# Project Acceptance

Team: 3200\_35, Sound Meter Data Visualisation.

# Objectives

The purpose of this document is to outline the tests required to pass to determine whether the project is of an acceptable standard to deliver to the clients. The testing strategy is a combination of Unit tests - which will test individual code snippets, integration tests - which test that the systems components (backend, frontend) work correctly together, system tests – testing the system as whole, and acceptance tests – testing the end-user experience. The tests will be performed using a variety of technologies, such as code testing modules, and the Selenium testing framework.

<<In this section, describe the purpose of this document. Forecast the testing strategy, the unit tests, integation test and system tests listed in this document.>>

# Document References

Requirement documents, which will be the product of the Sprint 1, are used to communicate the aims of the project in a clear, concise way that should reflect the needs of the client and stakeholders. The two main documents that will be focused on will be a Business Requirements Document (BRD), which details the high-level problems that our system should solve in relation to the customers needs. It will be concise and direct – in dot point form. We will also deliver a draft Functional Requirements Document (FRD), which defines how our system will behave to achieve the business requirements defined in the BRD.

Design documents will be delivered at the beginning of Sprint 2. Once the high-level business requirements have been finalised, we will detail how the system will be technically implemented. This will include a system architecture, as well as pseudocode examples of any algorithms and data structures.

There will be multiple stages of functional and non-functional testing before the final delivery of the project. Firstly, there will be code unit tests run on individual snippets. Secondly, there will be system tests, which test how components will work together. There will also be user acceptance testing, which tests the end-user experience compared to the requirements.

<<In this section, reference the major documents produced during project development. Explains the relationships among the requirements documents, design documents, implementation documents and the test procedures.>>

**Test Summary**

The scope of work is broken down into four epics.

1. Gathering sound data that will provide insights into how sound will affect fatigue.
2. Formatting data into a backend database.
3. Creating a web application that will be reusable by Thales
4. Displaying the data on a web application with effective and clear visualisations.

We will describe some high-level tests that will be applied to these systems.

![Diagram

Description automatically generated]()

<<In this section, describe the functions of the system tested in this document (Refer to Requirements Analysis Document and Problem Statement)>>

**Testing Strategy**

**Test A**

|  |  |
| --- | --- |
| **System to be tested** | Backend database |
| **Testers** | Kese, Caleb |
| **Timeframe** | During development and upon integration |
| **Test location** | In development and production environments |
| **Test method** | Unit tests, schema tests |

Test A is a test of the backend data base system, to ensure that it obeys ACID properties, the schema is correct, and it is properly normalised. These tests will primarily be performed in a scripting language such as Python, where data can be inserted and retrieved to verify its integrity.

**Test Specification**

The database must:

1. Hold data regarding sound-level measurements across different locations
2. Be ACID compliant
3. Resilient to updates and deletions.

**Test Description**

|  |  |
| --- | --- |
| **Location of test** | Development server |
| **Means of control** | Automatically using a test driver |
| **Data** | Input: SQL Select, Insert, Update and Delete queries  Output: Returned data/logs |
| **Procedure** | Python script |

**Test Analysis Report**

The performance of the test will determined by:

1. Whether all the SQL tests pass successfully
2. Whether the data returned is correct
3. Whether the database is ACID compliant.

**Test B**

|  |  |
| --- | --- |
| **System to be tested** | Frontend webserver |
| **Testers** | Shane, Darby |
| **Timeframe** | During development and upon integration |
| **Test location** | In development and production environments |
| **Test method** | Unit tests, integration tests |

Test B is a test of the front-end web server system, to ensure that it is responsive, and all features are working as required. It will be performed primarily using Selenium browser-emulation tests.

**Test Specification**

The webserver must:

1. Be able to visualise sound data in a meaningful way.
2. Be able to slice the data on meaningful axis

**Test Description**

|  |  |
| --- | --- |
| **Location of test** | Development server |
| **Means of control** | Manual input  Automatic selenium tests |
| **Data** | Input: End-user behaviour, automated tests  Output: Visualisations |
| **Procedure** | Selenium test |

**Test Analysis Report**

The performance of the test will determined by:

1. How interpretable the client believes the visualisations are.
2. Whether the data returned is correct
3. Whether the automated tests pass

**Test C**

|  |  |
| --- | --- |
| **System to be tested** | Sound collection |
| **Testers** | Aditi |
| **Timeframe** | During development and upon integration |
| **Test location** | In development and production environments |
| **Test method** | Unit tests, integration tests |

Test C is a test of the sound-collection system, to ensure that the devices we are using to measure the sound levels report timely and accurate data. This will be primarily conducted using a set of scripts that connect to the devices and compare them to other known sound levels. s

**Test Specification**

The sound collection system must be able to:

1. Accurate record sound levels over time

**Test Description**

|  |  |
| --- | --- |
| **Location of test** | Physical room, software emulation |
| **Means of control** | Manual sound level input |
| **Data** | Input: Varying sound levels in different rooms  Output: Accurate sound level readings |
| **Procedure** | Python scripts |

**Test Analysis Report**

The performance of the test will determined by:

1. How consistent the sound level measurements are
2. How accurate the measurements are