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 the client gives 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 out the back end of the application
3. Develop a dashboard that displays the sound levels in a meaningful way

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

|  |  |  |  |
| --- | --- | --- | --- |
| Theme | As a/an | I want to… | So that… |
| Data Collection | User | Collect sound recordings | I can monitor the dBs people are exposed in one location |
| Data Collection | User | Collect sound recordings from multiple locations | I can monitor the dBs people are exposed to over a variety of locations |
| Dashboard | User | View the dBs person A has been exposed to | I can analysis how that may affect person A’s fatigue |
| Dashboard | User | View the dBs group A has been exposed to | I can analysis how that may affect the groups fatigue |
| Dashboard | User | View the dBs of room A | I can understand the environments where sound induced fatigue is likely to occur |
| Database | User | Look at the historic noise levels for person A, group A | I can analysis historic trends |
| Database | User | Look at the historic noise levels for room A | I can see how the sound environment has changed over time |

The majority of the user stories will be identified for sprint 3 as we flesh out functionality, for the time being we are focused on developing a frame in which we can add user features further down the track.

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

1. Collect and store sample sound level recordings
   1. Use a phone to collecting sample sound readings
   2. Collect samples in a variety of different environments (quiet, normal, loud, very loud)
   3. Collect samples from an environment where the noise levels spike. Helpful when testing whether we can pick up abnormal noise increases.
   4. Convert sound recordings into a format we can store and conduct our analysis from
2. Build out the back end of the application
   1. Carefully plan out the schema for the database
   2. We will be using flask and SQL lite for the backend
   3. Build the database with sample data collected from the above point
   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 that displays the sound levels in a meaningful way
   1. The dashboard will be built using Typescript and React
   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
   4. Transform the visualisations into a dashboard
   5. 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 still work 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.