National College of Ireland

BSHC2

2014

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colin**  **Allen**    **15540607**  Github username: **Colin303** | **Keith Feeney**    **15015556**  Github username:  **keithfeeneyNCI** | **Patrick Lawlor**    **15014606**  Github username:  **Paddyyyyyyyy** | **Fearghal McMorrow**    **15014584**  Github username:  **Fearg6** | **Cedric Vecchionacce**  **15011071**  Github username:  **CedricVz** |

Audio Acoustic Assistant

Technical Report



Table of Contents

[Executive Summary 4](#_Toc480196189)

[Definitions, Acronyms, and Abbreviations 4](#_Toc480196190)

[1 Introduction 5](#_Toc480196191)

[1.1 Background 5](#_Toc480196192)

[1.2 Motivation 5](#_Toc480196193)

[1.3 Project Overview 5](#_Toc480196194)

[1.4 Target group 6](#_Toc480196195)

[1.5 Technologies 6](#_Toc480196196)

[1.6 Distribution of tasks 6](#_Toc480196197)

[1.7 Structure 6](#_Toc480196198)

[2 System 7](#_Toc480196199)

[2.1 Requirements 7](#_Toc480196200)

[2.2 User Requirements Definition 7](#_Toc480196201)

[2.3 Requirements Specification 8](#_Toc480196202)

[2.4 Functional requirements 8](#_Toc480196203)

[2.5 Use Case Diagram 8](#_Toc480196204)

[2.5.2 Requirement Type of room (ft2 or mt2) 10](#_Toc480196205)

[2.5.3 Requirement Select User Role 11](#_Toc480196206)

[2.5.4 Requirement Input for calculations 13](#_Toc480196207)

[2.5.5 Requirement Output from calculations. 14](#_Toc480196208)

[2.5.6 Requirement Comparing Results (req 3 and 4) 15](#_Toc480196209)

[2.6 Non-Functional Requirements 16](#_Toc480196210)

[2.6.1 Database Requirement 16](#_Toc480196211)

[2.6.2 Navigation Requirements 16](#_Toc480196212)

[2.6.3 Performance/Response time requirement 16](#_Toc480196213)

[2.6.4 Availability requirement 17](#_Toc480196214)

[2.6.5 Recover requirement 17](#_Toc480196215)

[2.6.6 Robustness requirement 17](#_Toc480196216)

[2.6.7 Maintainability requirement 17](#_Toc480196217)

[2.6.8 Portability requirement 17](#_Toc480196218)

[2.6.9 Extendibility requirement 17](#_Toc480196219)

[2.6.10 Reusability requirement 17](#_Toc480196220)

[2.7 Design and Architecture 18](#_Toc480196221)

[2.8 Implementation 19](#_Toc480196222)

[2.9 Graphical User Interface (GUI) Layout 20](#_Toc480196223)

[3 Discussions and reflections 21](#_Toc480196224)

[4 Conclusions 22](#_Toc480196225)

[5 Further Development 23](#_Toc480196226)

[6 References 25](#_Toc480196227)

# Executive Summary

This Project is about creating an app that provides the user the amount of acoustic materials they’ll need to perform an acoustic treatment to any room. With many people wanting to become content creators on YouTube, produce their own media or organise small events, there is few app on the Google Play Store that can help with the acoustics of a room or area just right. There are apps to tune instruments but not any that address the above issue.

The solution to this is the creation of the AAA app. The app addresses this niche in the market. The app gets the user type; whether YouTuber, home artist, or event organiser. This Boolean value is transferred to the Advice page. It uses a Login System. The login values are sent and stored in a database. (Potential future use; Users can login to website and see all saved results, email marketing), a room area calculator, room selected; whether bedroom or empty room. In the Recording page, a value is taken from the phone’s microphone. Calculations are done to get the dB value from this input. The Advice page shows the results; the user type and how much acoustic foam is recommended (based on calculations from previous intents). An image link is also provided, so that users can tap it and purchase the acoustic foam. There’s also a Facebook link that opens the Facebook application if installed (The link opens in the default browser if Facebook is not installed)

## Definitions, Acronyms, and Abbreviations

AAA – Audio Acoustic Assistant

Intent – A page in an Android Application

dB – decibels

RT60 – “RT meaning reverberation time and 60 being decibels, the RT60 of a room is the time in seconds it takes for a sound to die down completely”. (Hzandbits Sound Effects, n.d.)

# Introduction

The project is for the likes of YouTubers, home-made recording studios and for novice people in presenting an event such as party in a small venue. 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.

The aim is 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.

## Background

The original idea came from Cedric who is a sound engineer. In early 2016, Cedric saw there was a niche in the market or such an app. He wanted to create something that would help simplify the process, but was lacking knowledge for technologies and was unsure about how he would go about it.

## Motivation

The motivation stemmed from when Cedric and Keith created a website for the Web Application Development module in our previous semester. It was decided by the team that we would continue this and develop this into an Android application.

## Project Overview

The Project itself is run over 13 weeks. The first few weeks, we were dealing with the Proposal and the Requirements Specification. From Week 4 – 10 we were learning how to use Android Studio as well as learning specific coding relevant to the Project as well as dealing with errors, etc. This was in preparation for writing coding for the app. On Week 9 & 10, a prototype was developed to show our app in the Mid-Point Presentation. After this, we were building on the prototype to develop the app further to the end result.

## Target group

The target group is both men and women, 18 – 40, who run Android versions 4.1 (Jelly Bean) to 7.1 (Nougat). 98% of Android users use these version, with just 2% of users using a previous version. 2.8% are currently using versions 7.0 and 7.1, but this has been included this version to allow for future proofing. (Google, n.d.)

## Technologies

Android Studio

Android Studio uses Java, XML and C++ embedded into the software. Android Studio is used to create apps specifically for Android devices. In this Project, Android Studio is used to create our application.

SqlLite - holds values of login system

MySql,

Java, - programming in Android Studio

XML, - values for layouts, design, manifest

C++, - C-Binding in Android Studio

Adobe Fireworks & GIMP – image editing

## Distribution of tasks

## Structure

Brief overview of each section

# System

## Requirements

## 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

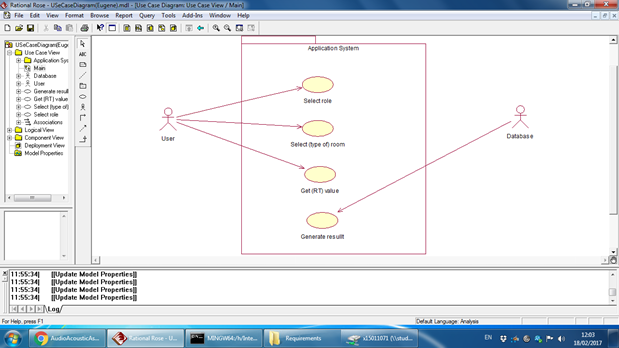
All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

## Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

## Use Case Diagram



* + 1. **Requirement Sound input (Digital Sound Processing)**

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.

* + - 1. **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 Type of room (ft2 or mt2)

* + - 1. **Description & Priority**

Select the type of room owned by the user

* + - 1. **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 Select User Role

* + - 1. **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

* + - 1. **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 Input for calculations

* + - 1. **Description & Priority**

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

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

* + - 1. **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 Output from calculations.

* + - 1. **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.

* + - 1. **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 them 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 Comparing Results (req 3 and 4)

* + - 1. **Description & Priority**
      2. **Use Case**

**Scope**

The scope of this use case is to compare values to check if results are correct.

**Description**

This use case describes how results are compared in order to check if RT has been reduced.

**Flow Description**

**Precondition**

The system holds the RT values of the room before and after calculations (with and without absorbing materials).

**Activation**

This use case starts when the user compares results.

**Main flow**

1. The system holds RT value of the room (microphone input).
2. The system holds RT value of the room with absorbing materials.
3. The system compares both values to check if they are the same.
4. If they are the same message displayed “Ideal Reverb Time Reached”
5. Else repeat process to add more materials.

**Termination**

The system displays congratulation message.

**Post condition**

The system goes into a wait state

List further functional requirements here, using the same structure as for Requirements 1 & 2. Most systems would have at least five main requirements.

## Non-Functional Requirements

### Database Requirement

All databases 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.

### Navigation Requirements

The navigation requirements include a mix of action bars and selection menus to improve compatibility. Screen relationship design for descendant and lateral navigation as well as ancestral and temporal navigation (Wireframe).

### 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 will be available on a variety of devices. This includes a variety of different smartphones and tablets all with different screen sizes

### 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.

## Design and Architecture

The focus on why we chose this architecture is because they are essential to our apps creation, development and success. Every successful application needs a simple and effective home screen with a logo that grabs the user’s attention.

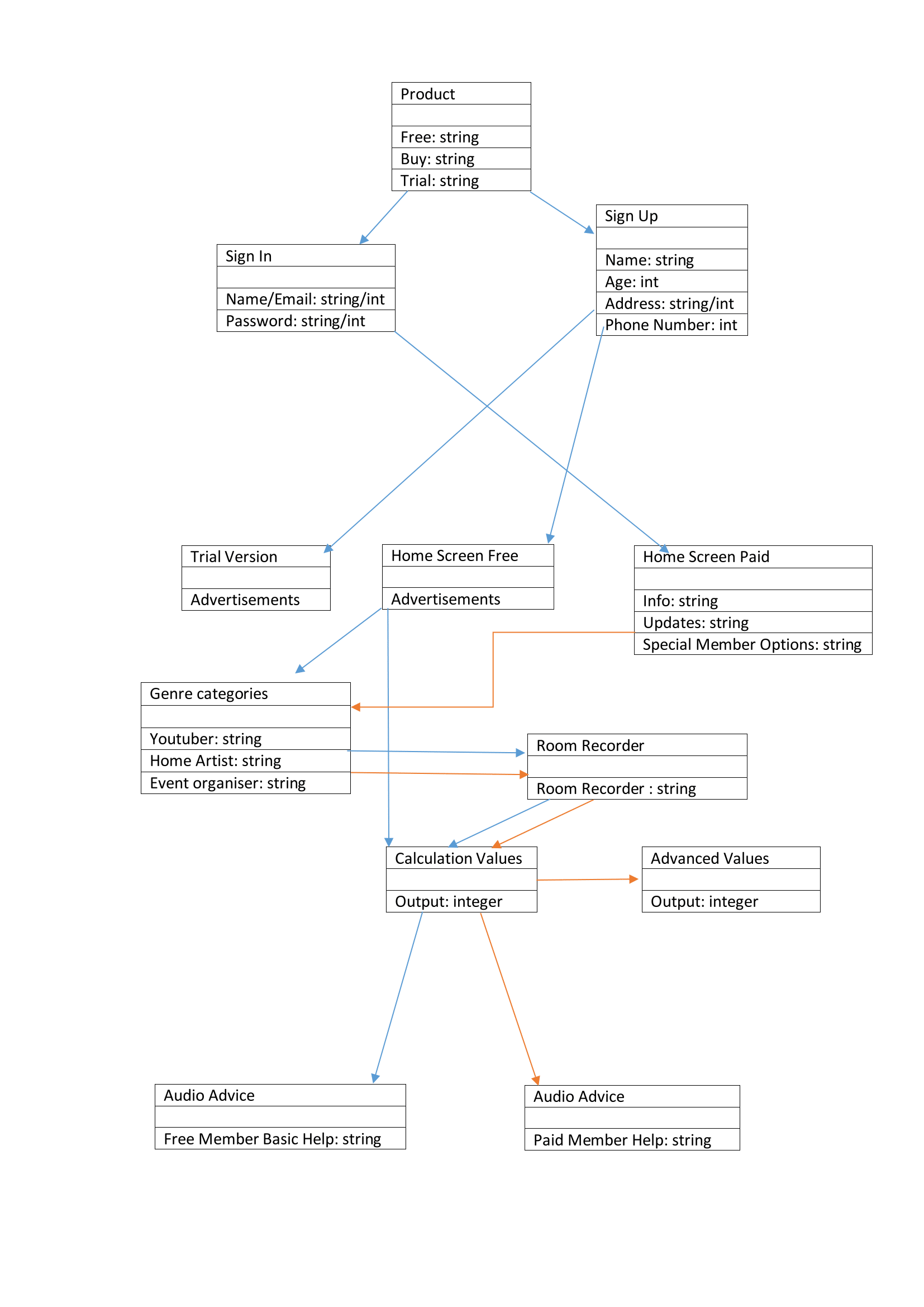
From there we need to create a user interface experience that satisfies the user and encourages them to learn how to use our application.

There will be two versions of the application, a free version with ads, and a paid version without ads, both will be separate uploads on the google play store.

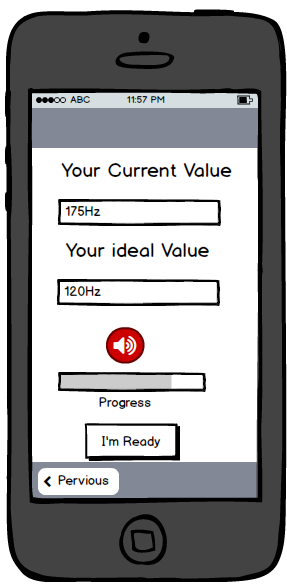
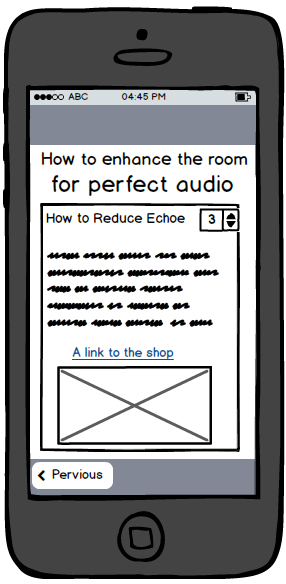
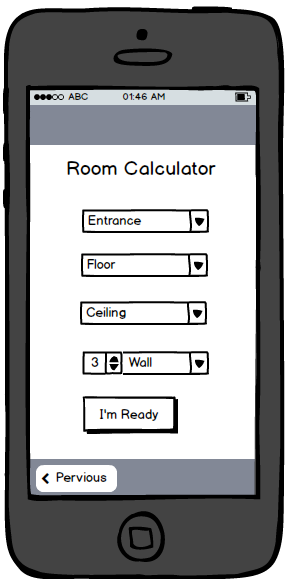
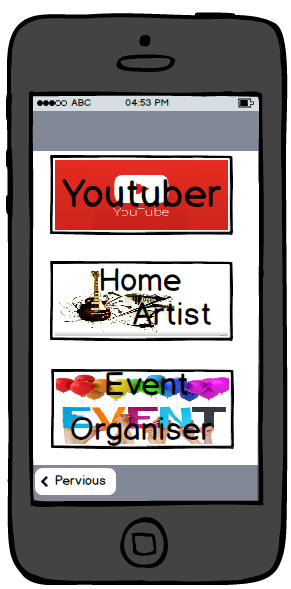
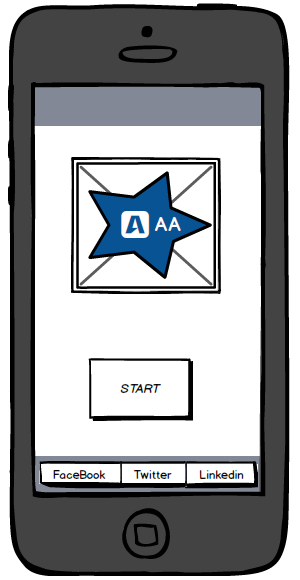
The premium version has all the content and is kept up to date with software improvements and changes as they happen. The free version will not have these options and is mainly for testing our product. The free version has mandatory advertisements that will help keep our application monetized. On the home page, you can choose your own specific user role, from YouTuber, amateur musician or any event regarding audio enhancements.

Moving forward will bring the user to the room recorder with all the calculators. The free version is limited, naturally, and only offers basic calculations and very few options. The premium version allows the user to access tips, tricks, help and a full range of calculators needed to pinpoint the perfect sound. When all the values have been added up the perfect formula is displayed to the user to help them improve their sound quality. More help and specifications are on offer with the premium version but the free version will offer hints and very little full-on support.

## Implementation



## Graphical User Interface (GUI) Layout



(Balsamiq Studios, LLC; McMorrow, F., 2017)

# Discussions and reflections

Discussions and reflections regarding:

Learning – what you learned

Skills – what skills you developed

Process – the Project development process

Client Feedback Session

Project Module

# Conclusions

Describe the advantages/disadvantages, opportunities and limits of the project.

# Further Development

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

* Room Measurement by using GPS or photograph
* Instead of using text field inputs to calculate room volume and Ideal RT, we’ll use GPS component on phone to calculate volume of the room.
* 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.

# References

Balsamiq Studios, LLC; McMorrow, F., 2017. *AAA - Grid.* [Online]   
Available at: https://epiccool.mybalsamiq.com/projects/aaa/grid  
[Accessed 17 February 2017].

Google, n.d.. *Dashboards.* [Online]   
Available at: https://developer.android.com/about/dashboards/index.html  
[Accessed 8 February 2017].

Hagelskjaer, C., n.d.. *Reverberation time (RT 60) – what is it and why is it important?.* [Online]   
Available at: http://www.hzandbits.com/articles/recording-studio-project-index/recording-studio-design-theory/reverberation-time-rt-60/#.WJnJ8TuLSUl  
[Accessed 7 February 2017].

Hzandbits Sound Effects, n.d. *Reverberation time (RT 60) – what is it and why is it important?.* [Online]   
Available at: http://www.hzandbits.com/articles/recording-studio-project-index/recording-studio-design-theory/reverberation-time-rt-60/  
[Accessed 17 April 2017].