|  |
| --- |
| HCC2, HCCE2, BSHC2, BSHCE2, BSHBIS2, BSHBISE2 |
| Requirements Specification (RS) |

|  |
| --- |
|  |

|  |
| --- |
|  |

Colin Allen,

Keith Feeney,

Patrick Lawlor,

Fearghal McMorrow,

Cedric Vecchionacce

(Group 14)

17 February 2017

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 14/10/2005 | 1 | Create | AB | X | X |
| 21/10/05 | 2 | Update | CD |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Arghir Moldovan | Lecturer |  |
| Eamon Nolan | Lecturer |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction 4](#_Toc316977392)

[1.1 Purpose 4](#_Toc316977393)

[1.2 Project Scope 4](#_Toc316977394)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc316977395)

[2 User Requirements Definition 4](#_Toc316977396)

[3 Requirements Specification 4](#_Toc316977397)

[3.1 Functional requirements 4](#_Toc316977398)

[3.1.1 Use Case Diagram 5](#_Toc316977399)

[3.1.2 Requirement 1 <name of requirement in a few words> 5](#_Toc316977400)

[3.1.3 Requirement 2 <name of requirement in a few words> 6](#_Toc316977401)

[3.2 Non-Functional Requirements 7](#_Toc316977402)

[3.2.1 Performance/Response time requirement 8](#_Toc316977403)

[3.2.2 Availability requirement 8](#_Toc316977404)

[3.2.3 Recover requirement 8](#_Toc316977405)

[3.2.4 Robustness requirement 8](#_Toc316977406)

[3.2.5 Security requirement 8](#_Toc316977407)

[3.2.6 Reliability requirement 8](#_Toc316977408)

[3.2.7 Maintainability requirement 8](#_Toc316977409)

[3.2.8 Portability requirement 8](#_Toc316977410)

[3.2.9 Extendibility requirement 8](#_Toc316977411)

[3.2.10 Reusability requirement 8](#_Toc316977412)

[3.2.11 Resource utilization requirement 8](#_Toc316977413)

[4 GUI 8](#_Toc316977414)

[5 System Architecture 8](#_Toc316977415)

[6 System evolution 8](#_Toc316977416)

# Introduction

## Purpose

The purpose of this document is to specify the requirements for the development of an audio acoustic app that we are creating. The intended customers will be YouTubers, voice actors, musicians, music producers and event organisers (for concerts, conferences and presentations). There is currently no popular app that can help in this area of expertise.

## Project Scope

The scope of the project is for the likes of YouTubers and home producers and for novice people in media production, specifically dealing with sound, to speed up the process of performing sound acoustic treatments. It is not necessary for the user to have prior knowledge. Sound engineering itself is rather complicated and can require expensive equipment and knowledge to get the acoustics of a room just right.

We aim to simplify this process by providing an Android app. The database within the app will contain an extensive amount of default figures and calculations, allowing the user to simply clap their hands to see how their current acoustics of their room is. Then they will be able to input the size of the room, and get the required measurements, RT60, size of materials in their room, such as their carpet or walls. Steps for this will be included.

The user will then clap their hands again to see how their room measures up. If it is OK, the room is treated and ready for the user to produce their media. If not, they can restart the steps.

This app itself will be completed in Android Studio and will run on Android version 4.1 (Jelly Bean) to 6.0.1 (Marshmallow). 97.2% of Android users use these versions (Google, n.d.), hence why we will release the app in these versions.

Our idea stemmed from Cedric. Early 2016, Cedric saw there was a niche in the market for such an app. He tried to create something, but was lacking the knowledge, technology and funds of how to create such an app.

In late 2016, Cedric and Keith created a web page/application for an Audio Acoustic Assistant for their Web Application Development project. This project is the next step in this process.

### Acronyms and Abbreviations

AAA - Audio Acoustic Assistant

RT - Reverb Time

RT60 - Reverberation Time, 60 (millionth)

AF - Absorption factor

### Definitions

RT60 - An RT60 of a room or area is the time in seconds it takes a sound or noise to decrease its dB value to 60 millionth to become inaudible, for example, one clap. (Hagelskjaer, n.d.)

Room - For the purpose of this document, “room” is defined as the likes of a bedroom, area (such as a studio) or an arena.

Absorption Factor - The property of a body that determines the fraction of the incident radiation or **sound** flux **absorbed** or absorbable by a material.

# User Requirements Definition

The Customers are people who needs a tool that will help and teach them step by step on how to perform an acoustic treatment to a room for audio production.

Most YouTuber's will work in environments where they will need noise control, they will receive instructions in how to easily and cheaply improve the acoustics and sound quality and use our app to find absorbing materials E.g absorption for Chroma screen (green screen) as an example, which is important to CGI (Computer-Generated Imagery) environments, which a lot of up-to-date YouTuber's use for their videos.

Home Artists (producers, singers, musicians) will use our app to acquaint themselves with how to change any room in their house for a perfect acoustic quality when rehearsing, recording or performing their music.

DJ's and bands work in many locations, from clubs and pubs to house parties, they need a way to control the sound behavior in any place in, and our app can make the difference.

Event Organizers will have many things to do, one of the most important things when they are organizing a place that will have a lot of people in it (like a conference, concert, wedding reception), is to make sure people can be heard and clarity in the voice is optimal, our app will show them where they can put the speaker for best sound propagation and the right materials that should be used in this kind of situations.

The App has a multipurpose design, it easy to use, uses a minimum of space and will have a calculation and recording audio capabilities specifies it to the user's location of use.

# Requirements Specification

## Functional requirements

### Use Case Diagram

### Requirement 1 Database

#### Description & Priority

The database will hold all the preset values (absorption factor, frequency constants and actions to be performed under specific conditions, all data that will be placed into every calculation and as part of the result.

The database creation is essential to the application as the application will not function without a database.

#### Use Case

**Scope**

The scope of this use case is to build a database for the application to hold the values needed for the calculation.

**Description**

This use case describes the database that will be used to hold values for the calculations.

**Flow Description**

**Precondition**

The system is in initialisation mode

**Activation**

This use case starts when a user selects a value to be placed into the calculation.

**Main flow**

1. The user selects the material.
2. The system selects the value that corresponds with that material.
3. The system places that value into the calculation.
4. The user selects another material.
5. Steps 1 – 4 are repeated until all values are inserted into the calculation.

**Termination**

The system has entered all values into the calculation

**Post condition**

The system goes into a wait state

### Requirement 2 Navigation

#### Description & Priority

The application will be able to guide the user through the process as easy as possible.

The application needs to have a simple design to allow the user to navigate through the application easily, because there is a lot of information and inputs for the user to understand.

#### Use Case

**Scope**

The scope of this use case is to ensure that the user is not overwhelmed by the application.

**Description**

This use case describes layout and design of the application

**Flow Description**

**Precondition**

The system is waiting for the user to select a button.

**Activation**

This use case starts when the user presses a button.

**Main flow**

1. The system identifies the button chosen by the user.
2. The system selects the corresponding page/tab/selection for the chosen button.
3. The system continues as normal.

**Termination**

The system has selected the page/tab/selection for the chosen button.

**Post condition**

The system goes into a wait state

### Requirement 3 Select User Role (according to user role)

#### Description & Priority

The application will ask the user to select a variety of roles based on their specific needs.

These selections will change the values and actions taken throughout the application so choosing a user role is essential

#### Use Case

**Scope**

The scope of this use case is to change the values and room conditions in the calculations based on the role chosen.

**Description**

This use case describes the different scenarios for creating an acoustic treatment and changes the values in the application depending on the role.

**Flow Description**

**Precondition**

The system is waiting for the user to select a role.

**Activation**

This use case starts when the user chooses a role.

**Main flow**

1. The system identifies the user role chosen by the user. (e.g Youtuber)
2. The system selects the corresponding range of values and conditions for the chosen role.
3. The system continues to user input phase with a pre-set template from the role chosen.

**Termination**

The system has selected the values for the chosen role and the activity is finished.

**Post condition**

The system goes into a wait state

### Requirement 4 Sound input (Digital Sound Processing)

#### Description & Priority

The application will be able to record an input from the phone’s microphone and read the sound sample to measure the reverberation time in seconds inside the room and what needs to be done to archive the ideal RT.

The sound input is essential for the application as the application will generate a value from it which will be stored as an initial state for the processes and calculations. The sound input function will make the application easier and more user friendly to those who are not familiar with sound engineering, which make up a large percentage of our target audience.

#### Use Case

**Scope**

The scope of this use case is to perform a “digital sound processing” from a microphone input and turn that sample into a value in seconds that can be used by the app.

**Description**

This use case describes how the recorded sample will be measured by the system to generate initial value.

**Flow Description**

**Precondition**

User grants **permission** to use device microphone.

Microphone turned on.

App records sound until it disappears.

App takes the length of the sample and turn it into a value in seconds, this value represents the initial RT of the room.

**Activation**

This use case starts when the user presses the sound input button.

**Main flow**

1. The system identifies that the user has selected sound input.
2. The system begins recording and gives the user a set of instructions (Make the room quiet, Clap, stay silent until the recording is finished)
3. The system records the sound from the clap until it disappears
4. The system turns this record into a value in seconds for the system to use alongside values received from the database.

**Exceptional flow**

E1 : Failed recording.

1. The system identifies that the user has selected sound input.
2. The system begins recording and gives the user a set of instructions (Make the room quiet, Clap, stay silent until the recording is finished)
3. The system records the sound from the clap.
4. The system cannot turn this sound into a value due to the recording being invalid.
5. The system asks the user to retry the process.

**Termination**

The system has turned the sound input into a frequency value.

**Post condition**

The system goes into a wait state

### Requirement 5 Type of room (Ft2 or mt2)

#### Description & Priority

Select the type of room owned by the user

#### Use Case

**Scope**

The scope of this use case is to change the values in the calculations based on the scenario chosen.

**Description**

This use case describes the different scenarios for creating an acoustic treatment and changes the values in the application depending on the scenario.

**Flow Description**

**Precondition**

The system is waiting for the user to select a scenario.

**Activation**

This use case starts when the user chooses a scenario.

**Main flow**

1. The system identifies the scenario chosen by the user.
2. The system selects the corresponding range of values for the chosen scenario.
3. The system continues as normal.

**Termination**

The system has selected the values for the chosen scenario and the system continues.

**Post condition**

The system goes into a wait state

### Requirement 6 Input for calculations

#### Description & Priority

This is how the user will input their specific values that will enter into the calculation.

This is essential because without the users input the calculation cannot be accurate for the users specific use case.

#### Use Case

**Scope**

The scope of this use case is to allow the user to enter their data easily and for it to work correctly with the values from the database to allow the calculation to function.

**Description**

This use case describes the users input into the calculation

**Flow Description**

**Precondition**

The system is waiting for an input

**Activation**

This use case starts when a user inputs their values into the system.

**Main flow**

1. A user enters their values into the system. (length x width for each surface, floor, walls, ceiling)
2. The system identifies the values entered by the user
3. The system places these values into the calculator

**Termination**

The system has accepted all inputs from the user and they have been placed into the calculator.

**Post condition**

The system goes into a wait state

### Requirement 7 output from calculations.

#### Description & Priority

This is the output from the calculations, this is what the user will receive after the calculation has used all the values from their input and the database.

This is essential as this output is what the users are using our application for, This output will tell the user what they need to do to improve the sound quality of their room.

#### Use Case

**Scope**

The scope of this use case is to give the user a value which they need to reach to have the optimal sound quality of their room, and to tell the what they need to do to reach this value.

**Description**

This use case describes the output of the calculations that the user needs to fix the sound quality of their room.

**Flow Description**

**Precondition**

The system has received the input from the user and the values from the database.

**Activation**

This use case starts when a user presses calculate.

**Main flow**

1. The system identifies the values from the input and from the database.
2. The system inputs these values into the calculations and receives the output
3. The output is displayed for the user, alongside a set of instructions to reach the optimal value.

**Termination**

The system displays the value and the instructions.

**Post condition**

The system goes into a wait state

### Requirement 8 Comparing Results (req 3 and 4)

#### Description & Priority

#### Use Case

**Scope**

The scope of this use case is to change the values in the calculations based on the scenario chosen.

**Description**

This use case describes the different scenarios for creating an acoustic treatment and changes the values in the application depending on the scenario.

**Flow Description**

**Precondition**

The system is waiting for the user to select a scenario.

**Activation**

This use case starts when the user chooses a scenario.

**Main flow**

1. The system identifies the scenario chosen by the user.
2. The system selects the corresponding range of values for the chosen scenario.
3. The system continues as normal.

**Termination**

The system has selected the values for the chosen scenario and the system continues.

**Post condition**

The system goes into a wait state

## Non-Functional Requirements

### Performance/Response time requirement

The performance requirements of the application are that the user can navigate the application seamlessly through the different sections. The application shouldn't crash when navigating through the different calculations, And each calculation should operate quickly and produce a result with minimal loading time.

### Availability requirement

The user will need to have an android device and google play store installed.

The application will have a set of basic calculations that are available to everyone through a free app on the google play store.

A full version of the application will give the user access to a full range of advanced calculations however this full version will cost a one time payment through the google play store.

### Recover requirement

In the event of the application crashing, the user will be given a prompt to shut down the application and to send the details of the crash to us via email.

### Robustness requirement

The robustness of the application will be strong as it will run on multiple versions of android from 4.4 - 7.

### Maintainability requirement

The application shouldn't need to be maintained or updated as the calculations and functions of the application are quite simple.

### Portability requirement

The application shouldn't be difficult to port to ios/windows as it is fairly simple in its design.

### Extendibility requirement

The application should be easily extended, Once the Database of values has been created adding new calculations or values will be easy.

### Reusability requirement

The code of the application could easily be reused in other applications, as the database and gui code could be reused to create a similar app that uses calculations outside of audio acoustics.

# GUI

Include mock-ups of the key pages or stages of the system. Explain how they are linked. Explain how you addressed above requirements in the design. It is important that the mock-ups are in line with the functional requirements above, e.g., if one of your requirements is “user registration” then one of the screens listed in this section should show a registration page.

# System Architecture

Use a class diagram to outline the structure of the system. Explain briefly why you have chosen this architecture. You might want to use Visio or Rational Rose to create these.

# System Evolution

There are many options that the app could progress with. Such examples include:

* Virtual Reality / Use of camera
  + Instead of a user depending entirely on the phone’s microphone. With the help of a virtual reality headset or when the user pans the camera, the user could see sound waves and how they bounce off specific surfaces. The user could also see if a surface was changed, how that could impact the sound wave on that surface.
* Radio equipment
  + If a user has a radio (e.g. Bluetooth) microphone or camera, the above could be implemented, but using external devices instead of native ones.
* Professional equipment
  + If a user is a DJ, for example, they could link their Android device with their equipment. The equipment would automatically adjust, based on the results on the app, to produce the best sound quality.
* Statistics
  + Collection of information from users could see which is the most used room type and surfaces. Defaults could then be added, making the app faster, based on these statistics.
* Request a professional
  + If the user is completely perplexed and would simply just want someone who would do it for them, the user could request a call to a sound professional who could assist them.
* Instrument Tuner
  + By customer demographics and feedback, we’d be able to see if a tuner for musical instruments is what users would like in the app. There are many apps on the Play Store that are instrument tuners, but depending on the user’s experience, they may not like to go into another app. This is not required on the current version as the app itself is not a tuner.