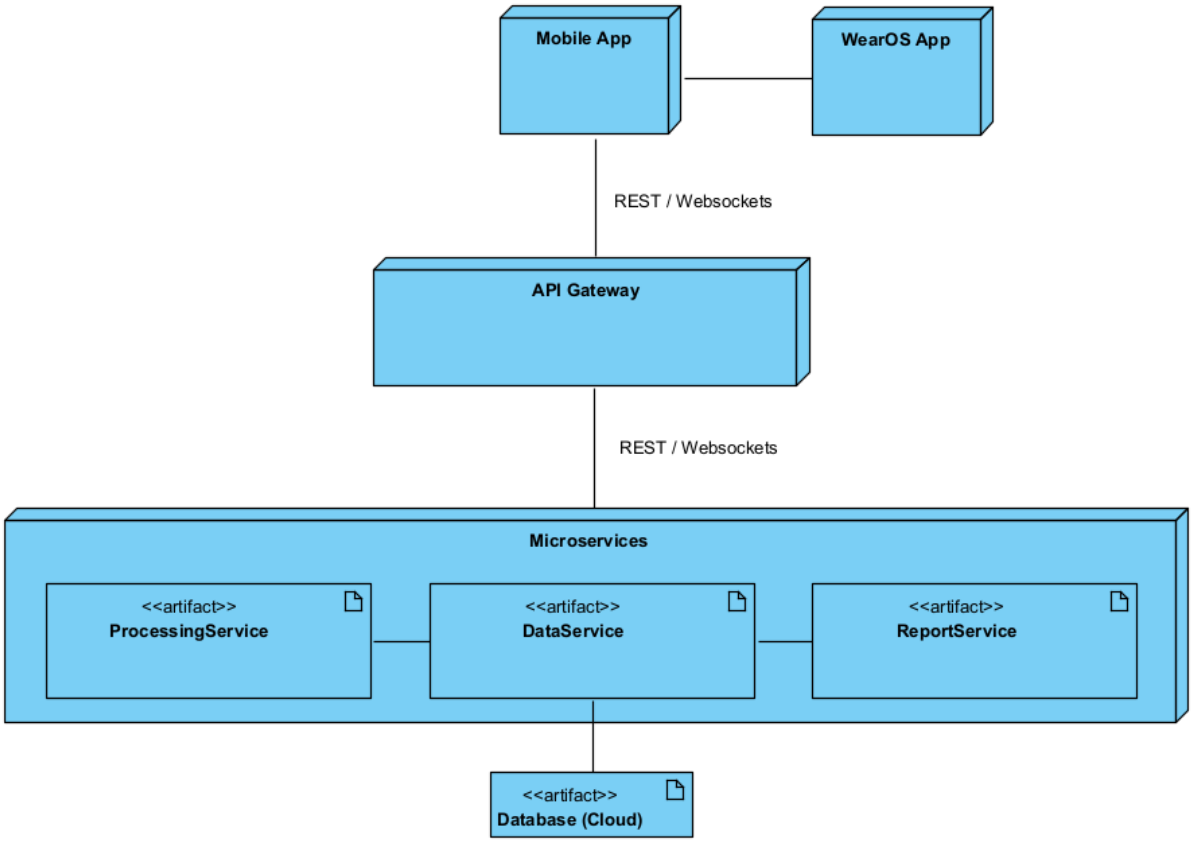
C4

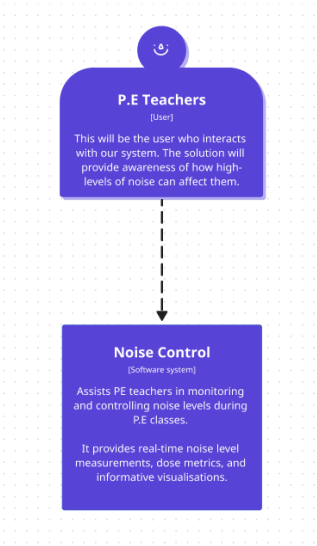
The diagram below was created by software students and it represents the backend system. I used it as inspiration and developed my own C4 model up to the C3 stage (third stage) with the assistance of the C4 model website as well. we received advice from the software group that we can design up to C2 and if we estimate that C3 will bring more value we could do that by making assumptions, the reason for this was because the structure of the micro services architecture was not defined further.

https://c4model.com



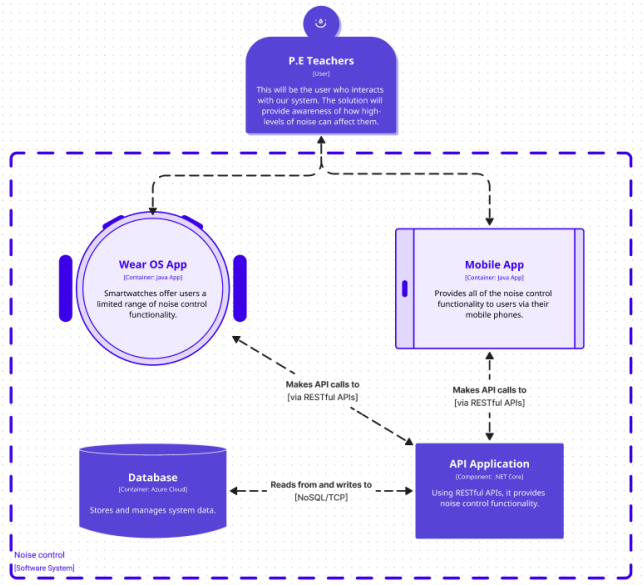
C1 System Context diagram:

The C1 system context diagram below shows the Users who are directly connected to the software system.



C2 Container diagram:

The Container diagram below shows the high-level shape of the software architecture and how different parts of it work together. It shows which responsibilities are assigned to each container and how they communicate with each other. In this diagram, the Users and the software system are directly connected to the containers. The software system is linked to the API application component, and the API application component is connected to the database.



C3 Report controller:

I was responsible for coming up with the components for the report controller, and I decided to come up with the following components which are Data aggregator, chart generator, and report exporter. I have a simple explanation of what each component does and why I chose them below.

Data Aggregator:

- Collects and combines data from different sources or datasets.
- Consolidates the data into a unified view for reporting.
- Performs data cleansing, transformation, and merging tasks.
- Ensures accuracy and consistency in the final report.
- Simplifies the process of accessing and analyzing data from multiple sources.

Why I choose Data Aggregator:

I choose to have a data aggregator in the report controller because it helps streamline and manage the data generated by the app. It simplifies the process of collecting and organizing the data, allowing for efficient analysis and reporting. The data aggregator helps to provide real-time insights on noise levels. Overall, the data aggregator enhances the functionality of the app by centralizing data management and facilitating informed decision-making.

Chart Generator:

- Creates visual representations (graphs or charts) based on report data.
- Helps users understand and interpret information visually.
- Allows customization of chart types, styles, and data ranges.
- Enhances the presentation and analysis of report data.

Why I choose Chart Generator:

I choose to have a chart generator in the report controller because it helps to visually represent sound data and other information in the app. It simplifies the process of creating charts and graphs, allowing users to easily understand data patterns. With the help of the data aggregator, the chart generator accesses and processes the collected data. It generates visual charts and graphs of the noise levels, and safer areas in gym halls, aiding user understanding and decision-making. Overall, the chart generator enhances the app's functionality by presenting data in a simple and engaging manner.

How Data Aggregator helps Chart Generator:

The data aggregator acts as a bridge between the raw data collected from various sources and the chart generator. It collects, organizes, and preprocesses the data, making it ready for visualization. This collaboration enables the chart generator to create meaningful visual representations that enhance the app's functionality and empower users to analyze and interpret the data effectively.

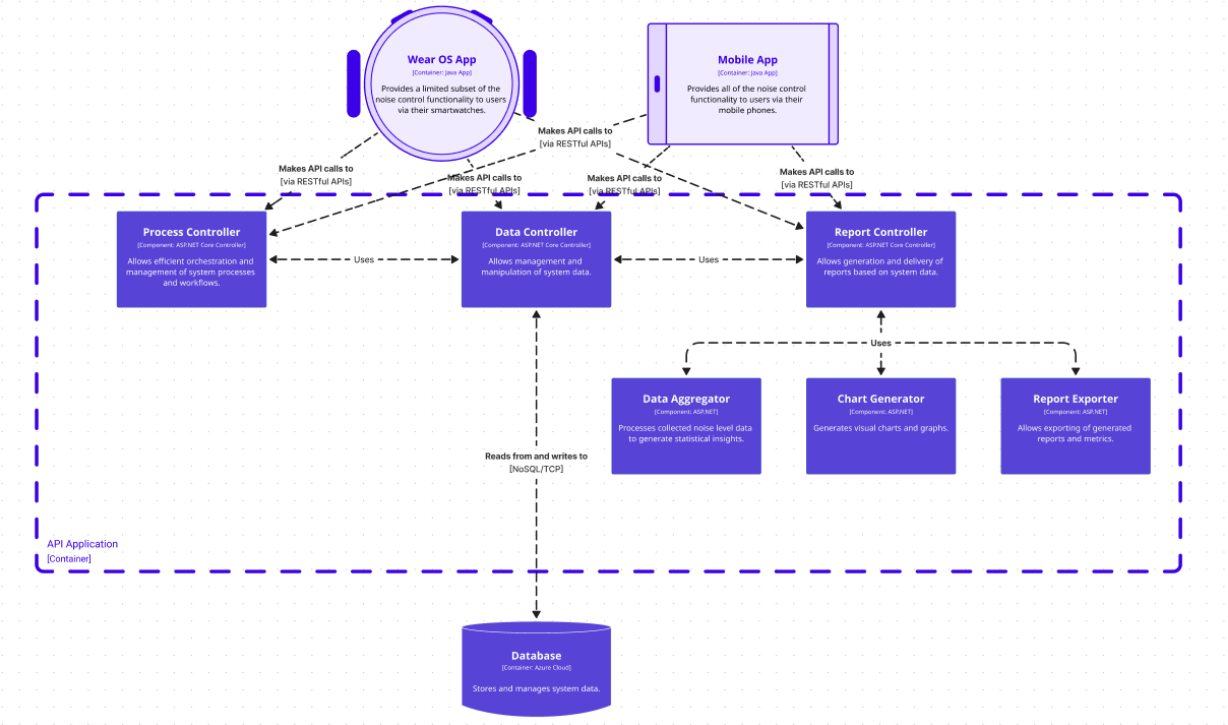
Report Exporter:

- Converts reports into different file formats (e.g., PDF, Excel).
- Delivers reports to multiple destinations.
- Provides customization options for report formatting, data filtering, and sorting.
- Enables scheduling of automated report exports.
- Ensures convenient and efficient distribution of reports.

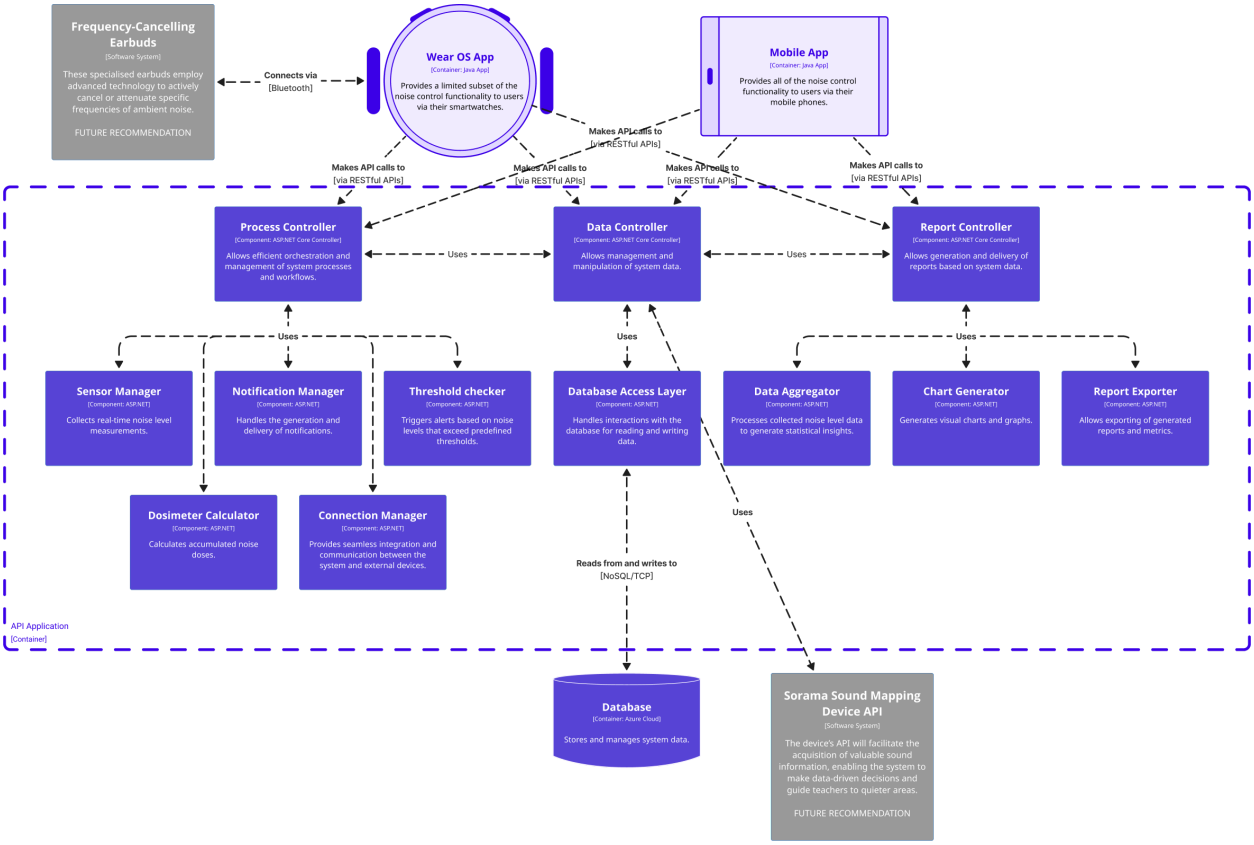
Why I choose Report Exporter:

I chose to include a Reporter Exporter in the report controller because it helps to generate comprehensive reports based on the data collected. The Reporter Exporter takes the processed data from the data aggregator and transforms it into meaningful reports

on noise levels and health risks. These reports provide valuable insights that allow users to make informed decisions and take appropriate actions. Additionally, the ability to export and share the reports enables effective communication with healthcare professionals or other relevant parties.



Complete C3:



Librarty Recommendation:

Recommendations for .NET libraries for chart generation, data aggregator, and exporting reports.

Chart generator:

The below two .NET libraries can be used to create interactive and customizable charts for the .NET applications.

High chart:

https://www.highcharts.com/?gclid=CjwKCAjwyqWkBhBMEiwAp2yUFi3DnGrhrdmcVY8oVItemhwAJdCDzGF0wUlGCQfaHR4XMuoOb2DmgRoCJ4EQAvD_BwE

Syncfusion Chart:

https://help.syncfusion.com/windowsforms/chart/overview

Data aggregator:

The below .NET library can be used to process collected data to generate statistical insights for the .NET applications.

Dapper:

https://www.learndapper.com

https://github.com/DapperLib/Dapper

Report exporter:

The below two .NET libraries can be used to edit, customize and generate reports in form of PDF, Excel, HTML, CSV, and more for the .NET application.

Syncfusion:

https://www.syncfusion.com/report-viewer-sdk

Stimulsoft:

https://www.stimulsoft.com/en/products/reports-net