Quantified Student

Dashboard Report



Created by G. Malisz

Spis treści

[1. Version History 2](#_Toc137048995)

[2. Overview 2](#_Toc137048996)

[3. Requirements 2](#_Toc137048997)

[4. Prototyping Static Layouts 3](#_Toc137048998)

[5. Dashboard Static Layouts 3](#_Toc137048999)

[5.1. First Layout 4](#_Toc137049000)

[5.2. Second Layout 5](#_Toc137049001)

[6. Dashboard Dynamic Layouts 6](#_Toc137049002)

[7. Usability Tests 6](#_Toc137049003)

[7.1. General Usability Test 6](#_Toc137049004)

[7.2. General Usability Test Findings 6](#_Toc137049005)

[7.3. A/B Usability Test 7](#_Toc137049006)

[7.4. A/B Usability Test Findings 7](#_Toc137049007)

[8. Prove of Concept 8](#_Toc137049008)

[9. Dashboard Development 8](#_Toc137049009)

[10. Conclusion 9](#_Toc137049010)

[11. Definitions, Acronyms, and Abbreviations 9](#_Toc137049011)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comment |
| 0.1 | 21.05.2023 | G. Malisz | First iteration of this document. Structure of the Dashboard development, Overview, Requirements, Layouts, General Usability Test and Conclusion. |
| 0.2 | 02.06.2023 | G. Malisz | Updated the nomenclature for usability tests, added dynamic Layouts, A/B Usability Test, PoC and Dashboard Development, updated Conclusion. |

# Overview

The following report provides an overview of the development of dashboard created to monitor and analyse student performance and its development. The dashboard is designed to provide key insights and metrics related to student progress, academic achievements, and overall performance. This report will highlight the iterative process of the dashboard design, the main components of the dashboard, the data sources used, and the key findings derived from the dashboard analysis and usability tests.

GitHub link <https://github.com/quantifiedstudent/Dashboard>

# Requirements

The requirements for the dashboard were agreed upon with stakeholder during interviews and utilizing the MoSCoW method (Must-Have, Should-Have, Could-Have, Won't-Have). These requirements can be referenced in the User Requirements Specification (URS) document, which outlines the agreed-upon features and functionalities of the dashboard. The usability tests conducted further refined the dashboard's layout and functionality to ensure it aligns with stakeholder expectations.

# Prototyping Static Layouts

Figma, a powerful design tool, was chosen for creating static layouts for the student performance dashboard due to its numerous advantages. The collaborative and cloud-based platform of Figma facilitated seamless collaboration and real-time feedback, allowing team members to provide their input and suggestions. This made it easier to incorporate corrections and adjustments based on user and stakeholder feedback, ensuring that the layouts accurately reflected their requirements. With Figma's extensive library of design elements and components, the design workflow was streamlined, enabling efficient creation of visually appealing layouts. Furthermore, Figma's responsive design capabilities ensured a consistent user experience across different devices. By leveraging Figma's collaborative features and efficient correction process, an iterative and user-centred design approach was fostered, resulting in refined and user-friendly static layouts for the student performance dashboard.

# Dashboard Static Layouts

The initial design drafts for the student performance dashboard were created with careful consideration of the stakeholder requirements and insights gathered during the interviews or inside QS team. The aim was to develop a dashboard that effectively addresses the specific needs and expectations of the stakeholder. Taking into account the agreed-upon requirements the design team generated two layout concepts for the dashboard. The design team incorporated the principles of information hierarchy, visual clarity, and responsive design. The layout designs aim to provide a user-friendly interface that facilitates easy navigation and data interpretation, while focusing on the aspects of Intuitive Navigation, Clear Visual Representations and Responsive Design.

Resources:

<https://www.datapine.com/blog/dashboard-design-principles-and-best-practices/>

<https://www.geckoboard.com/best-practice/dashboard-design/>

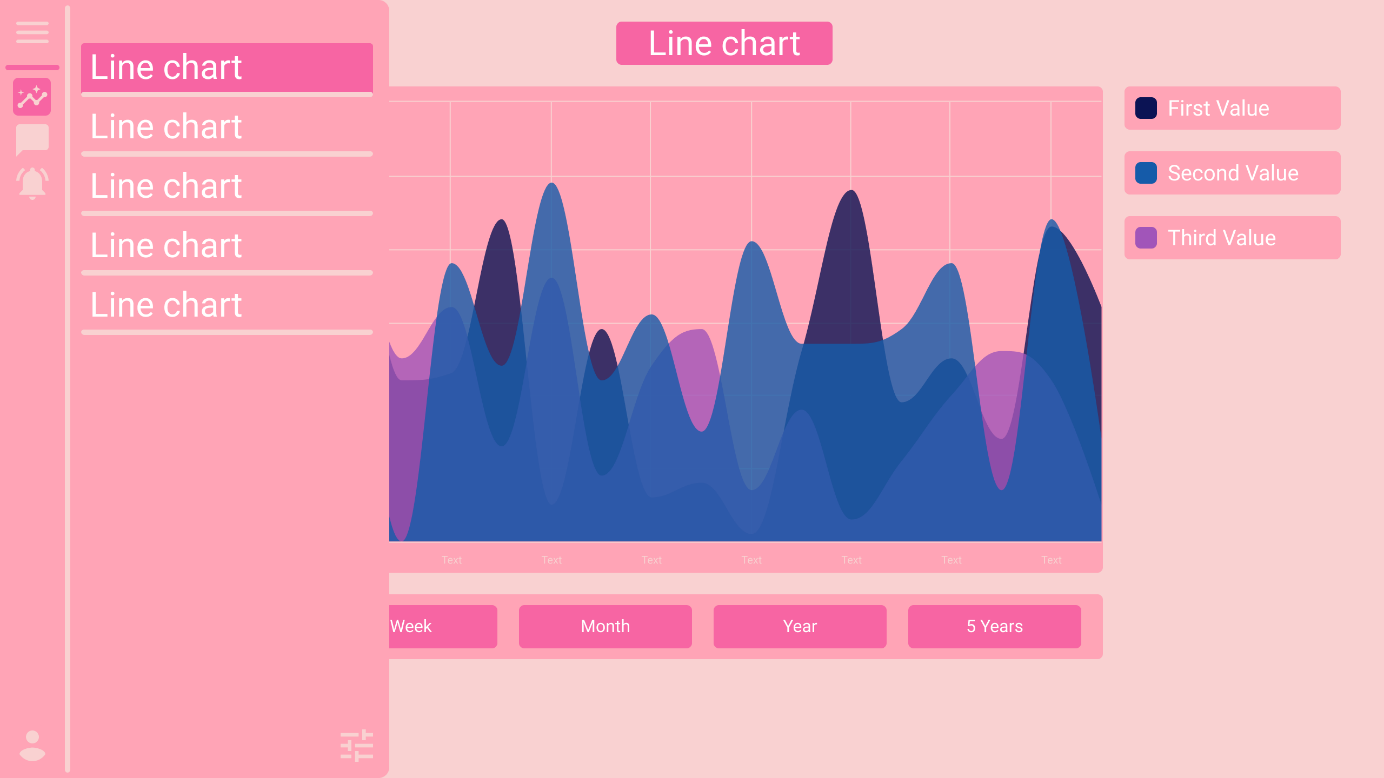
<https://uxplanet.org/10-rules-for-better-dashboard-design-ef68189d734c>

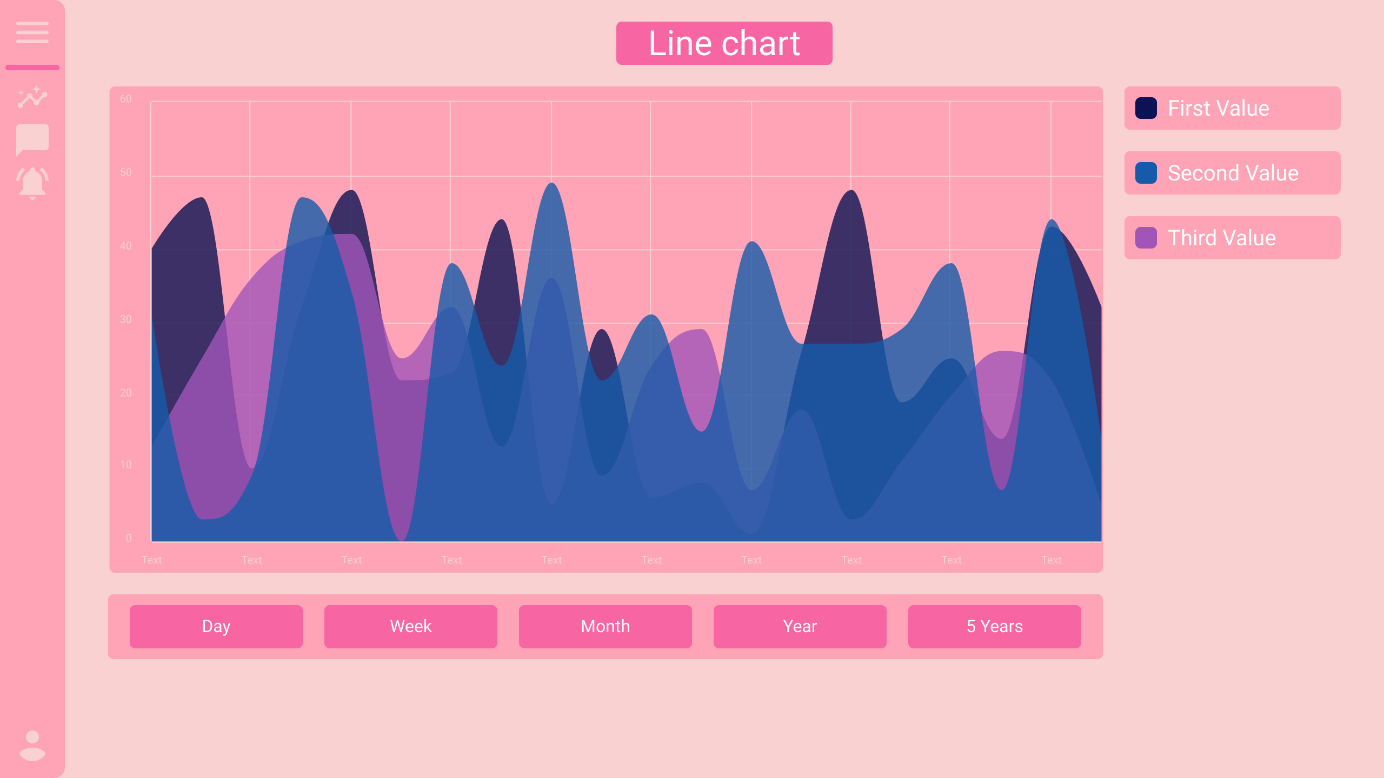
<https://dribbble.com/shots/2742901-Creative-Dashboard-Design?1464429214>

<https://elements.envato.com/dashboard-design-SYBHHQV>

## First Layout

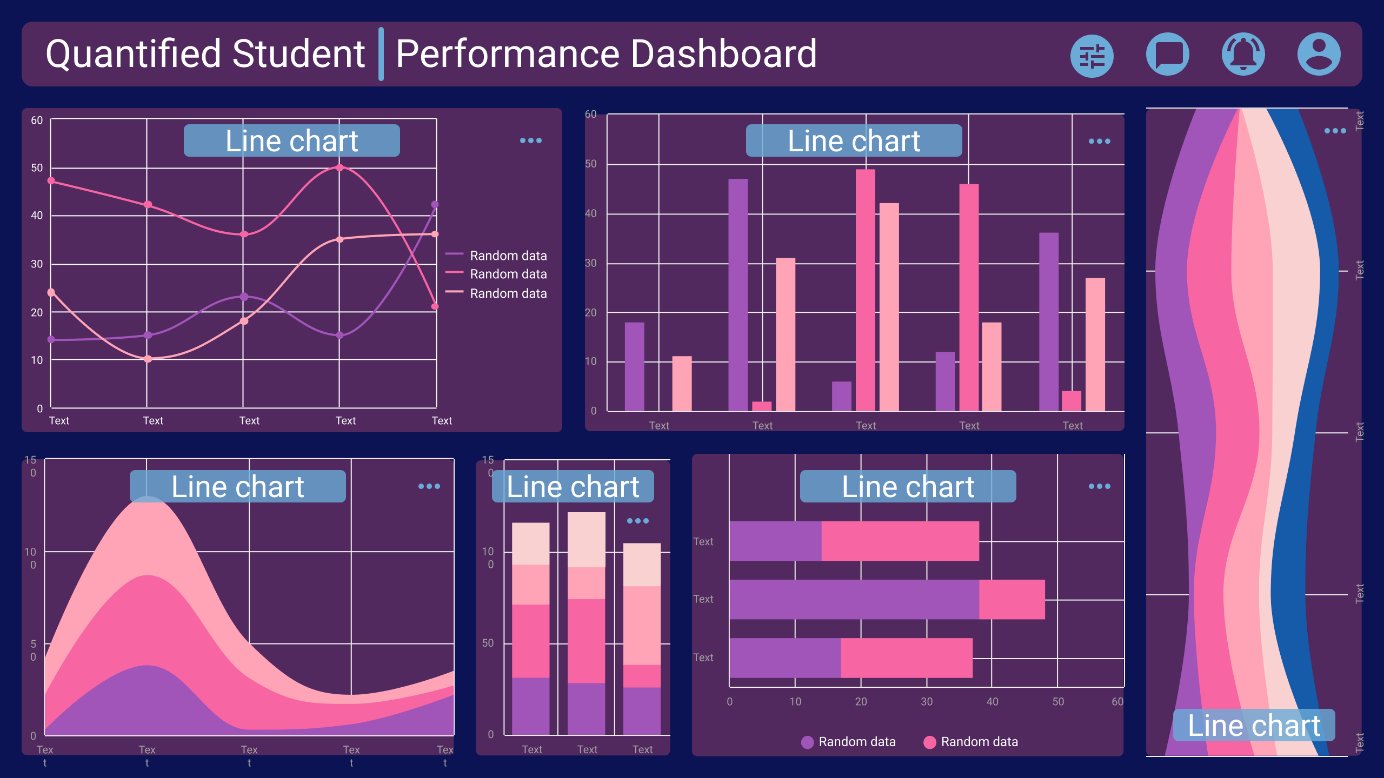
The first layout concept had a more traditional and static design approach, emphasizing minimalism and simplicity. It targeted clear representation of particular graphs or charts, allowing stakeholder to focus on specific metrics or subjects of interest. The navigation structure was intuitive, providing a straightforward and easy-to-follow layout for accessing relevant information.

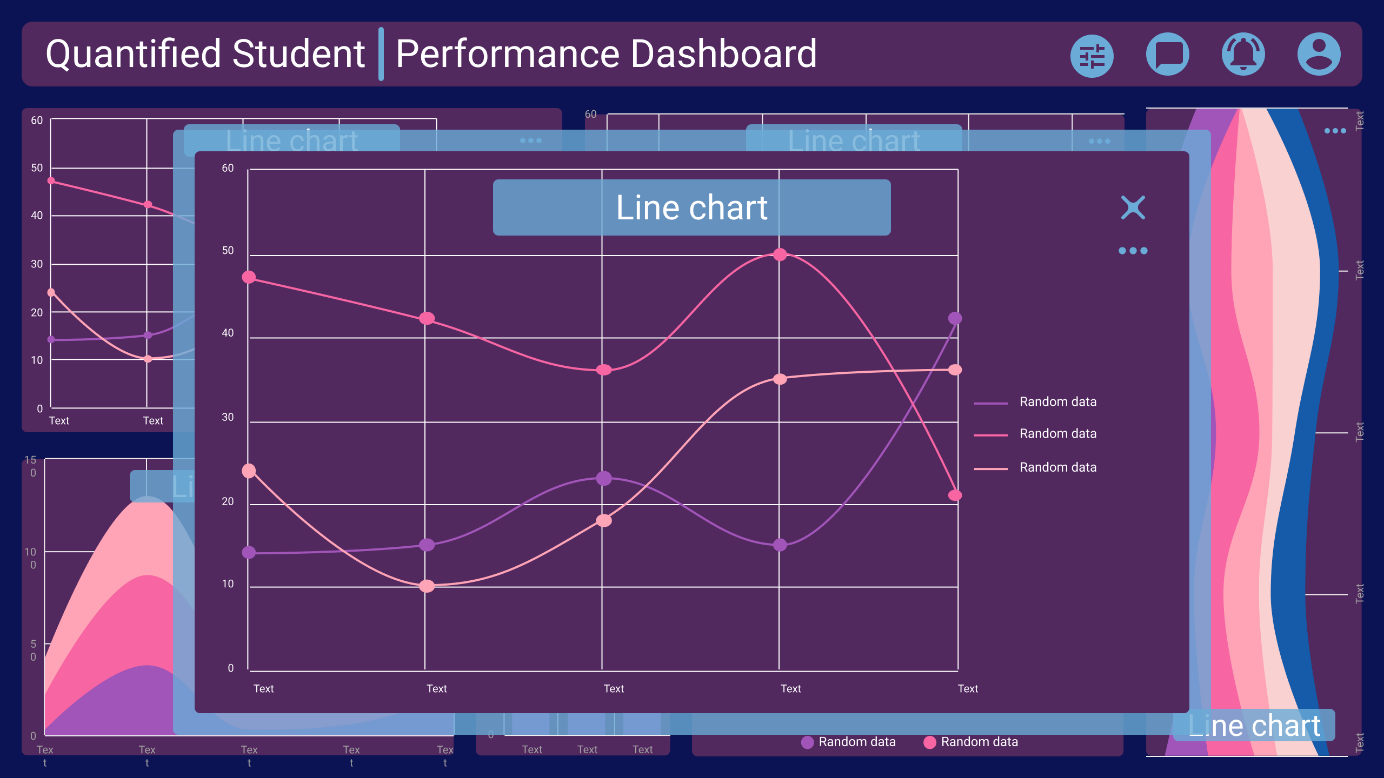




## Second Layout

The second layout concept adopted a more intuitive and interactive design approach. It featured a simple and structured layout that created quick access to all relevant information. The design aimed to be more dynamic, allowing stakeholder to personalize their view and interact with the data. Interactive elements such as customizable widgets and drill-down capabilities were incorporated to enhance user engagement and exploration of the data.





The design drafts served as a starting point for further refinement and iteration based on stakeholder and QS team feedback and usability testing results. The collaborative process between the design team and stakeholder ensured that the layout concepts aligned with the agreed-upon requirements and provided an intuitive and visually appealing interface for the student performance dashboard.

Dashboard Static Layouts can be found in the **QS - Dashboard Static Layouts.pdf** and **QS -** **Dashboard Static Layouts.fig** files

# Dashboard Dynamic Layouts

Later on, Figma's capabilities allowed for the expansion into dynamic prototypes, enabling improved A/B testing. With Figma's interactive features, designers were able to create prototypes that simulated the functionality and user interactions of the student performance dashboard. This allowed for more comprehensive usability testing, where different design variations could be compared and evaluated in real-time. By incorporating dynamic elements and interactions, designers could gather valuable insights on user preferences, behaviours, and engagement patterns.

Dashboard Dynamic Layouts can be found in the **QS - Dashboard Dynamic Layouts.pdf** and **QS - Dashboard Dynamic Layouts.fig** files

# Usability Tests

Usability tests play a crucial role in the development of any project, including the creation of a student performance dashboard. These tests are necessary to ensure that the dashboard meets the needs and expectations of the stakeholders and end-users. These tests offer advantages such as a user-centric design approach, identification of issues and improvements, validation of design choices, optimization of workflow and efficiency, and increased user satisfaction. By leveraging usability testing, designers can ensure that the dashboard delivers a superior user experience and effectively supports educational institutions in monitoring and improving student performance.

## General Usability Test

The general usability test for the student performance dashboard primarily focused on selecting the best dashboard layout and confirming the reliability and accuracy of the data sources. The test aimed to gather feedback from stakeholder and end-users regarding the usability, clarity, and effectiveness of the different layout options. Participants were encouraged to explore the dashboard prototypes, providing valuable insights on the intuitiveness of the navigation, the visual presentation of data, and the overall user experience. Additionally, the usability test ensured that the data sources integrated into the dashboard were reliable, up-to-date, and aligned with the stakeholder expectations. By validating the layout design and data sources in the initial usability test, the project team could proceed with confidence in further refining and enhancing the student performance dashboard layout designs.

General Usability Test can be found in the **QS - Dashboard General Usability Test.pdf** file

## General Usability Test Findings

The General Usability Test findings have provided valuable insights for the development of the student performance dashboard and the future development of the Quantified Student System. The test results have guided the selection of Layout 2 as the preferred option, while also identifying key data sources, such as weather, sleep, and stress, for integration into the system. These findings will drive the ongoing development process.

General Usability Test Findings can be found in the **QS - Dashboard General Usability Test Answers.pdf** and **QS - Dashboard General Usability Test Findings.docx** files.

## A/B Usability Test

An in-person A/B Usability Test was conducted to further evaluate the two layout designs (Layout A and Layout B) for the student performance dashboard. Building upon the insights gained from the first General Usability Test, this A/B test aimed to gather more detailed feedback and observations from participants. The test sessions were conducted in person, allowing for direct interaction with the layouts and facilitating real-time feedback collection. The layouts were enhanced with dynamic interactions, allowing participants to experience the interactive elements and functionalities of the dashboard. The feedback obtained during the first General Usability Test was also taken into account to address any identified usability issues and refine the layouts accordingly. This A/B Usability Test provided an opportunity to observe participants' preferences, navigation patterns, and overall satisfaction with each layout design.

A/B Usability Test can be found in the **QS - Dashboard A-B Usability Test.pdf** file

## A/B Usability Test Findings

The A/B Usability Test revealed that Layout 2 was preferred due to its improved usability and intuitive design. Valuable insights were gained for refining the student performance dashboard, based on the feedback for both layouts. These findings will guide the development process to create an optimized and user-friendly dashboard.

General Usability Test Findings can be found in the **QS - Dashboard A-B Usability Test Answers.xlsx** and **QS - Dashboard A-B Usability Test Findings.docx** files.

# Prove of Concept

To ensure the seamless integration of the student performance dashboard with the rest of the microservices and to test the effectiveness of the graph visualization, a proof of concept (PoC) was developed. The PoC served as a demonstration of the technical feasibility and functionality of the dashboard, specifically focusing on its connection with the other microservices and the accurate visualization of graphs. By creating this PoC, the development team was able to validate the communication protocols and data exchange between the dashboard and the associated microservices. Additionally, the graph visualization capabilities were thoroughly tested to ensure the accurate representation of student performance data in a visually appealing and meaningful manner. The PoC provided valuable insights into the system's performance, identified potential areas of improvement, and served as a foundation for further development and refinement of the student performance dashboard.

# Dashboard Development

For the development of the student performance dashboard, React and Typescript were chosen as the primary technologies due to several advantages they offer. The decision to use React was driven by its component-based architecture, which allows for the creation of reusable and modular UI components. This approach promotes code reusability, making it easier to maintain and scale the dashboard as the project evolves. React's virtual DOM further enhances performance by efficiently updating and rendering only the necessary components when there are changes in the data, resulting in a faster and more responsive user interface. Additionally, React's extensive ecosystem of libraries and community support provides developers with a wide range of tools and resources, further boosting productivity and accelerating development processes. Combining React's advantages with the familiarity of the development team and the compatibility with the existing Typescript codebase saved valuable time and ensured a seamless integration within the project. To leverage React's strengths for graph visualization, the Chart.js library was used, enhancing the dashboard's visual appeal and interactivity. The combination of React, Typescript, and Chart.js enabled efficient development and ensured a seamless user experience for the student performance dashboard.

QS - Graph Technology Research can be found in the **QS - Graph Technology Research.pdf** file inside **Research** folder

# Conclusion

The student performance dashboard provides a comprehensive overview of student progress, achievements, and areas for improvement. The requirements for the dashboard were agreed upon during stakeholder interviews, and usability tests were conducted to refine the dashboard's layout and functionality. The dashboard effectively utilizes data sources to present meaningful insights, enabling students and educators to make informed decisions and implement targeted interventions to improve student performance. The feedback from usability tests played a crucial role in enhancing the dashboard's usability and ensuring it meets the needs of the stakeholder. The student performance dashboard is an invaluable tool that fosters a supportive and effective learning environment, ultimately leading to improved educational outcomes.

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| URS | User Requirement Specification from file: QS - User Requirement Specification.docx |
| QS | Quantified Student |
| LTI | The integration technique that we will use to add an application to Canvas |
| DataAnalyseMicroservice | Main part of QS that is gathering and processing data |
| Microservice | Small independent services that communicates with Application |
| PoC | Prove of Concept |