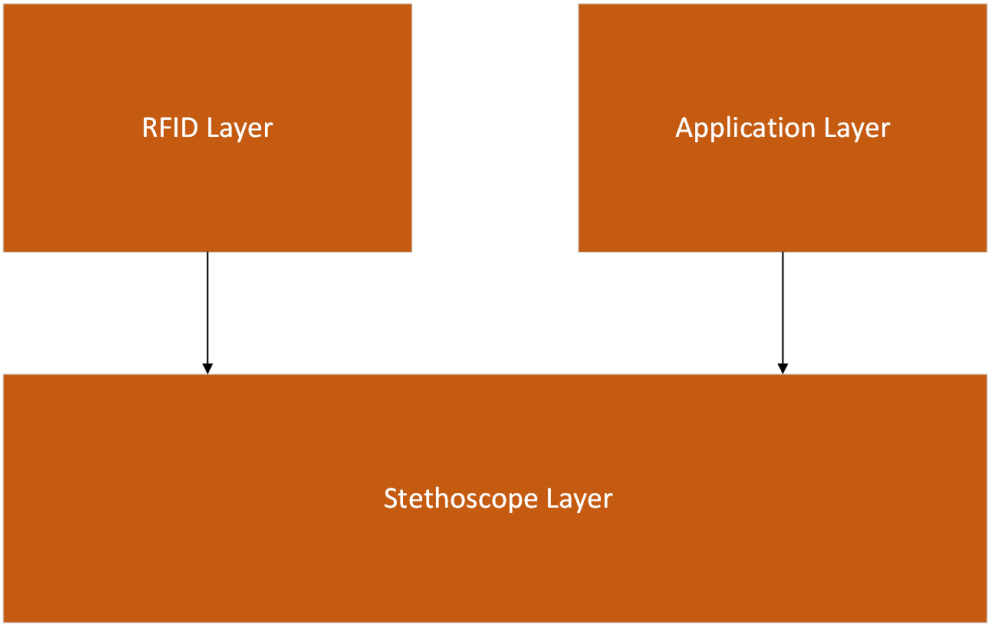

Figure 2: A simple architectural layer diagram

2.1 RFID Layer

The RFID layer is the smallest layer in this project. It is responsible for handling all interactions with the RFID tags. The purpose of this layer is to contain the 15 different RIFD tags, and have then each return their unique ID numbers when they are scanned by the Smart Stethoscope. The critical interface here is the RFID tags whose unique ID numbers inform the Smart Stethoscope as to which part of the body was just scanned.

2.2 Stethoscope Layer

The Stethoscope layer contains the RIFD scanner, the computer/database, and the headphones. It receives all of the sound settings and files from the Application layer. It also receives the RFID tag ID number from the RFID layer. The sound settings are stored within the sound database, and when it receives the tag ID number from the RIFD layer it uses it to determine which sounds to output to the headphones. The headphones receives the sound data from the database, and then plays it so that user can hear the programmed sounds. This layer interacts primarily with the Application layer which it receives and updates its sound database.

2.3 Application Layer

The Application layer is used to program the sounds that each part of the body will play. It allows for a Smart Hospital employee to upload audio files and program them to a specific part of the body. This is sent to the Stethoscope layer, and used to populate the sound database. The sound settings can receives update information directly from the user or from pre-saved user presets. This layer interacts primarily with the Stethoscope layer, which it updates its sound database.

3 Subsystem Definitions & Data Flow

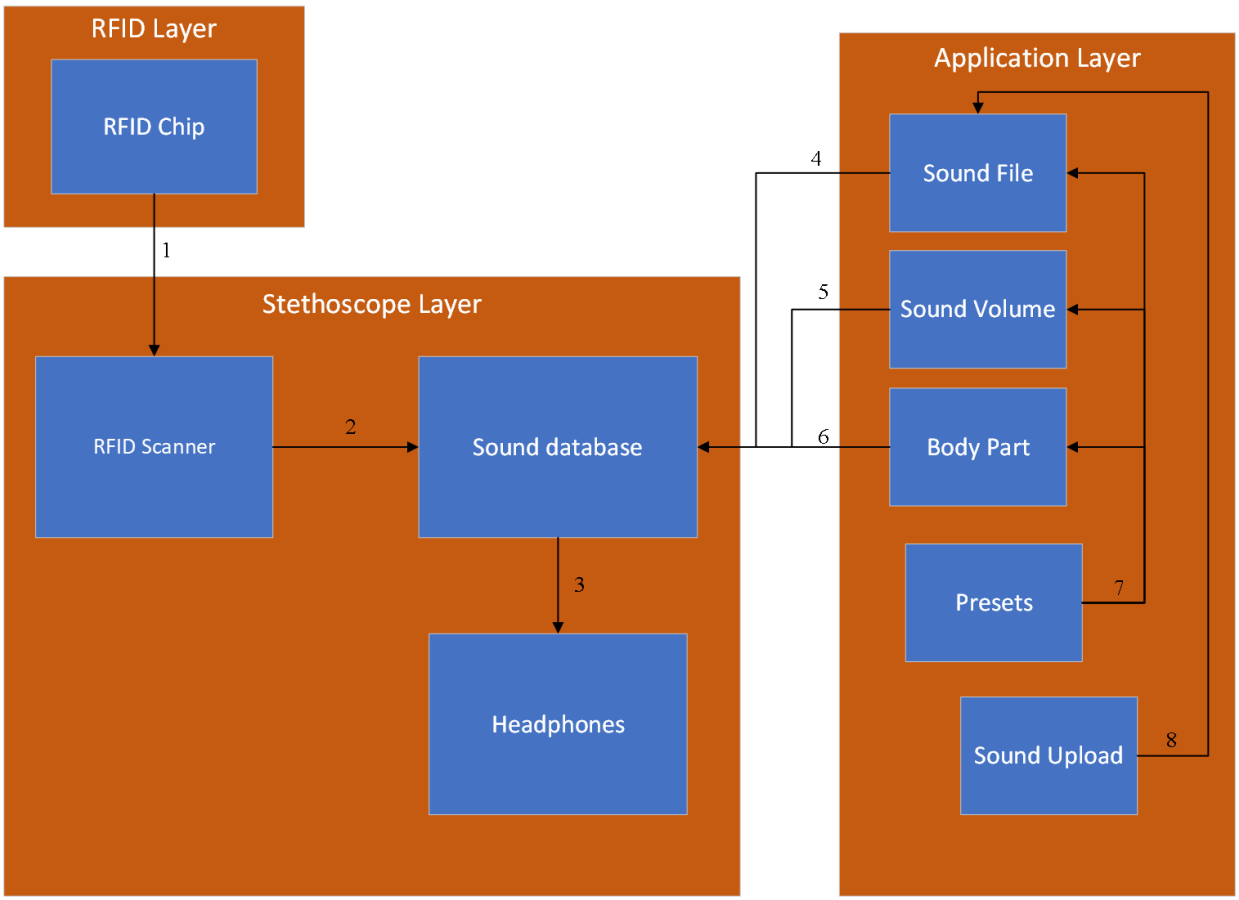

Figure 3: A simple data flow diagram

4 RFID Layer Subsystems

In this section, RFID Layer it contains all of the RFID tags on the shirt as well as all of the unique ID numbers that each of the RIFD tags has.

4.1 RFID Chip

This subsystem contains the RIFD tags themselves each which contains a unique ID number which is used to determine which RFID tag is scanned, and thus which part of the body the stethoscope is on.

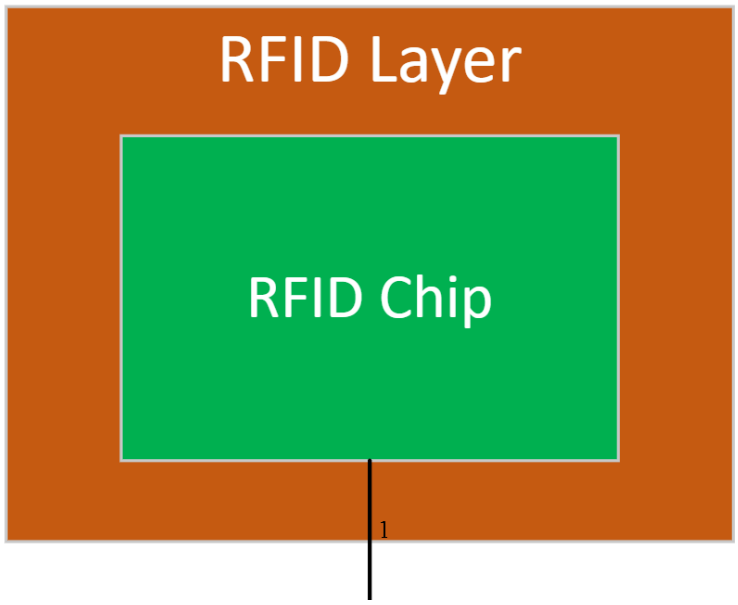

Figure 4: RFID chip subsystem

4.1.1 Assumptions

The RFID tag ID numbers are programmed into the smart stethoscope already.

4.1.2 Responsibilities

The primary responsibility of this subsystem is to transmit its ID number when scanned so that the Stethoscope knows which part of the body it is scanning.

4.1.3 Subsystem Interfaces

Table 2: RFID Chip interfaces

ID	Description	Inputs	Outputs
#01	Sends ID to RFID Scanner	N/A	Tag ID number

5 Stethoscope Layer Subsystems

In this section we look into all the subsystems contained within the Smart Stethoscope itself. From the RFID scanner, or the Sound databse, and finally the users Headphones. It controls the functionality of the project as a whole. This Layer takes the RFID tag ID number and uses it to determine what sound must be played and then plays it immediately to the user.

5.1 RFID Scanner

The RFID Scanner subsytem, is in charge of interfacing with RFID layer. The RFID scanner retrieves the ID number of the tags it scans and sends them to the sound database for further use.

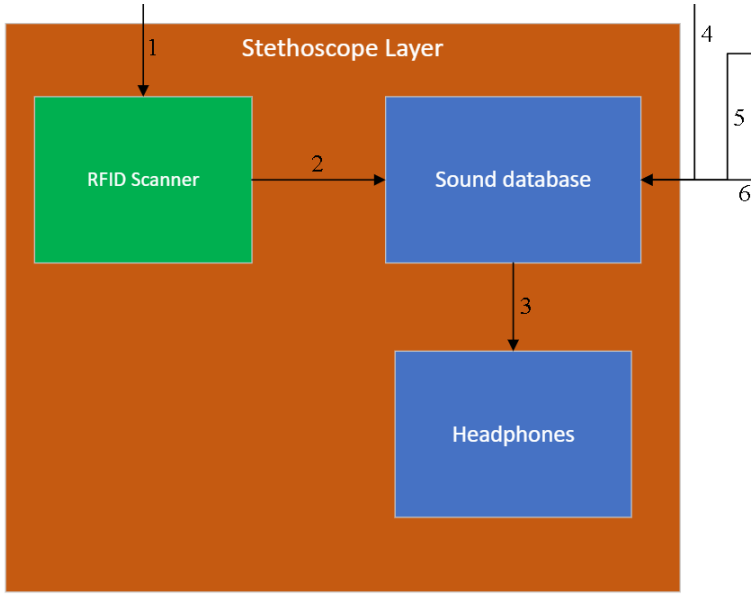

Figure 5: RFID Scanner description diagram

5.1.1 Assumptions

All RFID tags scanned will be a compatible frequency with the RFID reader themselves.

5.1.2 Responsibilities

Responsible for scanning the RFID tags on the shirt of the patient and transmitting its ID to the database for further use.

5.1.3 Subsystem Interfaces

Table 3: RFID Scanner interfaces

ID	Description	Inputs	Outputs
#01	Scanning the RFID tag	Tag ID number	N/A
#02	Sending data to the database	N/A	Tag ID number

5.2 SOUND DATABASE

This is the most pivotal subsystem within the Stethoscope it is the brain of the system. It controls which sounds are played based on its series of inputs that it recieves.

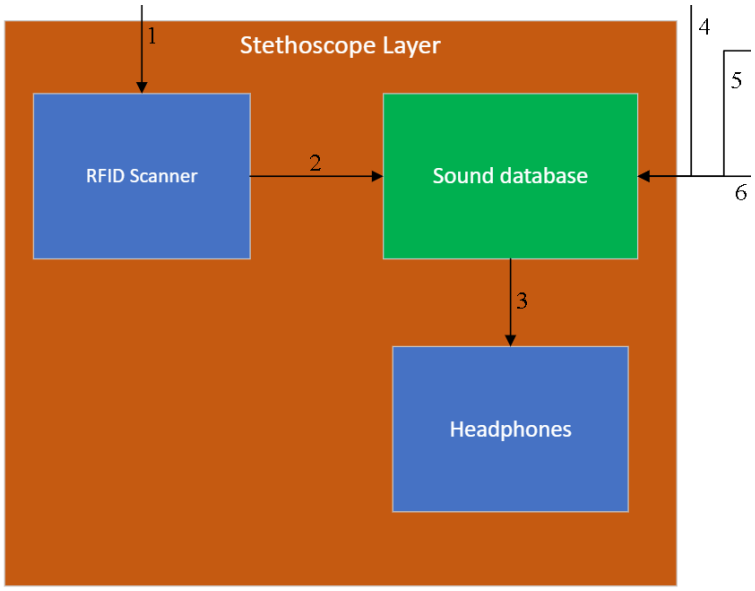

Figure 6: Sound database description diagram

5.2.1 ASSUMPTIONS

Sound files received from Application layer are in a valid format. Only one sound file is set per body part. RFID tag ID number is in a valid format for determining if it is a tag within our system.

5.2.2 RESPONSIBILITIES

Receives the ID number of the RFID tag from the RFID scanner, and determines if it is a tag within our system if not it doesn't play any sound. If it is a valid tag ID it used that ID to determine which body part the scanner is on, and play the corresponding sound at the correct volume determined by its input from the application layer.

5.2.3 SUBSYSTEM INTERFACES

Table 4: Sound Database interfaces

ID	Description	Inputs	Outputs
#04, #05, #06	Application writes the sound settings	Sound File Sound Volume Body Part	N/A
#02	Scanner sends ID number of tag scanned	Tag ID number	N/A
#03	Correct sound file sent to headphones	N/A	Audio file

5.3 HEADPHONES

This subsystem directly sends the feedback from the whole system back to the user. It the final step in the project, it receives the audio to play, and allows for the user to hear it.

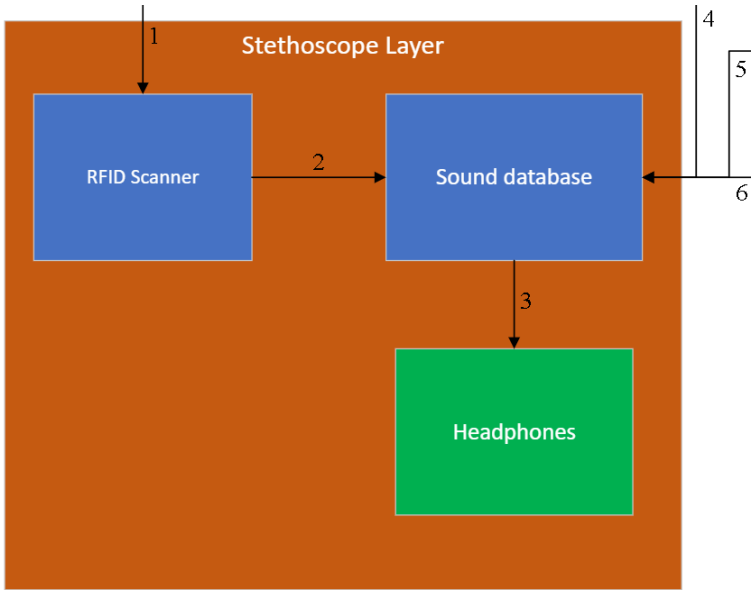

Figure 7: Headphones description diagram

5.3.1 ASSUMPTIONS

Audio file is in a valid format to be played.

5.3.2 RESPONSIBILITIES

Responsible for playing the sound at a reasonable quality so the the user can hear the correct auscultation sounds and determine what is wrong with the sim patient.

5.3.3 SUBSYSTEM INTERFACES

Table 5: Headphones interfaces

ID	Description	Inputs	Outputs
#03	Receives audio to play	Audio files	Audio

6 Application Layer Subsystems

In this section, the application layer's subsystems are meticulously dissected. Each subsystem is individually discussed in detail, highlighting its specific functions and responsibilities within the broader framework. Special considerations and trade-offs instrumental in shaping the chosen architectural approach are also explored. This rigorous analysis is instrumental in understanding the details and intricacy of the entire subsystem.

6.1 Sound Files

This subsystem handles the sound files of the application. It is responsible for storing and managing all audio files that will be used within the application.

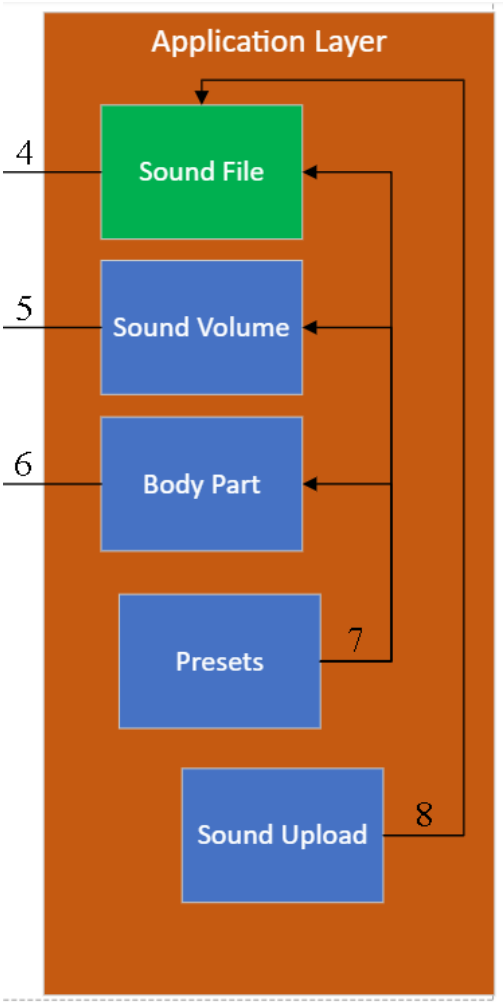

Figure 8: Sound Files description diagram

6.1.1 Assumptions

This subsystem assumes that all sound files are in a compatible format and have been properly encoded.

6.1.2 Responsibilities

The Sound Files subsystem is responsible for storing audio files, writing them to the Stethoscope, and deleting them if they are no longer required.

6.1.3 Subsystem Interfaces

Table 6: Sound Files interfaces

ID	Description	Inputs	Outputs
#08	Sound file input from Sound Upload subsystem	Sound file	N/A
#07	Sound file updated from saved Preset	Sound file setting	N/A
#04	Write to Sound database	N/A	Sound file

6.2 Sound Volume

This subsystem controls the volume level for each sound file. It ensures that each sound is played at an appropriate level, according to the user's preferences.

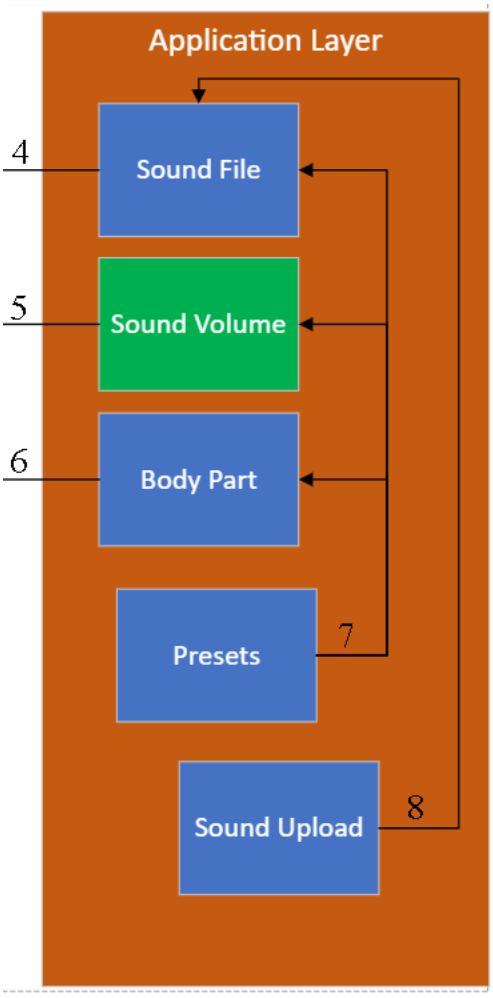

Figure 9: Sound Volume description diagram

6.2.1 Assumptions

This subsystem assumes that the volume level for each sound file is adjustable and that the user has the ability to change the volume at will.

6.2.2 Responsibilities

The Sound Volume subsystem is responsible for adjusting the volume level of each sound file and storing the volume level set by the user.

6.2.3 Subsystem Interfaces

Table 7: Sound Volume interfaces

ID	Description	Inputs	Outputs
#07	Volume level input from Presets subsystem	Volume level	N/A
#08	Sound file input from Sound Upload subsystem	Sound file	N/A
#05	Volume adjustment output to Sound Database	N/A	Volume level

6.3 Body Part

This subsection provides an in-depth examination of the "Body Part" subsystem within the application layer. Serving a pivotal role in the overall system architecture as it is the system that allows you to control which sounds plays on a specific part of the body.

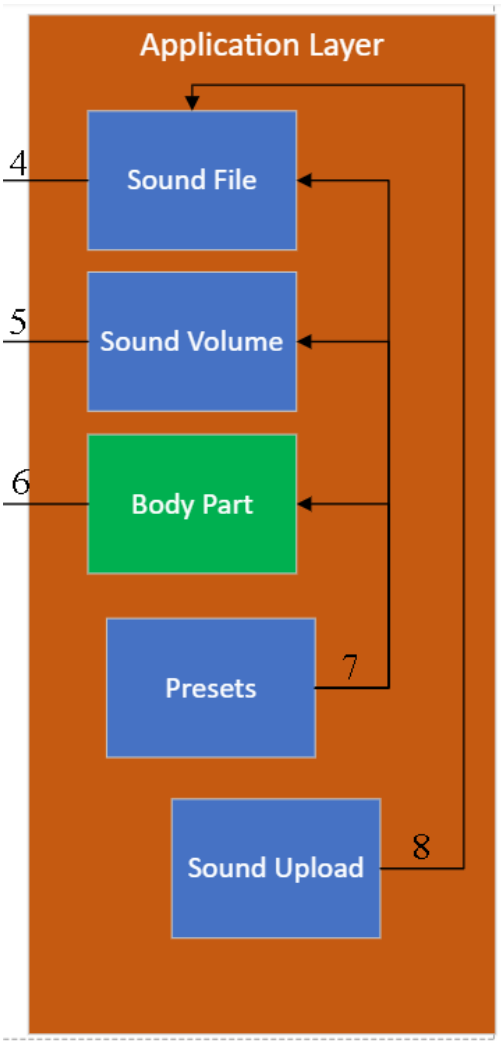

Figure 10: Body Part description diagram

6.3.1 Assumptions

Only one body part can be selected at a time, and only one sound can be programmed to a body part at a time.

6.3.2 Responsibilities

This subsystem is responsible for selecting which one of the 14 body parts that can be scanned on the RFID shirt play the sound currently set in the sound file subsystem.

6.3.3 Subsystem Interfaces

Table 8: Body part interfaces

ID	Description	Inputs	Outputs
#06	Write to selected body part in sound database	N/A	Body Part Data
#07	Body part setting updated from saved Preset	Body part setting	N/A

6.4 Presets

The Presets subsystem plays a critical role in managing and implementing the predefined configurations in our system. The architectural block diagram shown below provides a clear overview of the data flows involving the Presets subsystem and its interconnected subsystems.

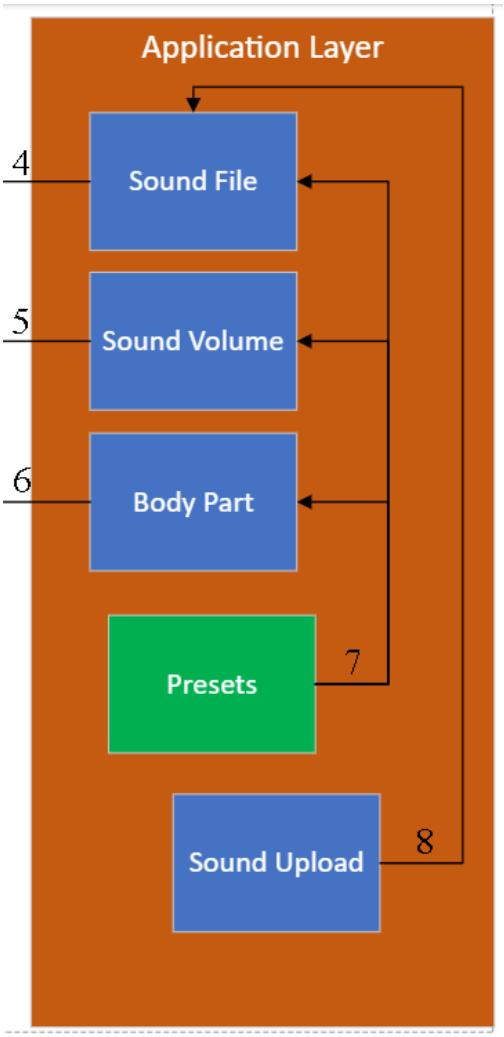

Figure 11: Presets description diagram

6.4.1 Assumptions

Presets will be previously saved valid settings, and all files needed will be present on system.

6.4.2 Responsibilities

Presets will allow for repeated reuse of certain scenarios for repeated learning. This will save the smart hospital staff time fully reprogramming the scenario.

6.4.3 Subsystem Interfaces

Table 9: Presets interfaces

ID	Description	Inputs	Outputs
#07	Update current settings to saved preset	Preset chosen by the user	Sound File setting Sound Volume Selected body part

6.5 Sound Upload

The Sound Upload subsystem is an integral part of the application layer, facilitating the uploading and management of sound files within the system. The architectural block diagram below depicts the subsystem's data flow dynamics, presenting a clear view of how it communicates and interacts with other subsystems.

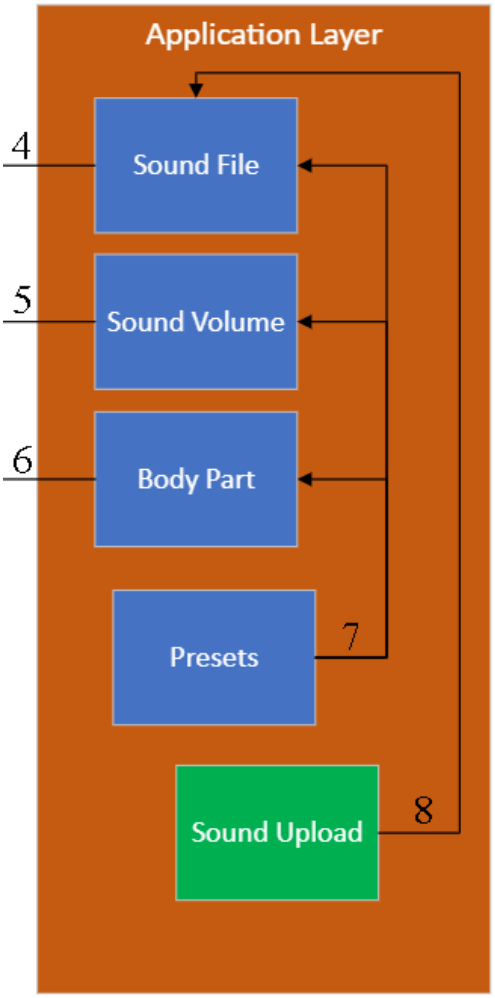

Figure 12: Sound Upload description diagram

6.5.1 Assumptions

The development and implementation of the Sound Upload subsystem is predicated on the following assumptions:
1. The subsystem will receive sound files that adhere to the specified format and size limitations.
2. Interactions with other subsystems and interfaces will conform to the defined communication and data exchange protocols.

6.5.2 Responsibilities

The Sound Upload subsystem has several critical responsibilities, including:
1. Facilitating the uploading of sound files from the user.
2. Conducting preliminary checks on the sound files to ensure they comply with the system's requirements.
3. Interacting with the storage subsystem to ensure successful saving of the uploaded sound files.
4. Communicating upload status and potential issues to the user or relevant subsystems.

6.5.3 SUBSYSTEM INTERFACES

Table 10: Sound upload interfaces

ID	Description	Inputs	Outputs
#08	Allow user to upload a new sound from the PC	Sound File	Send File to sound file settings

REFERENCES