**Graph Library Research**

Stress Wearables

|  |  |
| --- | --- |
| **Date** | 16-01-2023 |
| **Version** | 0.1 |
| **Status** | Concept |
| **Author** | Marinda Boshoff |

Table of Contents

[Introduction 3](#_Toc125045066)

[Research Question 4](#_Toc125045067)

[Research 5](#_Toc125045068)

[Conclusion 9](#_Toc125045069)

# Introduction

This purpose of this document is to demonstrate the research done into graph libraries for the Streaming Wearables and Stress Measurement Platform (SWSP) project. The purpose of the graph is to display a patient’s stress data and allow users to add comments to stress data points directly via the graph interface.

# Research Question

Question: What data does the Streaming Wearables and Stress Measurement Platform need to be presented and what is the best method to visualize this data to make it accessible to the end-user?

Sub questions:

1. What are the data visualization goals of the Streaming Wearables and Stress Measurement Platform?
2. What are widely used/popular graph libraries for React?
3. Which graph library suits the needs of the Streaming Wearables and Stress Measurement best?

# Research

1. What are the data visualization goals of the Streaming Wearables and Stress Measurement Platform (SWSP)?

The data visualization goal of the SWSP project is to display the stress datapoints collected from a patient’s wearable device in a graph. The graph should be filterable on e.g. daterange, type of data etc. The datapoints on the graph should be clickable in order to add comments to the stress data points. The graph should have axis and be explainable to an average health professional. It should be possible in the future to extend the graph by adding more functionalities e.g. filtering and markers. These goals were collected from the stakeholders at the beginning of the semester.

1. What are widely used/popular graph libraries for React?

Since a requirement of the project is that the frontend is built using React, research is limited to React graph libraries. Although it is sometimes possible to use JavaScript libraries in react, it can be tedious and not worth the extra time to set up. Thus only JavaScript libraires with existing react wrappers will be looked into.

To see which graph libraries are widely user and popular, I compiled a list of graph libraries from developer websites[[1]](#footnote-1)[[2]](#footnote-2)[[3]](#footnote-3)[[4]](#footnote-4) which rank React or JavaScript graph libraries.

While compiling the list I looked into each graph library’s GitHub repository to see how many weekly downloads the library has and if the library is being well maintained (when the last commit took place).

Thus, the chart libraries that will be looked into are:

|  |
| --- |
| **Recharts** |
| Weekly downloads: 1, 154, 905 |
| Last commit: 4 days ago |
|  |
| **React-chartjs-2** |
| Weekly downloads: 763,995 |
| Last commit: 7 days ago |
|  |
| **Victory** |
| Weekly downloads: 196,300 |
| Last commit: 4 months ago |
| **Visx** |
| Weekly downloads: 16,342 |
| Last commit: 13 days ago |
|  |

There are more graph libraries available, but the above-mentioned libraries are the most popular, hence it makes the most sense to look into.

1. Which graph library suits the needs of the Streaming Wearables and Stress Measurement best?

To decide which chart library suits the needs of the project best, the goals and wishes of the SWSP project mentioned earlier in this document need to be considered. To ensure the graph library will be easy to maintain and is being maintained by its developers, it is important to look at how large the community presence behind the library is and if the library has adequate documentation on its website. A well documented graph library will ensure new features can be added easily in the future.

Recharts

Recharts is the most popular React graph library and is also being well maintained according to the last commit date. The documentation for Recharts seems to be very complete. However, Recharts has fewer examples on how to customize the graph or implement extra functionalities which might make the implementation of future features more time consuming.

Chart, line chart

Description automatically generated

React-chartjs-2

React-chartjs-2 is slightly less popular than Recharts, but it is also being well maintained. Since React-chartjs-2 is based on Chartjs there is big community presence behind it and it is well documented. The documentation offers a lot of opportunities/ways to customize graphs or add extra functionalities. Therefore, this graph library seems to be a good choice for the platform.

Chart, line chart

Description automatically generated

Victory

Victory is a lot less popular than Recharts and React-chartjs-2 and is also less frequently maintained. Therefore, this chart library does not suit the needs of the project very well. The chosen library needs to be well maintained so that the project can keep using it for a long time. If the library is not frequently maintained it might cause security risks or cause the graph part of the platform to break. Victory is well documented.

Chart, line chart

Description automatically generated

Visx

Similarly to victory mentioned above, Visx does not has a lot of weekly downloads and is thus not very popular amongst developers. It is however being frequently maintained. But since there are better libraries available this library will not be suitable for the project.

A picture containing wire, flock, map, line

Description automatically generated

# Conclusion

In conclusion, the chosen graph library for the project is React-chartjs-2. This choice is mainly because React-Chartjs-2 is based on a popular JavaScript graph library Chartjs which has an large community behind it, is extremely well documented and is being well maintained. It was the easiest graph library to prototype with and has many possibility regarding features that can be added. The documentation is clear and easily understandable. Creating a graph with this library will satisfy the needs of the stakeholders.

1. https://technostacks.com/blog/react-chart-libraries/ [↑](#footnote-ref-1)
2. https://medium.com/dhiwise/top-10-amazing-chart-libraries-in-react-322cb91fad62 [↑](#footnote-ref-2)
3. https://aglowiditsolutions.com/blog/react-chart-libraries/ [↑](#footnote-ref-3)
4. https://attentioninsight.com/top-5-react-chart-libraries/ [↑](#footnote-ref-4)