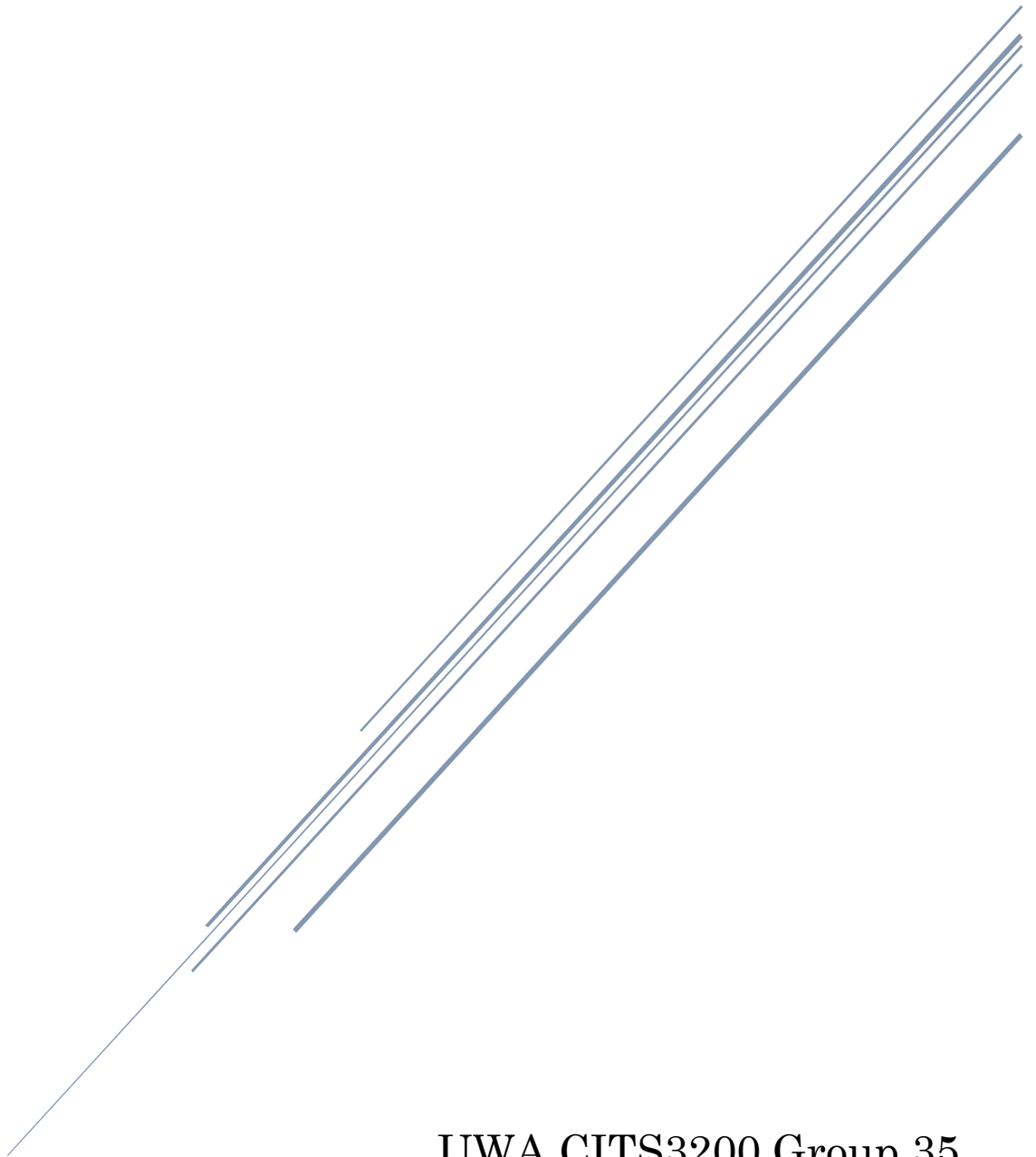


SOUND METER DATA VISUALISATION

Sprint 1



UWA CITS3200 Group 35

Caleb Cheng, Kese Gasbarri, Shane Monck, Aditi Malu & Darby Edwards

Table of Contents

1. Scope of Work.....	2
1.1 Project Background.....	2
1.2 Goals of Agreement.....	2
1.3 Objectives	2
1.4 Administration	3
1.5 Timeline.....	3
2. Skills and Resources.....	4
2.1 – Task.....	4
2.2 - Skills	4
2.2.1 - Front-end skills	4
2.2.2 - Back-end skills include	5
2.2.3 – Testing skills.....	5
2.2.4 - General skills include	5
2.3 - Resources.....	6
2.3.1 - Front end resources	6
2.3.2 - Back-end resources	6
2.3.3 - General resources.....	6
3 – Risk Register	7
4. Project Acceptance.....	9
4.1 - Objectives	9
4.2 – Document References	9
4.3 – Test Summary.....	9
4.4 – Testing Strategy.....	10
4.4.1 - Test A.....	10
4.4.2 - Test B.....	11
4.4.3 - Test C	12
5. User Stories.....	13
5.1 - Sprint 2 Goals	13
5.2 - User Stories.....	13
5.3 - Sprint 2 Tasks.....	13

1. Scope of Work

1.1 Project Background

Thales in Australia supplies and sustains a range of products for the Australian Defence Force (ADF). Thales works in partnership with the Royal Australian Navy; having designed and delivered a variety of solutions in both the underwater and above water domains.

The Clients Ben Hicks and Rebecca Waters who work for Thales, have come forward with a project that will enhance their current system of fatigue measurement for those serving at sea as to help with the health and wellbeing of the sailors and to improve productivity on board the ship.

The goal of the project is to display meaningful sound data using visualisation on a web application that will assist the Australian Navy Medical Officer in identifying sailors that could be experiencing fatigue and other health issues due to extended exposure to large sounds while working aboard the vessel.

1.2 Goals of Agreement

At the end of the project (Monday 18th of October) Thales will receive a web application that displays and conveys the information and data that will help users determine those at risks to fatigue due to sound.

1.3 Objectives

1.3.1

Task: Gather sound data and research that will outline the potential affects sound will have on sailors over a period.

Deliverable: Collected and stored multiple sample sound level recordings, as well as have research completed on what levels of sound and time exposure has on people.

1.3.2

Task: Format data into a database to be displayed on web application.

Deliverable: Built a sustainable back end for the application that has an appropriate schema for the database and runs on a local server.

1.3.3

Task: Create a web application that once finished can be easily migratable into the pre-existing web app used by the Australian Navy, it will Display the data, while outlining sailors that could been experiencing fatigue due to large exposure to sizeable amounts of decibels.

Deliverable: Display the data and information on the website, visualisation will be approved by the Client and to their liking, it will be similar to what their system already looks like. Written in the language that their pre-existing system is written in so that it is easily migratable.

1.4 Administration

Those working on the project are:

Full Name	Email
Caleb Cheng	22716794@student.uwa.edu.au
Darby Edwards	22713383@student.uwa.edu.au
Kese Gasbarri	22628168@student.uwa.edu.au
Aditi Malu	22526301@student.uwa.edu.au
Shane Monck	22501734@student.uwa.edu.au

Clients are:

Full Name	Email
Ben Hicks	Ben.Hicks@thalesgroup.com.au
Rebecca Waters	Rebecca.WATERS@thalesgroup.com.au

1.5 Timeline

The project is governed by three sprints following the Agile methodology.

Sprint 1 (2/8/2021 to 18/8/2021):

At the end of this sprint five deliverables will be delivered to the Client that will help set up the overall direction for the project.

These are:

- A Scope of Work
- A Skills and Resources Audit
- A Risk Register
- Project Acceptance Tests
- A set of stores to be completed in Sprint 2

Sprint 2 (23/8/2021 to 22/9/2021):

At the end of this sprint the team will demonstrate the set of interim goals proposed at the end of ***Sprint 1*** to the Client, where minutes will be recorded of the Client's thoughts about the system. As well as a final set of stories to see the project to completion, if the team is tracking ahead of where it expected to be at this stage, the new stories will include extensions to the project as originally envisaged. On the other hand, if difficulties have emerged, the new stories may involve reduction in project scope. Both situations will be negotiated with the Client.

Sprint 3 (27/8/2021 to 18/10/2021):

At the end of this sprint the final system is to be delivered to the Client, both as source code and as an installed system on the Client's platform of choice. A demonstration of the project will be given to the Client and feedback will be obtained.

Display meaningful sound data using visualisation on a web application that will assist the Australian Navy Medical Officer in identifying sailors that could be experiencing fatigue and other health issues due to extended exposure to large sounds while working aboard the vessel.

2. Skills and Resources

2.1 – Task

The project will require our group to draw upon prior skills we have acquired in web development and data analysis. We are also required to expand our knowledge and skill base to meet and fulfil our client's needs and requirements. We have constructed a skills audit to further understand our team's competencies and what areas will require attention and developing proficiency. The skills are broken into four distinct categories: frontend, backend, testing, and general skills.

2.2 - Skills

2.2.1 - Front-end skills

- HTML and CSS are required for structuring and styling the web page(s), which we will use to display our visualisations and analytics.
 - All group members have an adequate level of proficiency, having completed Agile Web Development last semester.
- JavaScript and React will display an array of visualisations in an interactive dashboard for user decision making and analysis. For example, D3 – Data-Driven Documents library has been identified as a potential library of use.
 - Collectively we have sufficient experience in JavaScript; however, we have minimal experience using React. We will undertake additional learning and online tutorials to acquire the necessary competency to complete the project.
- NodeJS is used to communicate with the API to collect the required data and perform pre-computation before passing it to the front end.

- No group members have had prior experience using NodeJS, and this is a skill that requires developing across the board. We will undertake additional learning and online tutorials to acquire the necessary competency to complete the project.

2.2.2 - Backend skills include

- SQLite will be used as the database engine to store sound levels, 'room' profiles along with the officer's location and exposures to dangerous decibel levels
 - We have the necessary skills to successfully implement a database that can adequately store and serve the data.
- Flask will be utilised to communicate with the database and pass the requested information to the API.
 - Our group has the adequate experience and skills required to implement a system capable of achieving the desired outcome.
- An API or WebSocket will be used for communication between the front end and the database
 - Our team has had limited experience deploying an API and will need to undertake additional learning via online sources.

2.2.3 – Testing skills

Testing will be required to ensure that our application performs as desired, which is vital as we will have five team members working on the project simultaneously. The information our application provides will aid in monitoring an officer's wellbeing whilst onboard a navy ship. Testing will take form in various ways, such as testing the integrity of the database, testing the staged data before displaying it to the end-user, and testing the different features we incorporate.

- Our team has had experience writing test cases from prior units; however, we require minimal additional resources to write test cases of this scale.

2.2.4 - General skills include

- Collecting sound readings from various environments, including operating a sound monitoring device and storing the data collected.
 - Although a relatively straightforward task, some thought will have to be given to how we sample the data from chosen locations to simulate the environment onboard a navy ship. Research is required to understand the different sound profiles of the various rooms on board a vessel and finding an accessible location to simulate that environment.
- Interpret, comprehend, and understand the data collected to draw meaningful conclusions and display the information in a way health officers can make informed decisions. We will require an understanding of the relationship between sound levels and fatigue.
 - Since this is an area that all team members had no prior understanding of before our first meeting. We have researched this field and will continue to do so as required to ensure our methodology is effective.
- Project management skills, including utilising the Agile method keeping our project on track.
 - We have all had some experience managing a project; however, this is their first time driving a project of scale for some.

- Communication skills are essential across all aspects of the project, from organising intergroup tasks and meetings to communicating our problems and findings to our client.
 - Communication is a skill that requires time and experience to master and develop. The best way to improve is by simply doing it and asking for feedback, learning from experience, and asking yourself what was effective and what wasn't.
- Communication skills are essential across all aspects of the project, from organising intergroup tasks and meetings to communicating our problems and findings to our client.
 - Communication is skill that requires time and experience to master and develop. The best way to improve is by simply doing it and asking for feedback and learning from experience what worked and what wasn't effective.

2.3 - Resources

Collectively, we have a strong base of knowledge in certain areas, allowing the group to build additional skills where required. We have identified a number of resources that will enable the group to develop the necessary skills, we have broken the variety of resources into sub-categories.

2.3.1 - Front end resources

- React
 - Academind – online video tutorials
 - freeCodeCamp.org – online videos and articles
 - w3schools – online modules and tutorials
- NodeJS
 - Programming with Mosh – online video tutorials
 - freeCodeCamp.org – online videos and articles
 - Tutorialspoint – online modules and tutorials

2.3.2 - Backend resources

- API
 - freeCodeCamp.org – online videos and articles
 - REST API Tutorial – online modules and tutorials
 - API Metrics – online modules and tutorials
- Websockets
 - freeCodeCamp.org – online videos and articles
 - Tutorialspoint – online modules and tutorials

2.3.3 - General resources

- Scholar articles and studies relating to the impact of sound on fatigue; from factors such as sound level, frequency and consistency.
 - For example, “Noise and Fatigue in Working Environments” – which conducted studies examining how noise can affect alert fullness and fatigue in the work place
- Mentorship from Matthew James, CTO of VGW.
- UWA and online material regarding agile methods, precisely the scrum method.

- Wrike – steps and strategies for running an effective project with the Agile methodology

The task also requires access to both hardware and software tools;

- Software tools include:
 - JavaScript D3 library
 - JavaScript React library
 - Flask web app framework for Python
- Hardware includes:
 - Access to sufficient personal computing power to construct the system.
 - Access to a sound recording device.

3 – Risk Register

The team has put together a risk register document to record all of the projects identified risks as well as ways to mitigate those risks. The risk register is vital as it minimises all unforeseen circumstances that will negatively or positively influence the project.

We have also added contingent actions, in case the risk does occur. The table also lists the likelihood, impact and severity of the risks to give a visual representation on how the risks might impact the project.

Risk description	Likelihood of the risk occurring	Impact if the risk occurs	Severity <i>Rating based on impact & likelihood.</i>	Owner <i>Person who will manage the risk.</i>	Mitigating action <i>Actions to mitigate the risk e.g. reduce the likelihood.</i>	Contingent action <i>Action to be taken if the risk happens.</i>
Project proposal and deliverable definition is incomplete/not well defined.	Low	High	High	Project Sponsor	Project purpose is re-analysed and discussed with the team with inputs from all sponsor members	Discuss with team about important aspects of projects to be fulfilled. Request high risk requirements to be cut down and removed.
Scope not well defined	Low	High	High	Team	Effective communication with client to make sure scope isn't missing details, or unnecessary features are not added which would lengthen the project lead time.	Share the plan and go through upcoming tasks with client as soon as possible.
Lack of communication with client causing lack of clarity and confusion	Medium	Medium	Medium	Project Manager	Effective and regular communication with client. Updating client regularly about the project. Ensure that we have detailed and up-to-date documentation on all aspects of project.	Correct misunderstandings immediately in project with client. Clarify areas that are not clear swiftly using assistance from Client
Scope creep/Risk in Scope Variation	Medium	High	High	Team	Keep documentation as up to date as possible to adapt to new changes if needed. Additional feature requests must be evaluated in terms of likely project impact, and decisions then need to be made in terms of whether to include or not include the new feature requests in the current phase.	Documentation of every new feature is necessary no matter how many changes. Authorisation from Client before any new changes are implemented.
Unresolved project detail conflicts	Low	Medium	Medium	Team	Hold regular meetings apart from formal meetings to ensure project details are discussed with every member. Give reviews of project plan to client in case of conflicts.	To resolve issues, discuss with team and client immediately as soon as conflict is identified.
Delay on early stage deliverables resulting in late submission.	Medium	High	High	Team	Ensure following the timetable as listed on CITS3200 page. Schedule other important requirements to ensure everyone is following the schedule.	Collaborate with team members to help recover from slow progress. Ensure that everyone is aware about immediate deadlines.
User experience	Low	Medium	Medium	Team	Make sure that website is user friendly and easy to use. Simple features will create the perfect user experience	Simplify features and discuss with client what parts are difficult to use to immediately change features
Lack of Scalability and Integration to Clients Software	Low	Low	Low	Team	Since team isn't aware of exact system used by client, it is difficult to know if integration will be completely perfect. Make sure client understands this.	Help client in changing features to ensure seamless integration and scalability to their systems.
Poor software quality	Medium	High	Medium	Team	Initiate review and testing early on, retest after fixes or changes. Plan for adequate time for testing and bug fixing. Make sure code is secure and ensures good speed performance.	Analyze the causes of errors with the objective of implementing process improvements. Test and change parts as required.
Poor team dynamics	Medium	Medium	Medium	Project Manager	Ensure team cohesion through proactive communication. Implement processes to escalate conflict resolution to auditor or unit coordinator if needed.	Correct misunderstandings and investigate reasons for conflict then provide feedback to each other on possible corrective measures.
In ability to obtain the required hardware	Medium	Medium	Medium	Team	Look at multiple source for obtaining a sound recording device that has the capability of providing the data required	Set up a back-up solution where we can record the data from our mobile phones and then simulate the collection the data we have collected

4. Project Acceptance

4.1 - 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.

4.2 – 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.

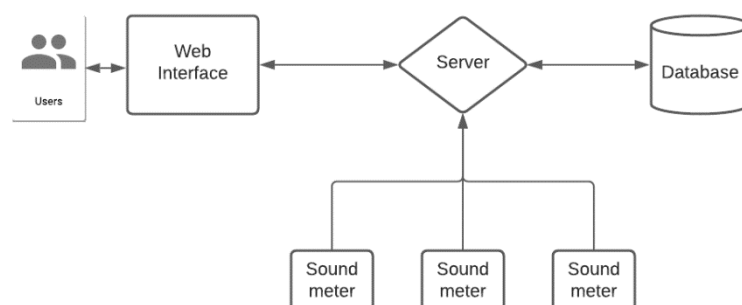
There will be multiple stages of functional and non-functional testing before the final delivery of the project. Firstly, there will be code unit tests run on individual snippets. Secondly, there will be system tests, which test how components will work together. There will also be user acceptance testing,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

4.3 – 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.



4.4 – Testing Strategy

4.4.1 - 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 be determined by:

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

4.4.2 - 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 frontend 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.
4. Whether the data returned is correct
5. Whether the automated tests pass

4.4.3 - 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

5. User Stories

5.1 - Sprint 2 Goals

For sprint 2, our goal is to have a skeleton of the final application up and running, with all the essential features working. Allowing us to present a working application to Thales giving us time to incorporate their feedback before commencing sprint 3 and finalising the application. As a team, we have identified three overarching goals which will be complete during sprint 2, being:

1. Collect and store sample sound level recordings
2. Build the back end of the application
3. Develop a dashboard displaying the sound samples in a meaningful way

5.2 - User Stories

To help understand what needs to be completed for each of the goals identified during sprint 1 for our second sprint, we have developed a number of user stories.

- As the Health Officer, I want to visualise the sound levels Officer A has been exposed to in order to monitor their fatigue levels.
- As the Health Officer, I want to visualise the sound levels Group A has been exposed to in order to see if there is any correlation across the Officers fatigue levels and noise exposure.
- As the Navy/Thales, I want an application that can collect noise levels in different rooms onboard a ship, so we have a database for conducting further research
- As the Health Officer, I want to visualise the noise levels in Room A to plan the Officers schedule, i.e., what environments will an officer fatigue quicker and how long can an Officer optimally function in that environment.
- As Thales, I want to have the capability to have multiple sensors within one room monitor sound to provide a better sample in larger rooms
- As the Health Officer, I want the data presented to me in an easily interpretable interface so I can draw conclusions for the information quickly and effectively

These User Stories outline the basic functionality we are planning to implement in sprint 2; more user stories will be identified for sprint 3 as we flesh out functionality. We are focused on developing a frame in which we can add user features further down the track and have the core functionality specified by the client completed to start with.

5.3 - Sprint 2 Tasks

We can further break down the three goals for sprint 2 into smaller actionable sub-tasks.

1. Collect and store sample sound level recordings
 - a. Use a phone to collecting sample sound readings giving the frontend and the backend some data to work with
 - b. Collect samples in various environments (quiet, normal, loud, very loud), expanding the variety of samples helping develop the functionality of only displaying people exposed to dB over x .
 - c. Collect samples from an environment where the noise levels spike. Helpful when testing whether we can pick up abnormal noise increases in the rooms we will be monitoring.

- d. Convert sound recordings into a format we can store and conduct our analysis from.
- 2. Build the back end of the application
 - a. Carefully plan out the schema for the database, so we can begin populating the database with sample data.
 - b. Implement the schema in SQLite using Flask as the database abstraction layer and object-relational mapper
 - c. Populate the database with sample data
 - d. Write functions that will be required to access and store data in the database
 - e. The database will be running on a local server for sprint 2
- 3. Develop a dashboard displaying the sound samples in a meaningful way
 - a. The dashboard will be built using Typescript and React for easy integration with Thales current software
 - b. Come up with a variety of visualisation in PowerPoint to give the client an indication of what they will be looking at.
 - c. Check to ensure the visualisations satisfy the client's requirements and begin to implement the visualisations in Typescript
 - d. Visualise the sample sound recordings in the dashboard

We have chosen to work on all major components of the application simultaneously, so as we develop the application and iteratively tweak the functionality and performance, all parts will work harmoniously together. For example, suppose the sound collection methodology must change for an unforeseen reason. In that case, we can quickly adapt the backend to handle the data, and the dashboard can be adjusted to use the data.