|  |
| --- |
| HCC2, HCCE2, BSHC2, BSHCE2, BSHBIS2, BSHBISE2 |
| Requirements Specification (RS) |
| [Type the document subtitle] |

|  |
| --- |
| insert all student names  [Pick the date] |

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 set out the requirements for the development of ……………

The intended customers are ……….

## Project Scope

The scope of the project is to develop a …………….The system shall have a ……………

John Smyth was involved in discussions with John Ryan from AN Company Ltd. To elicit the following requirements

This section also details any constraints that were placed upon the requirements elicitation process, such as schedules, costs, or the software engineering environment used to develop requirements.

## Definitions, Acronyms, and Abbreviations

AD Another Definition

……..

# 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 Database creation

#### Description & Priority

The database will hold all the values that will be placed into every calculation.

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 Sound input

#### Description & Priority

The application will be able to record an input from the phone’s microphone and read the frequency to measure the reverberation time in the room and what needs to be done to correct the acoustic quality.

The sound input is essential for the application as the application will still be functional by allowing the user to access the calculations. However, 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 record the reverberation time of a room and turn that recording into a value that can be measured by the system. Known as RT60.

**Description**

This use case describes the recording that will be measured by the system.

**Flow Description**

**Precondition**

The system is in recording sound through the phones microphone to turn the sound into a value using the sounds frequency.

**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
4. The system turns this sound into a frequency value 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 frequency 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 3 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 calculation

**Termination**

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

**Post condition**

The system goes into a wait state

### Requirement 4 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 5 Scenario

#### Description & Priority

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

These selections will change the values throughout the application so choosing a scenario is essential

#### 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

**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

# 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

This section describes how the system could evolve over time.