# 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
   1. Use a phone to collecting sample sound readings giving the frontend and the backend some data to work with
   2. 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*.
   3. 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.
   4. Convert sound recordings into a format we can store and conduct our analysis from.
2. Build the back end of the application
   1. Carefully plan out the schema for the database, so we can begin populating the database with sample data.
   2. Implement the schema in SQLite using Flask as the database abstraction layer and object-relational mapper
   3. Populate the database with sample data
   4. Write functions that will be required to access and store data in the database
   5. The database will be running on a local server for sprint 2
3. Develop a dashboard displaying the sound samples in a meaningful way
   1. The dashboard will be built using Typescript and React for easy integration with Thales current software
   2. Come up with a variety of visualisation in PowerPoint to give the client an indication of what they will be looking at.
   3. Check to ensure the visualisations satisfy the client's requirements and begin to implement the visualisations in Typescript
   4. 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.