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](#_Toc386657037)

[Definitions, Acronyms, and Abbreviations 4](#_Toc386657038)

[1 Introduction 5](#_Toc386657039)

[1.1 Background 5](#_Toc386657040)

[1.2 Motivation 5](#_Toc386657041)

[1.3 Project Overview 5](#_Toc386657042)

[1.4 Target group 5](#_Toc386657043)

[1.5 Technologies 5](#_Toc386657044)

[1.6 Distribution of tasks 5](#_Toc386657045)

[1.7 Structure 5](#_Toc386657046)

[2 System 6](#_Toc386657047)

[2.1 Requirements 6](#_Toc386657048)

[2.2 User Requirements Definition 6](#_Toc386657049)

[2.3 Requirements Specification 6](#_Toc386657050)

[2.4 Functional requirements 6](#_Toc386657051)

[2.5 Use Case Diagram 6](#_Toc386657052)

[2.5.1 Requirement 1 <name of requirement in a few words> 6](#_Toc386657053)

[2.5.2 Requirement 2 <name of requirement in a few words> 8](#_Toc386657054)

[2.6 Non-Functional Requirements 9](#_Toc386657055)

[2.6.1 Performance/Response time requirement 10](#_Toc386657056)

[2.6.2 Availability requirement 10](#_Toc386657057)

[2.6.3 Recover requirement 10](#_Toc386657058)

[2.6.4 Robustness requirement 10](#_Toc386657059)

[2.6.5 Security requirement 10](#_Toc386657060)

[2.6.6 Reliability requirement 10](#_Toc386657061)

[2.6.7 Maintainability requirement 10](#_Toc386657062)

[2.6.8 Portability requirement 10](#_Toc386657063)

[2.6.9 Extendibility requirement 10](#_Toc386657064)

[2.6.10 Reusability requirement 10](#_Toc386657065)

[2.6.11 Resource utilization requirement 10](#_Toc386657066)

[2.7 Design and Architecture 10](#_Toc386657067)

[2.8 Implementation 10](#_Toc386657068)

[2.9 Graphical User Interface (GUI) Layout 10](#_Toc386657069)

[3 Discussion and reflection 11](#_Toc386657070)

[4 Conclusions 12](#_Toc386657071)

[5 Further Development 13](#_Toc386657072)

[6 References 14](#_Toc386657073)

# Executive Summary

Maximum 300 words. The abstract should mention the problem being addressed, describe the technical solution.

## Definitions, Acronyms, and Abbreviations

AD Another Definition

# 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). Most Android users use these version, with a minute amount (2%) of users using a previous version. 2.8% are currently using 7.x, 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

This section describes the set of objectives and requirements for the system from the customer’s perspective. What are the clients saying they want?

## 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

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The Use Case Diagram provides an overview of all functional requirements.

### Requirement 1 <name of requirement in a few words>

The heading of this section should read, e.g., “Requirement 1: User registration” or “Requirements 1: Participant takes test”

#### Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

#### Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**Post condition**

The system goes into a wait state

### Requirement 2 <name of requirement in a few words>

#### Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

#### Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**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

Specifies any other particular non-functional attributes required by the system. Examples are provided below. **Remove the requirement headings that are not appropriate to your project.**

### Performance/Response time requirement

### Availability requirement

### Recover requirement

### Robustness requirement

### Security requirement

### Reliability requirement

### Maintainability requirement

### Portability requirement

### Extendibility requirement

### Reusability requirement

### Resource utilization requirement

## Design and Architecture

Describe the design, system architecture and components used. Describe the main algorithms used in the project. (Note use standard mathematical notations if applicable).

An architecture diagram may be useful. In case of a distributed system, it may be useful to describe functions and/or data structures in each component separately.

## Implementation

Describe the main classes/functions used in the code. Draw the Data flow diagram

Consider to show and explain interesting code snippets where appropriate.

## Graphical User Interface (GUI) Layout

Provide screenshots of key screens and explain.

# 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

With more resources, where could the results of this project lead to?

# References

It is recommended that students use the Harvard style:

References to web sites must include the access dates.

The library provides a study guide on Harvard style referencing.