Presenting Automatic Analysis Of Uroflowmetry Curves

UNIVERSITY OF TWENTE | UMC UTRECHT

H. Slagman| Creative Technology | 31-01-2020
Client: UMC UTRECHT
Supervisor: Job Zwiers
Critical Observer: Andreas Kamilaris

Table of contents

Abstract

The department of Urology at the University Medical Centre (UMC) of Utrecht has developed a classification algorithm to find and recognize symptoms in Uroflow measurement data for patients with Lower Urinary Tract Symptoms (LUTS). Our aim with this research is to create a dashboard that visualizes and explains the finding of said classification algorithm, and to implement the classification algorithm in a modular way to pave the way for further development. During this research, codesign based on concepts with the algorithms creator was implemented in addition to feedback from the head of Urology at the UMC to ensure the best possible result. The end product was evaluated using the System Usability Scale test and achieved a total score of 86.8, suggesting that the Uroflow Dashboard is clear and easy to use. The evaluation of the stakeholders and department of Urology was mostly positive, informing us that the Uroflow Dashboard did give them more insight into the classification algorithms workings. This being said, they would like to see more visual cues in the curve visualization to nudge the user to a correct flow description, even without additional textual feedback from the dashboard.

1 Introduction

1.1 THE PROBLEM

A method has been created to visualize uroflow curves, enhanced by an algorithm that points out certain areas of interest on the curve. By selecting and pointing out these symptoms to experts, the algorithm is meant to aid them in finding the correct diagnosis. The intent of the visualization should be to quickly find a diagnosis based on the algorithms output, but as it turned out, the visualization was too bulky and only led to confusion due to screen clutter, as a lot of information is being presented at once. This can also be concluded from an expert review conducted by Jeroen van Haaren. The average score based on 2 experts was a 2.58 out of 4, which seems decent, but should be improved upon drastically, as J. van Haaren suggest in the conclusion of his research. This conclusion states: “The first draft of a tool is built to perform ’component based’ analysis of uroflowmetry curves. Current performance is not yet sufficient and modifications are recommended to improve analysis.” [1]

Figure 1 A snapshot of the output of the classification algorithm [4]

1.2 DATA VISUALIZATION

Based on questionnaires that have been conducted 0n the current solution, as seen in figure 2, a lot needs to be approved upon (as mentioned in the section above). In an interview with the client it became apparent that there are not many hard requirements for the solution, as it is very difficult to create a visualization that fits the needs of all the urologist at the UMC at once. Therefore we have opted to create multiple lo-fi prototypes and combine all the best and necessary features to create one final visualization program that satisfies most.

One of the hard requirements that should be met, is that the visualization should contain explanatory feedback. This means that for every symptom the algorithm highlights, a description should be added on why the algorithm selected that particular symptom, which can be complemented with all the values the algorithm used to determine it.

1.3 THE SOLUTION

During this report, codesign was implemented using a multitude of different concepts to achieve the best fitting solution for our client. The Uroflow Dashboard. The Uroflow Dashboard was developed using a limited set of free to use and open source tools. During the development, different elements were added and removed based on feedback of our client and stakeholders, to create the best fitting concept we possibly could.

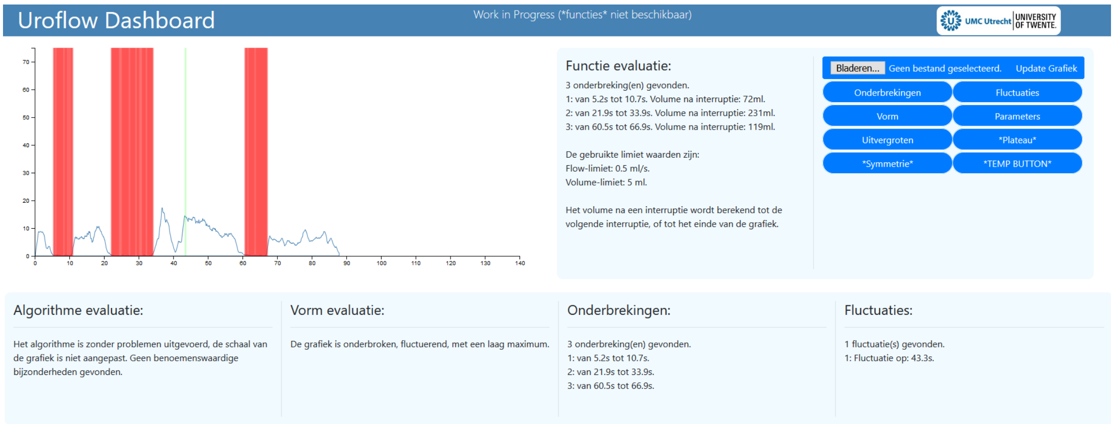


Figure 2: Uroflow Dashboard preview

As the current state of the dashboard does contain the core functionality for this use case, suggestion were made for functionality that our client would like to see in the product, if its further developed.

1.4 Research Question

Based on the introduction and problem statement, the following research question arises.

“The main task is to create an insightful and clear visualization of the classification algorithms output data to aid urologist in discovering symptoms and making diagnosis, while explaining the algorithms choices.”

This in turn, creates the following sub-questions.

“How to select and show certain symptoms based on the algorithms criteria, and give explanatory feedback based on the algorithms choices ?”

“How to integrate the algorithm in the overall application in a modular way?”

1.5 Report Outline

During the course of this report, we hope to inform the reader about the development of the Uroflow Dashboard, its design choices, and the development of the algorithm and its software engineering practicalities. First, background information on Uroflowmetry and J de Haan's classification algorithm will be presented, after which similar products will be presented and evaluated. The discovery of similar products is complemented with a literature review on data visualization in general, to find the optimal way to present data inside the Uroflow dashboard.

Next, concept design and requirement discovery are discussed. During this project, there was a lot of codesign involved with J. de Haan, the developer of the algorithm, and prof. de Kort, Urologist at the UMC. Due to this, concepts and requirements are closely intertwined, as new concepts gave rise to new requirements. After a multitude of requirements have been found, they will be summarized during specification, and functionality based on the requirements are listed.

During the realization phase, a number of data visualization tools will be evaluated, and the one best fitting our use case will be selected for the further development of the Uroflow Dashboard. On top of this, all other tools selected for the development of the Uroflow Dashboard will also be discussed. Afterwards, the basic setup and the workings of the classification algorithm are explained. Lastly, a short summary of the finished product is given.

During the evaluation phase, a multitude of user test has been conducted, and the results will be presented.

In the discussion, the results of the user test will be further discussed, and limitation of this research are presented. Furthermore, recommendations for future work are presented.

In the conclusion, all of the phases are summarized to determine if the project was successful.

2 State of the art

This chapter will focus on providing background information on the current state of solutions in visualizing uroflowmetry data. In addition to the above, this chapter will also include a literature review based on different methods that are used for analyzing and/or visualizing uroflowmetry curves. Also, a brief look will be taken into the different tools that are available to create a visualization that meets the clients requirements.

2.1 BACKGROUND INFORMATION.

At the UMC Utrecht, an algorithm has been created to aid urologist in the diagnosis of patients with lower urinary tract symptoms (LUTS). To gain a better understanding of how this algorithm works, first a basic understanding of the method of uroflowmetry is required.

2.1.2 Uroflowmetry

Uroflowmetry is an noninvasive and inexpensive way to provide objective and quantitative data of both the storage of a patients bladder, as well as symptoms that may arise during urination.[2], [3] The way this test is conducted is as follows; patients go to a special toilet, and urinate like they would normally do. The special toilet will measure the change in volume inside the toilet opposed to time. This results in a graph showing the outflow in mL/s against time in seconds, creating, in the ideal case, an arc shaped graph, a so called flow shape. From this graph, a diagnoses can be drawn based on the patterns of the graph.[4]

2.1.3 The algorithm

As there is a lack of standardization in the medical world regarding to flow shapes, they are also described very inconsistently in literature[5]. To overcome this problem, J. de Haan has created an algorithm that classifies these flow shapes to give urologist a helping hand in defining abnormalities in said flow shape. At first this was done using machine learning, but in later revisions this was changed to a limit value approach. The algorithm used to differentiate based on four symptoms. Normal flow, Staccato flow, Interrupted flow, and Long flow. [6]

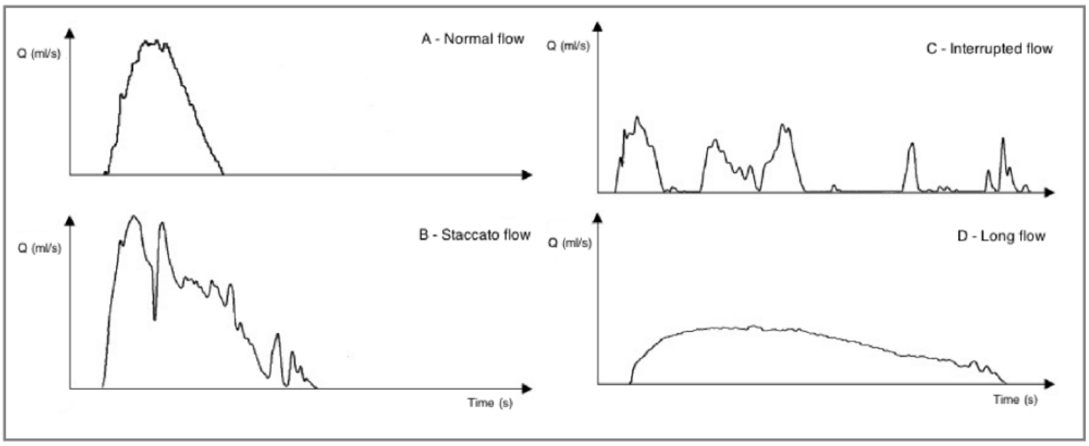


Figure 3 Different symptoms the algorithm can classify. [5,6]

Now that the algorithm has been switched to a limit value based approach, it no longer selects one of the afore-mentioned symptoms, but rather analyses the entire graph for abnormalities and highlights them. The components the algorithm uses to determine these abnormalities can be found in *table 1.*

Component	Description
Voided volume	Total amount of voided volume
Voiding time	Time between start and end of voiding
Max. flow	The maximal flow measured in a UFM measurement.
Average flow	The average flow calculated in a UFM measurement
Difference between max. and avg. flow	The average flow of a UFM measurement subtracted from the maximal flow of the same measurement.
Deceleration time divided by acceleration time	The time after the maximal flow of a UFM measurement is reached divided by the time before maximal flow is reached. DTAT ≤ 2.
Deceleration time divided by maximal flow	The time after the maximal ow of a UFM measurement is reached divided by the maximal ow. DS _ 0.6
Voiding time divided by maximal flow	The total time between start and endpoint of voiding divided by the maximal ow. TQ _ 1.
Amount of interruptions if present	The number of times the ow drops to a value below 0.2 ml/s and instants of occurrence. interruptions should be zero.
Amount of staccato peaks if present	The number of times the ow has a peak whose fluctuation is larger than the square root of the maximum flow rate [18] and instants of occurrence. number of staccatopeaks should be zero.

Table 1 Components used in algorithm [1]

2.2 LITERATURE REVIEW

As the scope of visualizing uroflowmetry is really detailed, the main focus of this literature review will be broadened to visualizing in general. In essence, this should suffice the most basic needs in researching on how to create a clear and insightful curve, where later on the functionality of the algorithm can be added. In addition to this, we will also explore some ways on how to guide users through a visualization, as too much interactivity may create confusion as well.

2.2.1 Introduction

The first step into creating a good visualization of any kind, in fact, is preventing information overload. Information overload is a very common phenomenon in our current age of information. In other words, the visualization needs to clear. If this is not the case, the information gets concealed in the data, making it a challenge for the user to actually find what he or she is looking for. [7]

One way to prevent this issue, is to give the user the possibility to interact with the visualization, opting on and off certain elements, and the ability to selectively filter on what they need. Interactivity also proves to be helpful to gain better understanding of the dataset overall, as it gives the user the ability to explore the dataset. [8]

Therefore, we will explore multiple methods of integrating interactivity into visualizations to complement the user's ability to understand, filter and select data based on their needs. To achieve this, multiple techniques will be looked into, and their similarities and differences will be evaluated.

2.2.2 Basic interactivity

This is the most obvious solution for preventing screen clutter. Filtering gives the user the option to show or hide certain elements from the visualization, reducing information overload, thus also reducing cognitive strain. A few methods used by Nikolaou et al. to reduce cognitive strain are only presenting detailed data once the user clicks a point of interest, and enabling or disabling certain options [9][8].

Lixin et al. [10] describes a method of filtering which is used a lot on spatial data. This method creates a viewport, which decreases the overall size of the visualization to a selected region, hiding all other information the user is not interested in. This can easily be transferred to a curve like visualization, by extending the start and end of area of interest of the curve over the whole visualization.

Another way of doing this is zooming. Zooming however, effects the whole visualization, instead of selecting a region. Zooming is also described as changing the position and size of data-elements on the screen [11]. Important is, when zooming in and out, the visualization should, except from it scale, never be altered. [12]

Selection is also a good example of use of interactivity, and can be used in different ways. Keim [13] mentions selection as a subset of filtering, however, Yi et al. [14]describes it as marking certain data points to make them stand out, giving the user the ability to keep track of them.

2.2.3 Additional information and labeling

To provide a clearer insight into the visualization, the visualization should be able to provide more information when the user requires this. This can be done by selecting one or multiple points of interest and labeling them with additional data, or show detailed data of the selected point(s) on the side. [15][16]. This is different from filtering in such a way that to show this data, an explicit action is required, and not all of the specific data for every point is available by default. Keim et al. also mentions zooming as a way to provide additional information, only presenting more detailed data when the user could want it, and when there is enough free space in the visualization to show it.[13] Selection could also be used to add more detailed information to certain points marked by the user. This provides extra information where desired, and does not clutter the screen as much as displaying detailed data for every point [14]

Labeling, in its core, it also used to be able to differentiate between data. This can be done in many ways. For example, by shape and by color. [17] This means that multiple dataset, and or supplements to a dataset can be integrated into one graph, without creating confusion. This again, can be combined with filtering, to turn on or off this supplemental data. This technique is also used by Tam et al. based on color and shape and Schäfer et al. based on shape. [18]

2.2.4 Guiding to interactivity

Based on research done Blascheck et al[19]., there are a lot of ways to guide users through interactivity that can be implemented into visualizations. In this research, a study group was asked to interact with an amount of different visualizations, and their fixations were recorded. Ideally, we would want the user to focus more on the graph itself, as opposed to the menu, controls, and/or documentation. The methods explored are:

- Eyes only
- Reading text
- Opportunistic interactions
- Entry points
- Structural interactions
- Permutation interactions
- Leveraging the familiar

In this review, the main focus will be on the opportunistic, structural and permutation interactions. These are more active interactions, where others are either passive or don’t have added value for our own research.

2.2.4.1 Opportunistic interactions.

Subjects that interacted opportunistic typically select an arbitrary control and examined the chart to see what changed. This is more an explorative action to familiarize oneself with the controls, and is something we should avoid. This means the users spends more time “playing around” with the controls then actually looking at the chart. Results confirm this. One way to avoid this is to have limited and very intuitive controls, preventing explorative behavior from happening.

2.2.4.2 Structural interactions.

Much like opportunistic interaction, however, in a structured fashion, starting with one group of controls, exploring all option, and moving to the next group of controls. Again, this is enforced by explorative behavior and should be prevented. However, the results yielded by structural interaction are more desirable then those by opportunistic interaction, based on time spent looking at the graph. One could argue that this is proof in favor of grouping control based on control-impact.

2.2.4.3 Permutation interactions.

Based on structural interactions, mainly focused on tying all combinations of controls. This again enforces explorative behavior and can be prevented by making all controls as independent from each other as possible.

2.2.5 Conclusion.

Multiple ways of adding interactivity and the uses of controls to use that interactivity have been explored, and evaluated. There are many different ways to aid users through a visualization.

Zooming provides more detail of the shape of the graph, while remaining utterly truthful, as both the x,y-scales change accordingly. Closely related to this is the viewport solution, which only affects the x-scale, with its goal to let the user focus on a small part of the visualization. Both can be used to add more detailed data to the graph. Filtering and selecting provides the user with many different options to change the appearance of the graph, and reduce cognitive strain.

One might not suspect it, but according to Preece[20], the order in which these methods are implemented also matters. Based on her research, she suggest “overview first, zoom and filter, then details-on-demand”

Based of the research by Blascheck et al. [19], it is essential that all controls, such as filtering, zooming, and opting on or off certain options, are grouped together in their respective (sub)-menus, and should work independent form each other. This is to limit explorative behavior and maximize the visualizations efficiency.

It has yet to be determined which interactivity works best for our client, and this is why we propose a case study for further research, providing our client with different visualizations, containing different combinations of interactivity. These can be later on combined to one visualization.

2.3 RELATED WORK

In this section, related works and projects will be discussed. Given the detailed medical scope of this research, papers describing solutions will, which may or may not have already been finalized to working projects, will also be discussed.

2.3.1 Voice activated Uroflow Application [21]

Dashbouquet Development is a company based in Belarus, and developed a voice activated Uroflow application. The way this application takes it measurement are by sound acquisition. The accuracy of this is questionable, but as we are mainly after visualization, we will leave measurement methods and algorithms out of consideration.

Their visualization is fairly standard to that of a normal Uroflow graph, with the exception of colored ticks at certain values for the ml/s axis.

Figure 4, Voice activated Uroflow Application [26]

Dashbouquet also provides a history of all measurements, which can be found in the history and the diary section.

The main difference between the two aforementioned sections is appearance and data density. The history tab contains a short overview, containing measurement points with their corresponding volume, max flow, and time of measurement. The diary sections adds to this the average flow. History of previous measurements could be of interest to us, if the user requires an easy way to compare measurements.

Furthermore, the overall look of the application feels quite professional. A huge benefit of the main screen (far left in figure 3.) is the easy overview of real measurement data complementing the graph. Adding this more detailed data by default without cluttering the screen could be to an advantage for our clients.

2.3.2 P-Flow Diary [22]

P-Flow Diary is a system designed by Kazuyoshi Johnin, Kenichi Kobayashi and Akihiro Kawauchi. The product was designed to easily discover lower urinary tract symptoms (LUTS) in children. The product can easily be used anywhere and can easily record all uroflows and simultaneous frequency volume charts (FVC) at home.

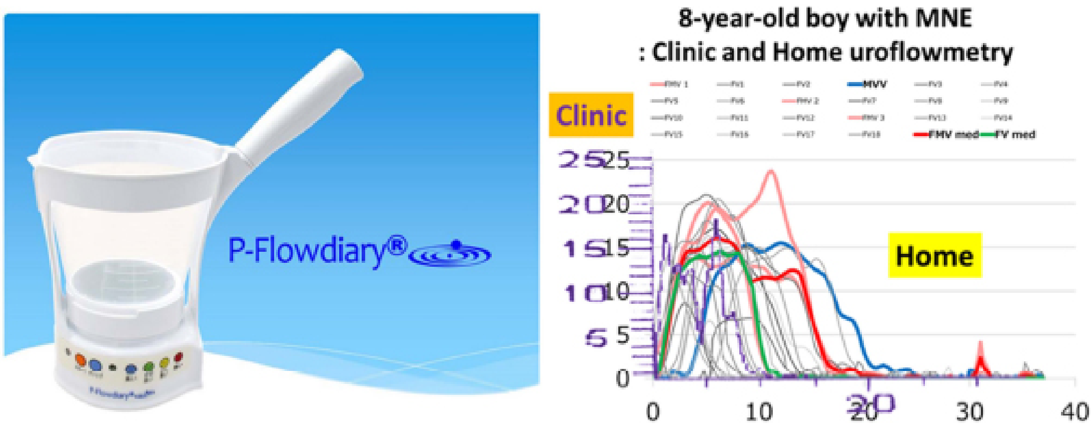


Figure 5 P-Flow Diary Product and data visualization.[27]

The product seems to be a very novel and useful idea, the way of visualizing the data leaves something to be desired. Graphs are all over the place, many graphs cross over each other without a clear way of telling what is what. There is too many information contained in one visualization. Furthermore, there are 2 sets of axis drawn over each other, with no clear indication of what axis belong to which graph. This stresses the importance of being able to select data to be shown and filtering as very important options to be contained in the design process of our clients data visualization.

2.3.3 Uroflow Tracings [23]

Uroflow Tracings is another application where sound is used to acquisition an Uroflow graph. Uroflow Tracings however, differs a lot in looks from the Voice activated Uroflow Application.

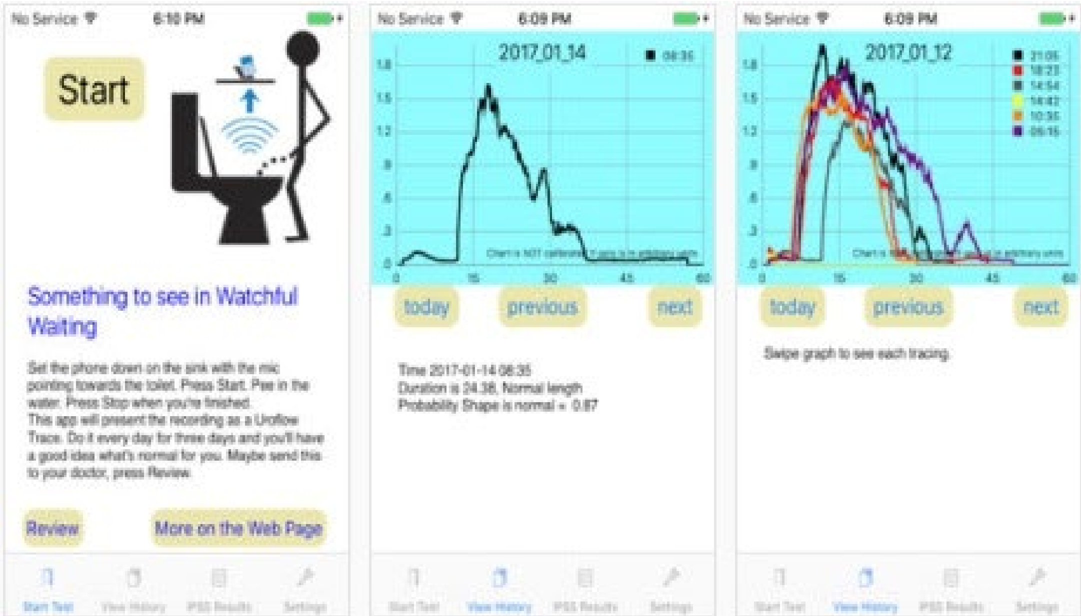


Figure 6 Uroflow Tracings snapshot [28]

Where the overall look is less professional, the function to draw previous graphs over the current graph for easy comparison, can be very useful. As opposed to P-Flow Dairy, Uroflow tracings graph, will it may get a bit confusing when too many graphs are drawn over each other, looks much clearer and easier to understand. Axis all use the same values, graphs are distinguishable enough to tell them apart where it matters, and time of measurement is included in the legend.

Again, I believe the comparison part is key here. This time however, it's also visualized, instead of showing raw measurement data. Other ways of marking, labeling and/or filtering could improve the cognitive strain factor of this visualization by making graphs a bit less opaque, and showing key datapoints of each graph.

2.3.4 Minze's Hospiflow and Homeflow [24]

Minze has developed a special toilet accessory that can be fitted on any toilet and is able to take uroflow measurements. Accompanying this toilet accessory are 2 types of dashboards. The Minze Hospiflow and the Minze Homeflow. These two work closely together, as data from the Homeflow can easily be passed on to the users physicians Hospiflow

Hospiflow integrates patient details and uroflow data into one clear overview. It contains history, detailed data and uroflow graphs all in one overview.

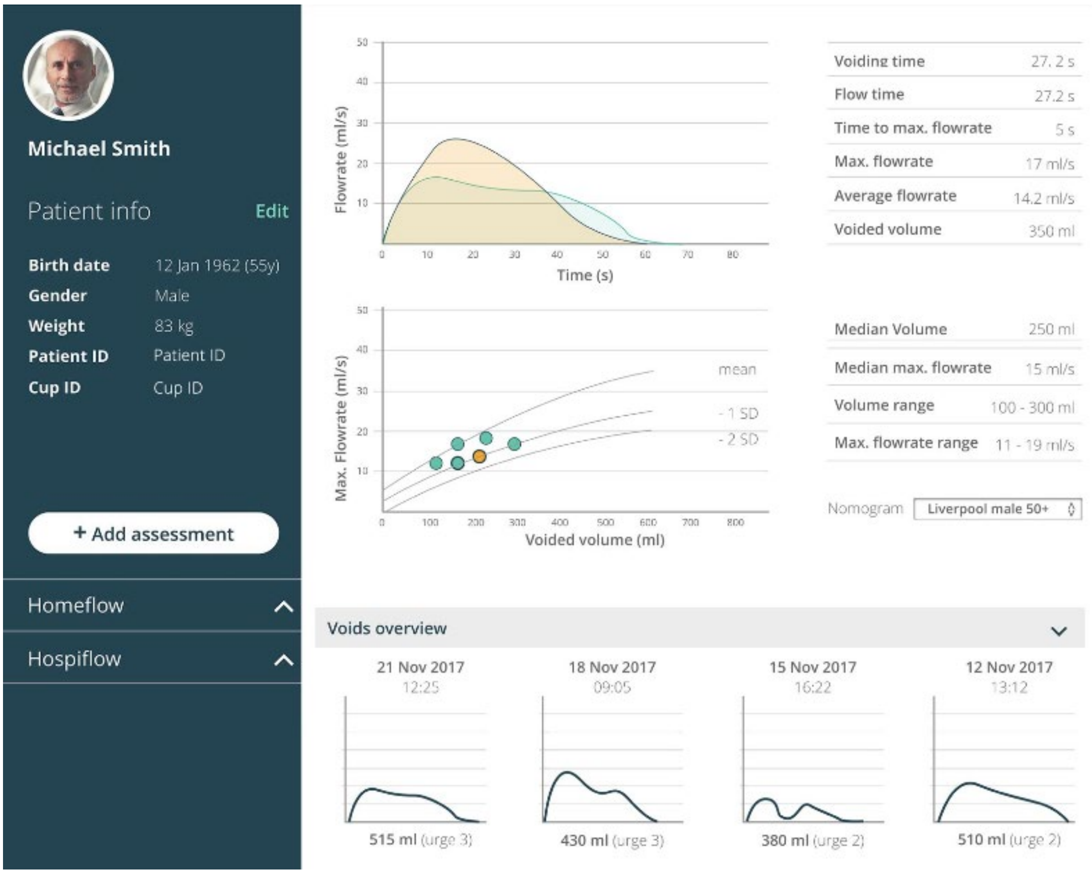


Figure 7 Minze's Hospiflow dashboard [29]

As adding history is based on our users requirements, its benefit can only be speculated. The statistics for any current measurements however, can be very useful to verify measurements and to identify potential false readings. Furthermore, these statistics can also be compared to statistics of earlier measurements, allowing easier comparison than comparing hard data. Statistics are also being used by Porru et al. [25], to find symptoms in uroflow data, and for easy comparison between home and office uroflowmetry. No visual representation were included in this paper.

While statistic may be good indicator, I still strongly believe in the ability to draw multiple graph over each other for comparison, something that is missing in Hospiflow.

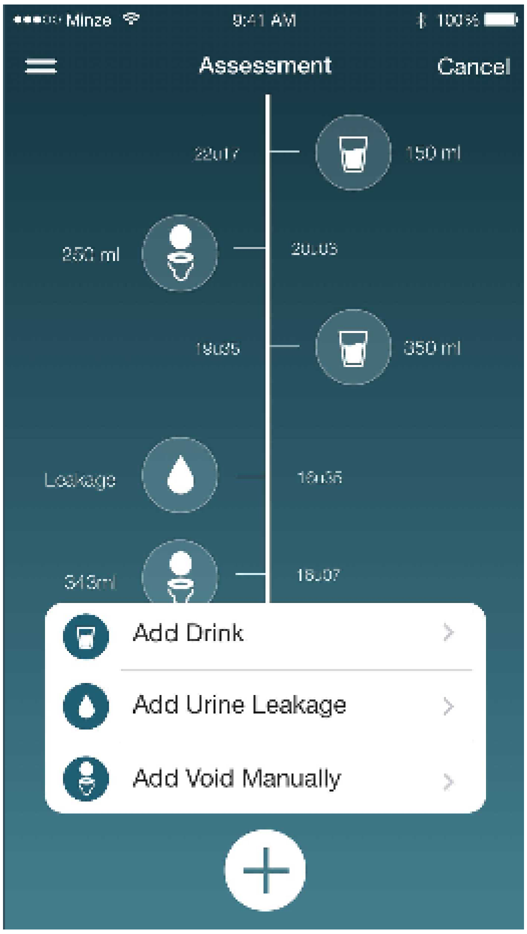
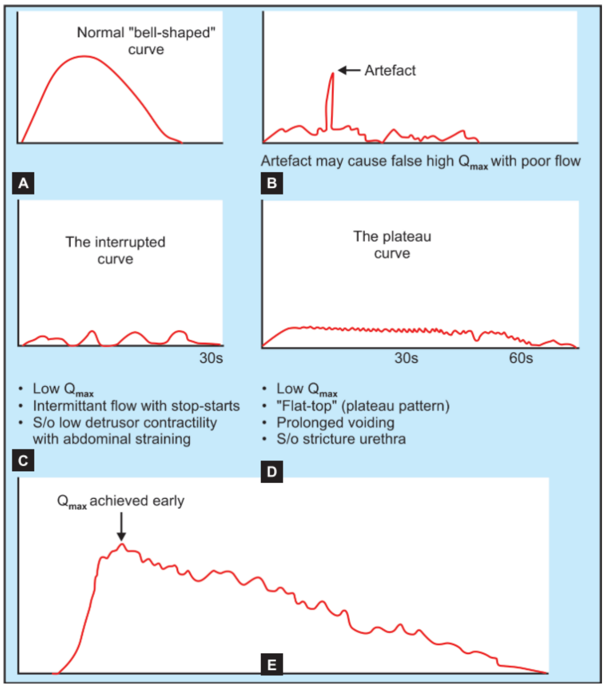


Figure 8 Adding entries into Minze's Homeflow [29]

Overall the graphs that are present are very clear, and everything a physician could need to identify lower urinary tract symptoms is there, in detail, and in a very clear and professional looking overview.

Homeflow adds to this the option to add multitude of other entries to complete a patients diary, such as fluid intake, and urge to pee during a measurement. This is a very interesting take on adding more diagnostic data to uroflow measurements, but lies outside the scope of our research.



2.4.5 Urology for nephrologist.

In this research paper Arora et al. [26], they discuss the core of urology, its main symptoms and possible diagnosis for lower urinary tract symptoms. They mention ways to easily discover the severity of symptoms using rule of thumb, and talk about some underlying causes of these symptoms. The thing we are most interested in, is the way these symptoms are visualized.

Figure 9 Symptoms in Urology (Urology for Nephrologist) [26]

What stands out to me here, is the fact that there are no ticks on the axis, with the exception of the occasional tick mark on the horizontal axes. Further research into this paper has lead me to believe that exact data is not explicitly required to set a preliminary diagnosis, as long as the graph is not distorted and the scale of the axis always remains the same. If the scale of the axis were to change in an unparallel fashion, it would distort the graph, and no reference can be made to non-distorted graphs. This visualization could be seen as an advocate for keeping detailed data to a minimum, and only display it when the user requires. This would lower cognitive strain and prevent screen clutter.

2.4 CONCLUSION

There is no universal way to visualize uroflow graphs, although a lot of visualizations have similarities. What we have learned from our state of the art is that under any circumstances, the visualization needs to be clear, effective, and easy to use. Furthermore, we should refrain from adding to much information to the graph, to prevent confusion and/or graph distortion.

Needless to say, the amount of explorative behavior should also be kept to a minimum. One way to achieve this is by using control grouping, and/or limiting the amount of interactivity to functions that are pure necessity.

3 Concept design and requirement discovery

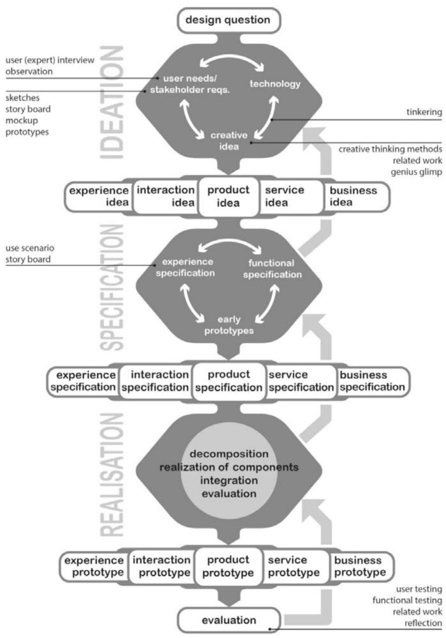
In this section, the design process will be discussed. After that, different designs will be brought to light, specifically tailored to our client's needs based on found requirements. These designs will be evaluated and adjusted based on feedback from the client. Lo-Fi prototypes will be created using Adobe XD, a program that allows you to create high end graphics and UX design with low level interactivity with relative ease. This should allow us to evaluate the first designs based on look and feel, without the extra effort of implementing the algorithm in each version.

As the requirements where not clear at the start of the project, we have opted to present the client with different concepts and designs, to narrow down options and find out what it really is our clients wanted. We understand that this might create some confusion during the reading of this paper, therefore we have included the following structure.

All concepts are listed and evaluated and the requirements that were discovered during feedback sessions are listed underneath each concept. Furthermore, a list of revised requirements is added in section 5.1, Revised requirements.

3.1.1 THE CREATIVE DESIGN PROCESS

For the design process of this project, the Creative Design Process was implemented. [27]



Foremost, the design is mostly focused on the framework, as opposed to the visualization itself. The main reasons for this was strict user requirements, which will be discussed later, and the fact the user already knew which data visualization was to be used for the data itself, namely a line or area graph.

The main core of the Creative Design process is a huge aid in the design, as it forced reiterative thinking. Based on this reiterative thinking, requirements came to light, that otherwise would be unthought of. These requirements are more design objectives, such as; "Buttons are preferred at the right side of the screen".

Figure 10: Visual representation of the Creative Design Process [27]

After a discussing a few concepts with the client, the iterative way of work of the Creative Design process again presented itself valuable. Using this process, new and improved concepts could be created in a rapid fashion. The core role the design process had in this was to make minor changes based on the most approved upon design, to find out where we could approve even more.

3.1.2 MOSCOW TECHNIQUE [28]

Another strategy that was implemented during the design phase is the MoSCoW-technique. This technique used the requirements derived from the Creative Design Process and expanded upon them.

The Moscow-technique forces prioritization of requirements and brings other requirements forward during execution. The main purpose of this technique is to reach a common understanding with the stakeholders on the importance of each requirement. This is done by pursuing the following principles.

- Must Have
- Should have
- Could have
- Wont have

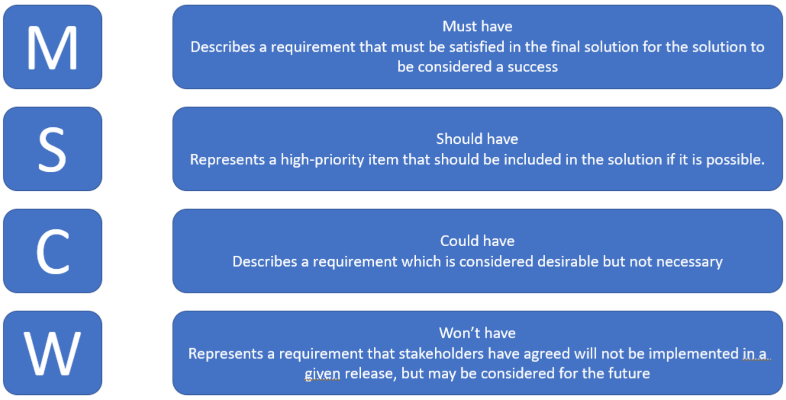


Figure 11: MoSCoW Technique [53]

Must have requirements can easily be identified by answering the question, "What happens if this requirement is not met?". If the project comes to a halt, this is a must have requirement. Other ways of identifying must-haves are things like, the product would be unsafe without this, or even illegal, etc.

Should have requirements are requirements that are optional, but really should be implemented when given the chance. They are vital solutions to certain problems and very desirable, but the project will function without these requirements in place.

Would have requirements are similar to should-haves, but would have less impact if they are left out of the project. They are more a kind of "would-be-nice-to-have" requirement, but it wont be too big of a deal if these requirements are not fulfilled

Wont have requirements are requirements that would not have popped up during the implementation of the Creative Design process. Normally, if we think of requirements, we think of what the client wants, not what the client doesn't want. Wont-have requirements helps us to steer the project in right direction by avoiding pitfalls that the client would dislike, which in turn might lead to a delayed implementation.

During the remainder of this paper, requirements that were identified during this project will immediately be coded with either **MH, SH , WH** or **NH** to prioritize them in Must Have, Should Have, Would Have or Wont Have (Not Have) requirements.

3.2 STAKEHOLDER INTERVIEWS AND REQUIREMENTS.

After an explorative interview with both J. de Haan and prof. De Kort, which is summarized in Appendix A, on their views on the further development of the interface of the classification algorithm, one clear requirement presented itself. The old interface (Figure 1) needed to be more clear, and needed explanative feedback. This introduced the following design statement, which is also the main research statement:

"The main task is to create an insightful and clear visualization of the classification algorithms output data to aid urologist in discovering symptoms and making diagnosis, while explaining the algorithms choices."

One other requirement that was very clear from the start was that the graph that was to be presented needed to have a constant scale under any circumstances, to prevent warped views. This warpage can be best explained with the aid of an image.

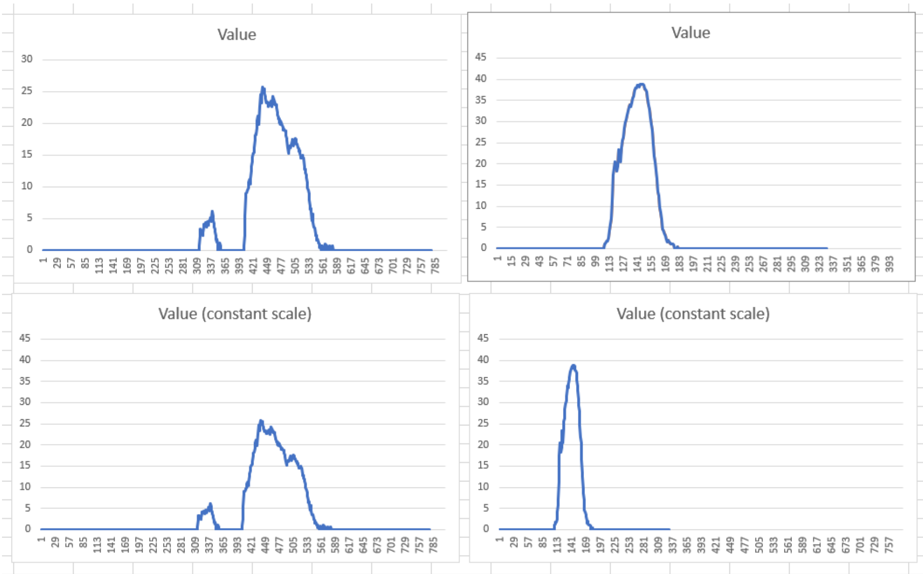


Figure 12: Warpage in non-constant scale graphs

As can be seen in the image above, the topmost graphs look quite similar, but the bottom graphs have more distinctive features between them. The topmost graphs use an ideal axis scale for the data they contain, as the bottom graphs use the same scale. Using a non-constant scale can pose a huge risk to a diagnosis, as the axis might be misread, or the attending urologist might not realize the scale of the axis has changed. Furthermore, non-constant axis also prevent easy comparison with reference data. The use of non-constant axis should be prevented at all costs. More on risk can be found in appendix J and section 8.4 Ethical evaluation . , where an ethical evaluation report can be found.

Requirements at this stage of the project:

- Clear and insightful with explanative feedback. **MH**
- Constant scaling of the axis on graphs that are used. **MH**

An overview of all final requirements will be presented later on.

Based on these first interviews, it was not yet very clear in which direction the whole design process was headed. To narrow this down, a first concept was made to bring up more requirements based on the feedback.

3.3 CONCEPTS

3.3.1 Method

In this section, the first concepts and will be presented and evaluated. The tool that was used for the creations of these concepts is Adobe XD[29]. Adobe XD allows for rapid prototyping by drawing elements and adding quick and simple interactivity. As Adobe XD is mostly used for UI/UX design in apps and websites, it should fulfill all our needs, as the dashboard itself contains a lot of UI design.

Adobe XD also allows us to upload our concepts to the cloud, where our clients and stakeholders easily can interact with the concepts on any device. This does not require them to download the program itself, as it is al presented in a browser window. These so called demos give a good representation of an eventual end product, as they also allow the user to interact with the concept, like they would in the final version of the project. Furthermore, they can give direct feedback on any element they like, or dislike, or overviews in general. This feedback is linked to the elements, which greatly increased the speed of the iteration process, as we can easily identify elements that need to be changed.

Links to the online demos of the Adobe XD concepts will be listed in Appendix C: Online resources.

All of the concepts were the product of codesign with J. de Haan and his supervisor prof. De Kort, as they provided a lot of feedback and other solutions. J. de Haan also provided the classification algorithm which is to be integrated into all of the concepts. The concepts can be found in section 3.3.3 trough section 3.3.7

3.3.2 Evaluation criteria for concepts.

Here the different evaluation criteria for the first concepts will be listed and explained. Based on these criteria, areas can be found that need to be improved upon, find features that are currently missing that should be added, or even current features that need to be discarded.

- Usability
    - Number of clicks
    - Overall ease of use of the application
- Clarity
    - Screen clutter
    - Clarity
    - Amount of feedback
- Interactivity
    - Control grouping

3.3.3 Concept 1

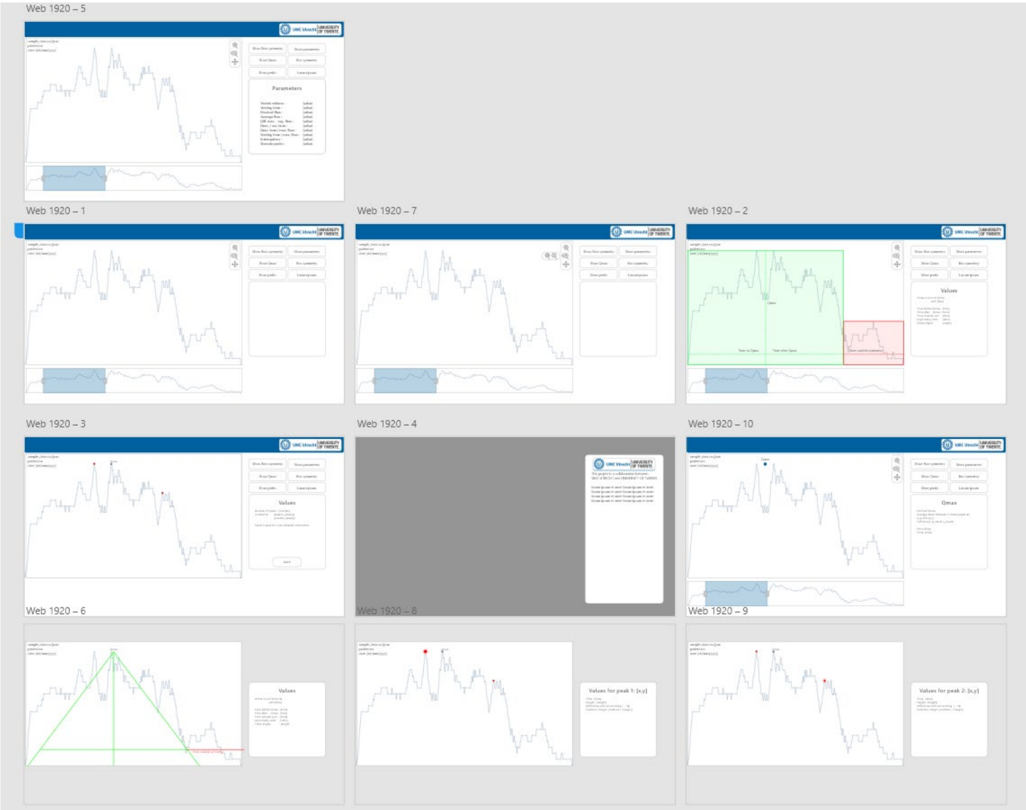


Figure 13 Concept 1 overview (Adobe XD)

Concept one features the most needed features in a practical fashion. Detailed data is accessed by using the buttons on the side panel, without cluttering the graph with extra information, with the exception of certain timings.

Furthermore, a small amount of control grouping is implemented. Extra zoom options are available under the “reset-zoom”-button, such as, reset zoom entirely, or undo last zoom/panning action.

This prototype contains a small amount of actual graph interactivity, as the buttons will display most of the information. In this graph, the user can select different staccato peaks (if present) to display more detailed information.

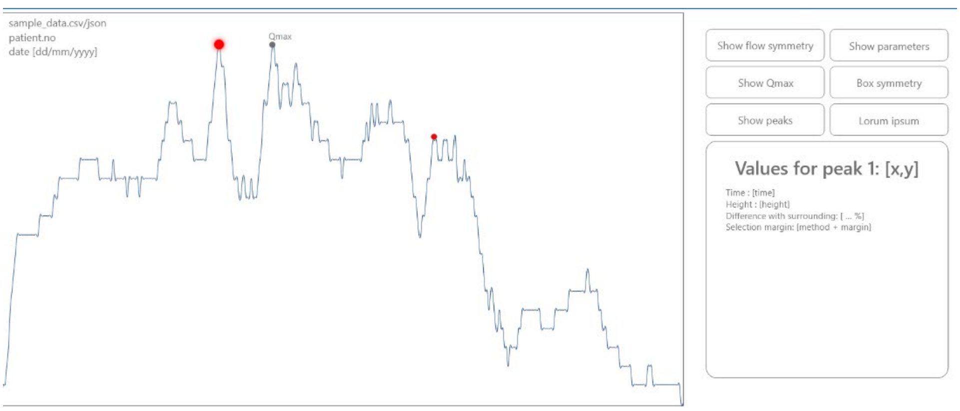


Figure 14 Detailed staccato peak information after selecting a peak (Adobe XD)

Figure 15 Box symmetry and context graph (Adobe XD)

At the bottom of the screen, a context graph can be found. This is a graph that has been scaled down along the y-axis, to allow the user to see at which part of the graph they are looking at, after zooming and panning has occurred. The "real" graph can also be adjusted by using the handles on the context graph.

As of ease of use, all parameters can easily be accessed by clicking the "Show parameters" button. This will show all relevant parameters considering the current dataset. Furthermore, calculations done by the algorithm will be shown and explained in the textbox at the side panel. Furthermore, all buttons use globally recognized icons, which are self-explanatory. Examples of this are the magnifying glass for the zoom-button, and the crossed arrows for moving around.

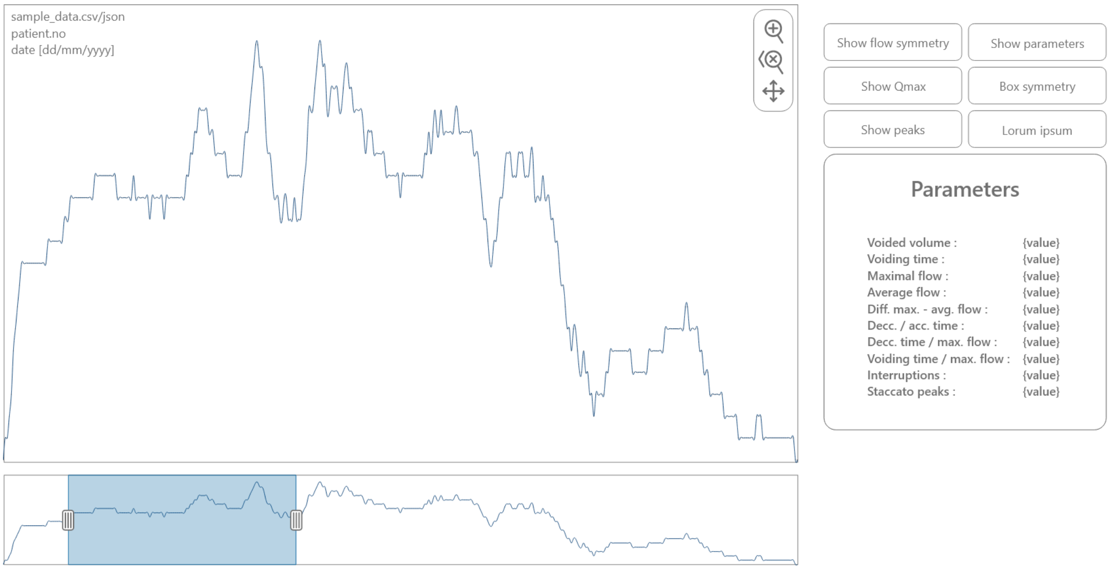


Figure 16 Parameter overview (Adobe XD)

3.3.4 Concept 2

The second concept was designed parallel with the first concept. Based on the fact that most diagnoses will be done on a computer (as we are not developing a mobile version of the dashboard), there is a lot of screen real estate, meaning we have much more space to visualize things on a PC, as opposed to a mobile phone screen or tablet. This screen real estate can easily be accommodated by implementing bootstrap [30] in the dashboard. In a future version of this concept, bootstrap will allow for responsive resizing and placement of graphs and or elements.

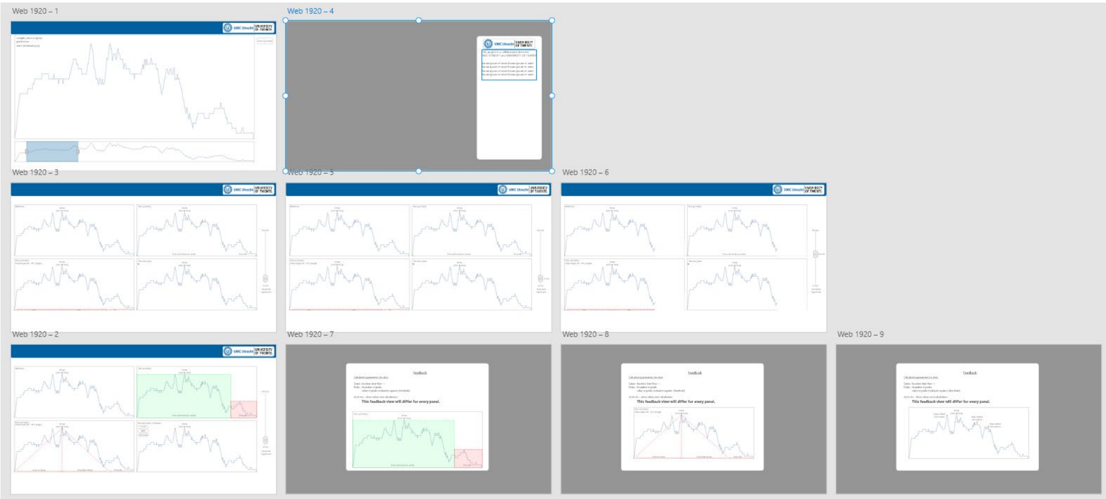


Figure 17 Concept 2 overview (Adobe XD)

Using all this screen real estate, ease of use was prioritized in this concept, removing as much interaction as possible, reducing the number of click to the lowest amount possible. We have shifted from one graph with buttons and parameters on the right, to four graphs displaying all kinds of different information. The information shown in the four graphs in Concept 2 is explanative feedback that in Concept 1 can only be reached after pressing a button.

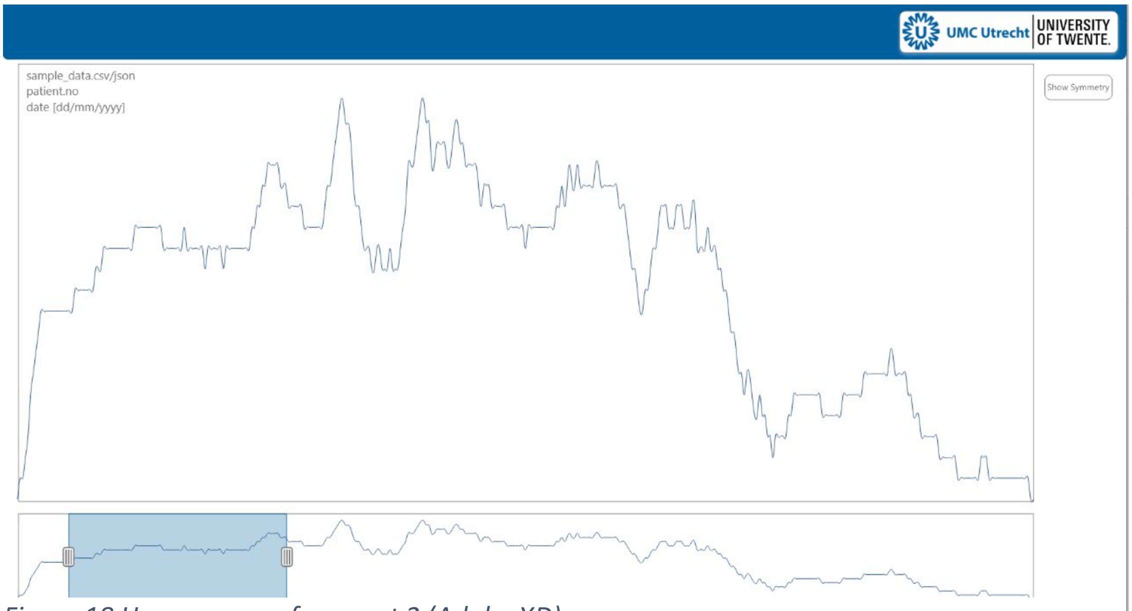


Figure 18 Homescreen of concept 2 (Adobe XD)

Another feature that was introduced in this concept is the limit slider, to change the behavior of the algorithm during execution and evaluation of the data. This feature was heavily requested, as the classification algorithm is still subject to change.

The last extra feature that was added was a way to toggle between different methods of calculation for finding staccato peaks. During this stage, we found out that the 3 most common methods, as described in the classification

The concept starts out with one big graph for easy superficial diagnosis and inspection. Next to this big graph, there is a button that will take the user to the overview screen containing the four graphs, with explanative feedback of the classification algorithms choices.

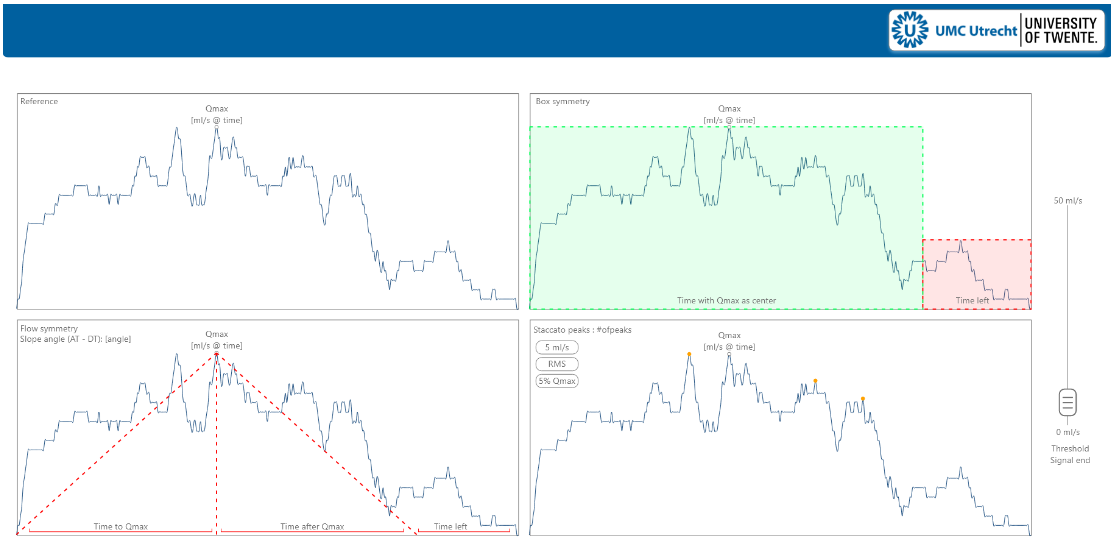


Figure 19 Detailed overview of concept 2 (Adobe XD)

Behind this detailed graph, another layer of complexity can be found. When clicking on one of the four graphs, a window pops up containing more detailed information. This popup view contains detailed calculations, findings and reasonings for the ground of said findings. During the time this popup window is active, the background will be “greyed out” to prevent distraction from data below the popup.

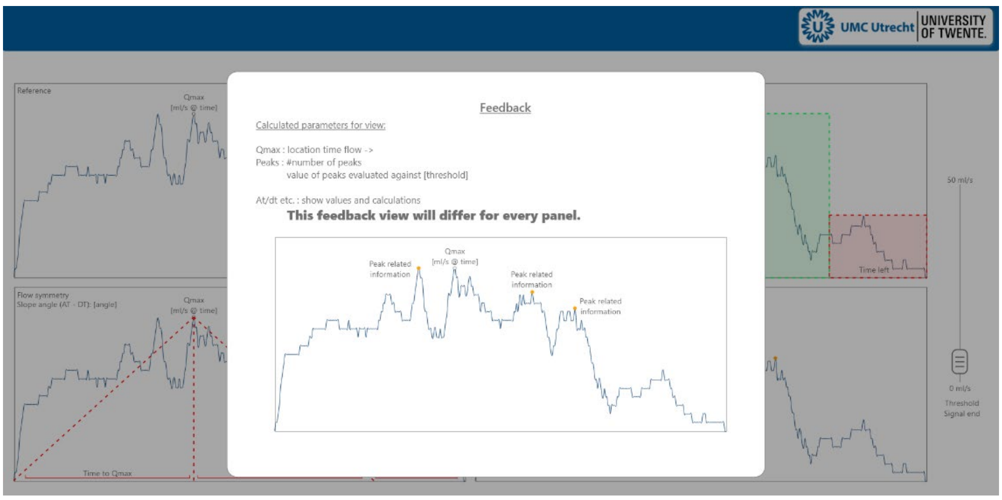


Figure 20 Detaild popup view of a graph in concept 2 (Adobe XD)

Using concept 1 as a reference, the following changes are made regarding the evaluation criteria as described in 3.3.2 Evaluation criteria for concepts.

- The number of clicks has drastically been decreased. Only one click is needed to go from the single graph basic view to the more detailed four graph view. One additional click is needed to access even more detailed information about one of the four views in particular
- Screen clutter is a bit of a problem here. At first, the screen clutter is very minimal. When the user is looking at the one graph at the home screen, there is almost nothing that can pose as a distraction from the huge graph presented. However, when the user reaches the more detailed screen, there are 4 graphs presented simultaneously. This can create quite a bit of strain on the user, as there is presented a lot of information at once. To combat this, the user needs to be very selective in what he or she wants to know and look at the corresponding graph. Screen clutter decreases again when the more detailed popup view is accessed, as the background (the 4 graph view) is greyed out.
- Interactivity stays more or less the same. The user can still navigate the graph in the main view, and has the option to access more detailed views in the 4 graph view.
- Control grouping has seized to exist, as there are almost no buttons left. Controls are completely based on intuitiveness of the user. Scrolling up causes the graph to zoom in, scrolling down to zoom out. Clicking presents more detail. It might be a good idea to introduce a small tutorial to make user aware of this, but these actions were chosen, as they are globally known.
- The amount feedback has also changed in multiple ways. Feedback is more selective based on the users actions, as it is presented in popup windows. To add to this, it has become more explanative, changing from values in the parameter windows, to descriptive feedback in popup windows.

Based on feedback from stakeholders on concept 2, a few more requirements arose.

- Descriptive feedback is highly preferred. **(MH/SH)**
- Add more on why the algorithm has made certain choices, based on values **(MH)**
- Limit slider to accommodate for the algorithm being subject to change **(SH)**

3.3.5 Small interim evaluation interview based on concept 1 and 2.

Between concepts 2 and 3, a small evaluation interview was held. During this interview, a multitude of things was discussed. All these things are more technical of nature.

- Should data be stored on the dashboard side?
- How do we get measurement data to the dashboard?
- What can we do to implement security?
- Should the dashboard be customizable?
- How will the dashboard be deployed?

Should data be stored on the dashboard side?

No, to prevent security risk, no data is to be stored on the dashboard side. This would make it a lot harder to get a green light for deployment.

How do we get measurement data to the dashboard?

Measurement data will need to be standardized into a specific format to import it into the dashboard. Preferred formats are Comma Delimited Values (CVS), or JSON. Both have its pros and cons. CVS files are easy to edit for people with limited knowledge of information technology, as they can be opened in Microsoft Excel and will appear in neat columns. JSON files are more object oriented. Every measurement point will be converted to an object, where the variables become properties of said object. The benefit of this is that D3 can easily import JSON files, and using JSON files grants us the ability to single out a specific measurement point more easily during evaluation.

For now, this standardization process is out of the scope of this research. A small MATLAB script has been created to get the sample data included with the classification algorithm into a csv format and exported. This script and its underlying repository can be found in Appendix D: MATLAB to csv conversion script. These datasets are used for testing and validating the system.

What can we do to implement security?

For now, security lies outside the scope of this research, but will be recommended as future work. To facilitate for security as of for now, no data will be stored on the server side of the interface.

Should the dashboard be customizable?

Maybe later on, for now what we are looking for is an "easy-to-understand" solution. The less people can do to confuse themselves, the better. This can be later on implemented if desired. Overall, the dashboard is supposed to be an aid for diagnosing, not a tool to play around with to explore features.

How will the dashboard be deployed?

The dashboard needs to be a stand-alone program, preferably reachable in a browser window, for easy compatibility across devices. Running the program in a browser also makes the path to further department-wide implementation easier if needed.

Added requirements

The following new requirements emerged from this interim evaluation interview.

- Storing of data is now allowed on the server side of the dashboard to prevent privacy and security issues **(NH)**
- Finding a way to standardize measurement data for easy importing into the dashboard **(WH)**
- Find a solution to increase security. **(WH)** Further security risk are also discussed in Appendix I: Ethical Evaluation report.
- The dashboard should not be customizable, at least not during the first stage of implementation and deployment. **(SH)**
- The dashboard should be deployed in a stand-alone fashion, preferable accessible by a browser window. **(MH)**

3.3.6 Concept 3

The third concept takes a way different approach. There is no home screen, and no buttons to be pressed. All the information is presented at once. The amount of graphs has been reduced to two, and more visual feedback is added. The limit slider is removed.

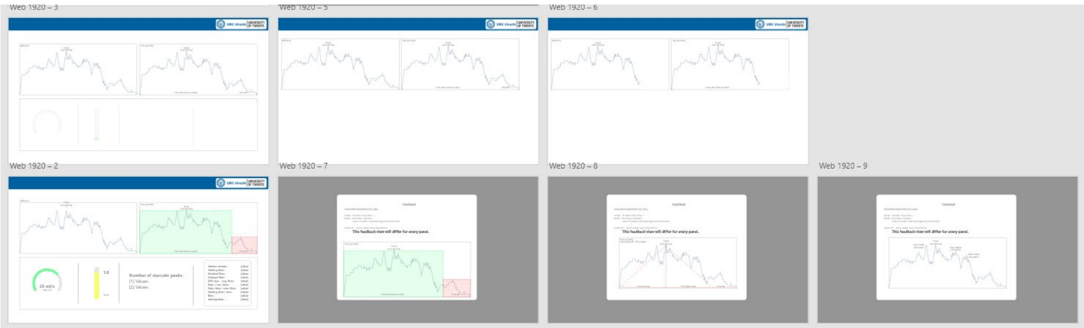


Figure 21 Concept 3 overview (Adobe XD)

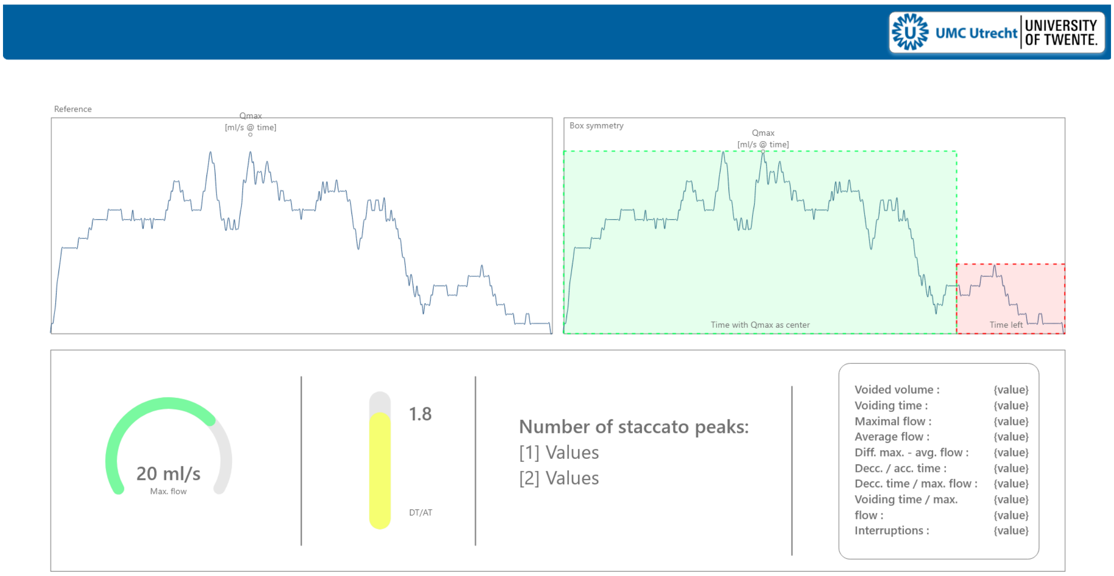


Figure 22 Concept 3 homescreen (Adobe XD)

A few new features have been introduced, there are the circle gauge, and the bar gauge. These gauges can be assigned to a lot of different variables, and are an easier way to spot if a variable is behaving strangely. Based on the deviation the variable has from a set norm, the bar will change color accordingly. Red means a large deviation, yellow a medium one, and green means a small deviation. It is not yet clear which variables are to be visualized using these gauges, but they have to be less data sensitive ones, which do not rely on the data imported. An example that was mentioned during a feedback sessions was the option to visualize the maximal flowrate, as opposed to the more regular flowrate we usually see in uroflow measurements. The problem with this visualization is that we need a constant which the visualization is based on, a reference value so to speak, and those are not yet very clearly defined.

Furthermore, feedback on peaks has become textualized and is immediately available at the home screen. The popup view has remained for more detailed analysis.

Using concept 2 as a reference, and again using the evaluation criteria as described in 3.3.2 Evaluation criteria for concepts., the following changes have been made.

- The number of clicks has decreased by one, due to removal of the old home screen containing the large graph overview. This has increased the ease of use of the application. Another factor that added to ease of use is the immediate availability of textualized feedback on staccato peaks. This feedback can be changed to any other variable in the future if the client desires.
- Clarity has increased, screen clutter has been decreased. The four graphs have been reduced to two, displaying only the most essential information. Parameters are available, but neatly put to the side.
- Amount of feedback has slightly increased, as there is textual feedback immediately available, and more detailed feedback is still present in the popup window.
- Interactivity has been decreased, the only way the user can interact with the dashboard at this current concept is by clicking on a graph to request more detailed information.

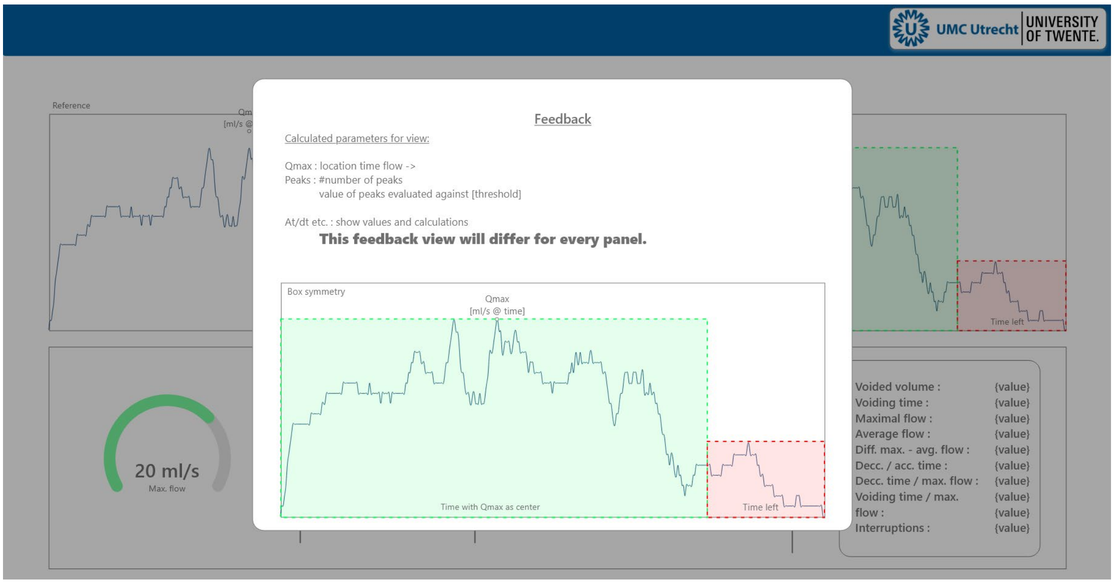


Figure 23 Concept 3's popup window, containing more detailed information (Adobe XD)

The response on feedback 3 was very positive. The client and stakeholders really thought the introduction of the circular gauge was a good idea, and presented variables they would love to see visualized in that way. During the feedback on concept 3, the design shifted to an even more textualized way of feedback, which can be seen in concept 4.

A major change that was requested was to move even further away from graphs as an aid to interpret data, using less visualizations based on the data itself.

Added requirements:

- Give more textual feedback based on the data and the classification algorithms choices. **(MH)**
- Move away from graphs and introduce ways of visualizing single variables. **(SH)**
- Add more detailed information, which should be available when requested **(MH)**

After having received feedback, I engaged in another interview with J. de Haan about eventual extra functionality that could be added. One of these functions is drawing a reference line over the actual data, as a representation of the rough shape of the graph. This rough representation can then be compared to reference shapes, which are used to identify more obvious symptoms. This feature is present in Concept 4.

Added requirement:

- Add a function that draws the rough path of the graph for easier comparison with reference data. **(WH)**

3.3.7 Concept 4

Concept 4 is a more finalized version based on the requirements discovered in earlier concept. For this reason, concept 4 is more sophisticated and will take longer to discuss. The purpose of this concept is to act like a guide for the final project. A few minor changes will be made later on.

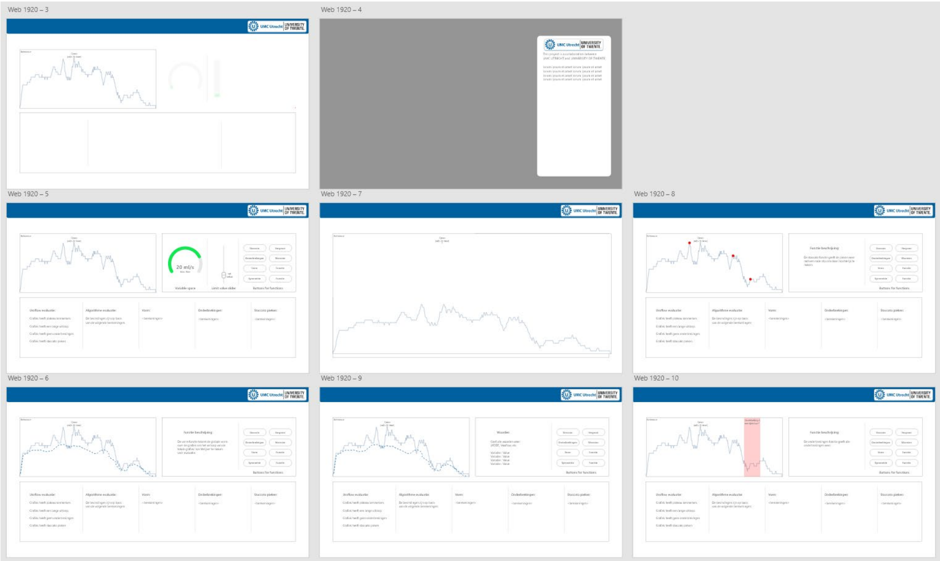


Figure 24 Concept 4 overview (Adobe XD)

The home screen of concept 4 (Figure 32) features a lot more interactivity, while maintaining a clean look overall. More functionality has been added, and a lot more textual feedback is present. Detailed feedback can still be accessed using the function buttons at the right of the screen. The amount of graphs has been reduced to one, which is used a reference for the classification algorithms feedback. The bar gauge has been removed, and the limit value slider has been reintroduced.

The textual feedback at the bottom of the screen is divided in five sections, namely Uroflow Evaluation, Algorithm Evaluation, Shape, Interruptions and Staccato peaks. In these sections the relevant feedback will be given corresponding to the name of each section.

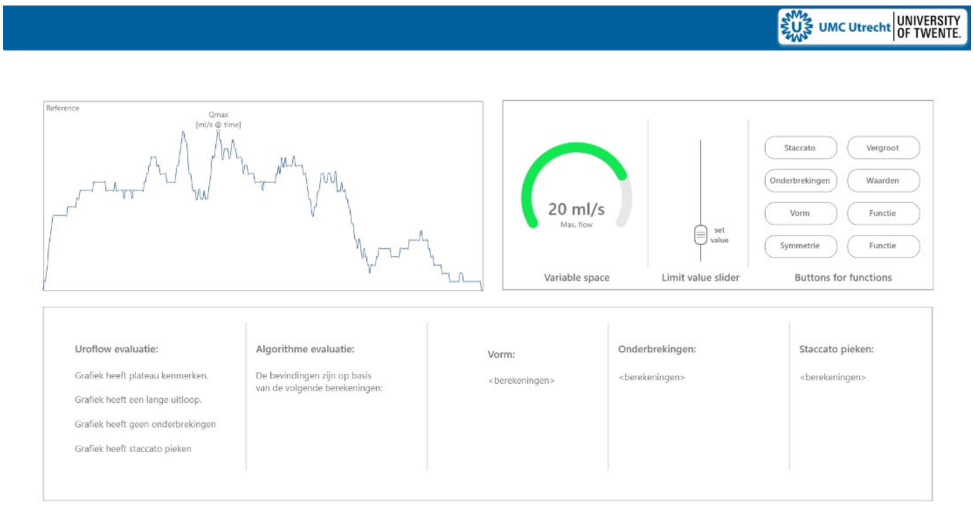


Figure 25 Concept 4 home screen (Adobe XD)

The first function we will take a closer look at is the staccato function. Pressing the corresponding button will update the graph in the main viewport, highlighting fluctuations that are of interest. Furthermore, the limit slider and circle gauge disappear, to make way for a more detailed explanation of which function was used, what the function does, and the results of said function. This is the same for all of the other functions, with exception of the enlarge function, which, as the name suggest, enlarges the reference graph to a full window view if the user so desires.

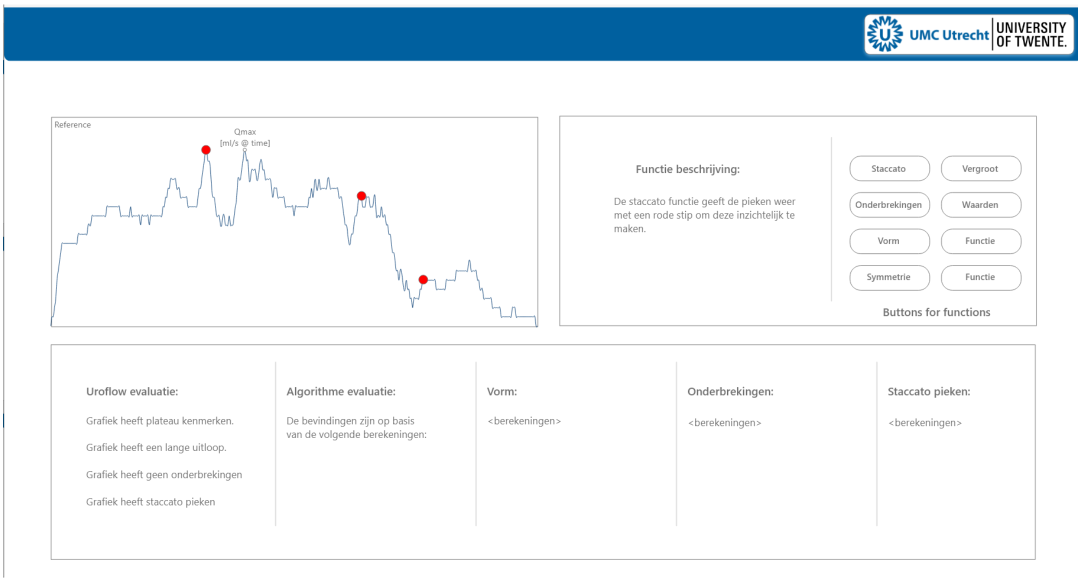


Figure 26 Concept 4 staccato function (Adobe XD)

The interrupt functions will highlight parts of the graph where interruptions where found, and will present the location and duration of a found interruption. Detailed information and explanation will again be available to the left of the buttons, this location will be further referred to as the dialog box. Please note that in Figure 34, the highlighted part is not actually an interruption, but was inserted for demonstration purposes only.

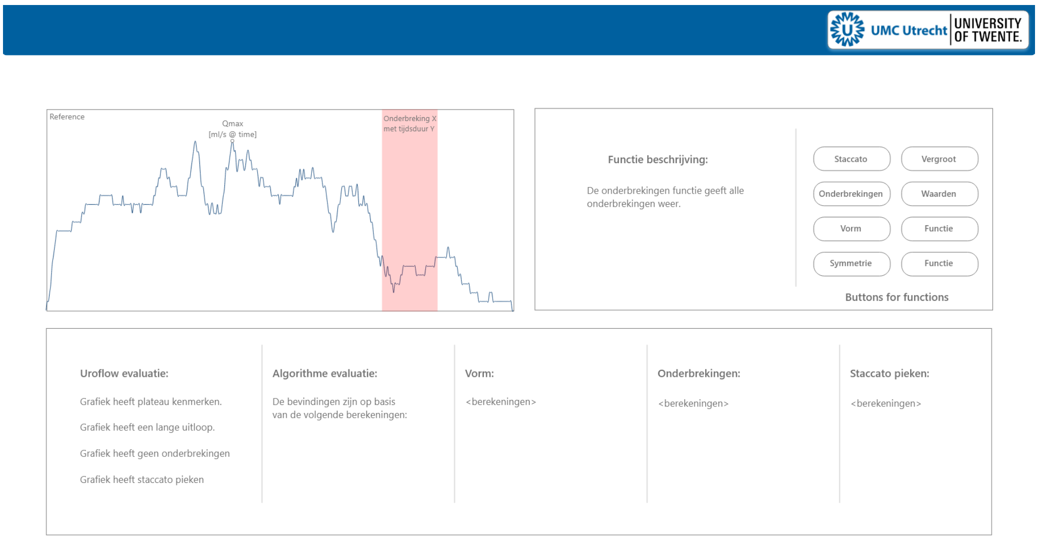


Figure 27 Concept 4 interruption function (Adobe XD)

The shape function, as introduced in concept 3, has made its way into concept 4. Pressing the shape button will draw a rough outline of the graph. The method that will be used for this is still to be determined. Calculations for creating this rough outline, and the method used will be presented in the dialog box.

Figure 28 Concept 4 shape function (Adobe XD)

The enlarge function will currently clear the graph of any extra information and will enlarge it to a fullscreen view. Clicking anywhere on the graph will cause the dashboard to revert back to the homescreen. A good alternative here is to keep any highlights caused by other functions to allow for a closer inspection of said highlights.

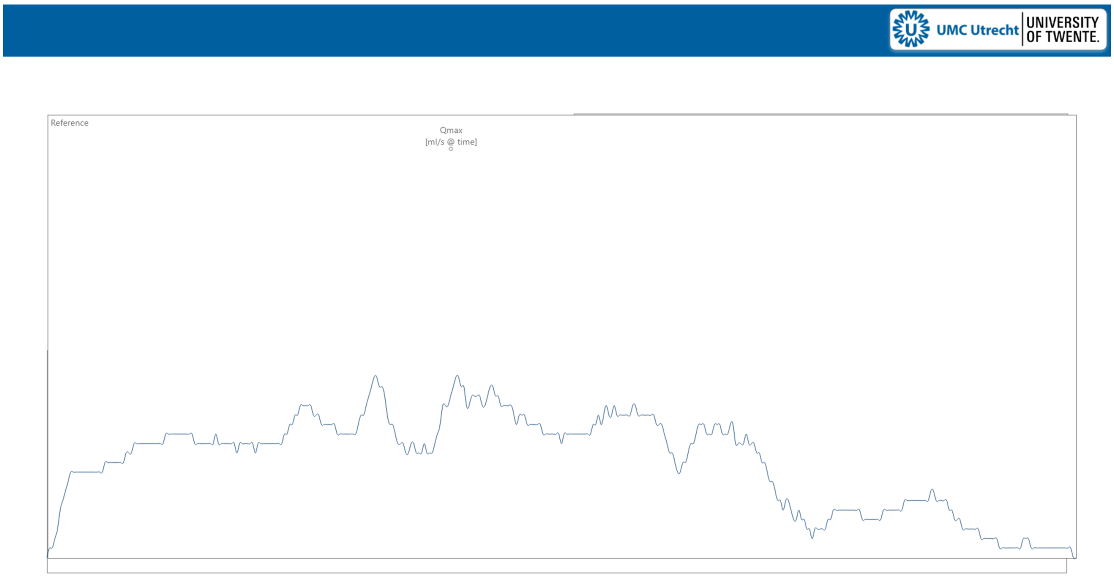


Figure 29 Concept 4 enlarge function (Adobe XD)

The parameter functions will present all variables in the dialog box, for further inspection. Addding an explanation to how all these parameters were calculated is unnessecary.

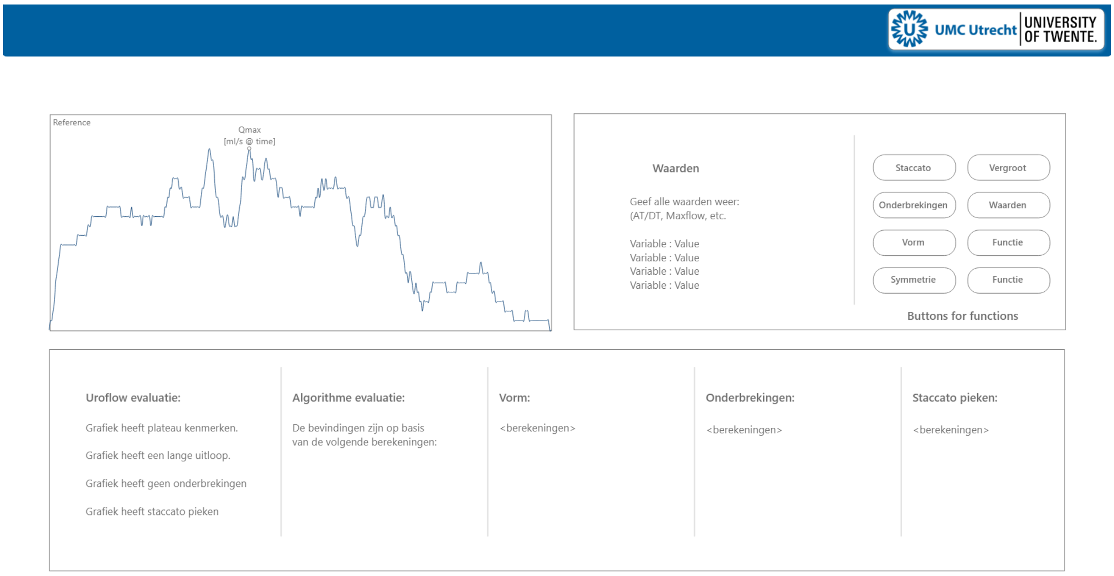


Figure 30 Concept 4 parameter function (Adobe XD)

As there is a lot more functionlity in concept 4, an interactivity overview has been included. In this interactivity overview, lines are drawn from buttons to their corresponding views. A full line represents a whole new screen, where a dotted indicated a pop-up view. Lines that have a dead end mean that if the user clicks at the same button again, the view will revert to the previous view. This image can be found in Appendix G: Interactivity overview Concept 4.

Using concept 3 as a reference, and again using the evaluation criteria as described in 3.3.2 Evaluation criteria for concepts., the following changes have been made.

- The number of clicks has been greatly increased, as detailed information is now only accessible by using functions.
- The overall ease of use of the application might have decreased slightly, due to adding a lot more functionality. The only way to counter this is to make buttons as self-descriptive as possible.
- Screen clutter has not increased, despite the fact that more information is present at the screen. All the information is neatly divided in a non-distractive manner.
- Amount of feedback has increased with a vast amount. A lot more feedback is now textualized. Popup windows are gone.
- Control grouping is reintroduced. All the buttons are neatly compacted into a grid. The slider is right next to this grid.

4 Specification

In this chapter, an outline will be given on all found requirements and how to achieve these requirements. After that, a number of data visualizations tools will be discussed to find the one that best fits our needs.

4.1 REVISED REQUIREMENTS

In this section, all the requirements will be compiled into a single overview. On top of this, features will be listed which should be present according to the requirements.

4.1.1 List of all revised requirements

- Clear and insightful with explanative feedback. **MH**
- Constant scaling of the axis on graphs that are used. **MH**
- Descriptive feedback is highly preferred. **(MH/SH)**
- Limit slider to accommodate for the algorithm being subject to change **(SH)**
- Storing of data is now allowed on the server side of the dashboard to prevent privacy and security issues **(NH)**
- Finding a way to standardize measurement data for easy importing into the dashboard **(WH)**
- Find a solution to increase security. **(WH)** Further security risk are also discussed in Appendix J: Ethical Evaluation report.
- The dashboard should not be customizable, at least not during the first stage of implementation and deployment. **(SH)**
- The dashboard should be deployed in a stand-alone fashion, preferable accessible by a browser window. **(MH)**
- Give more textual feedback based on the data and the classification algorithms choices. **(MH)**
- Move away from graphs and introduce ways of visualizing single variables. **(SH)**
- Add more detailed information, which should be available when requested **(MH)**
- Add a function that draws the rough path of the graph for easier comparison with reference data. **(WH)**

4.1.2 Requirement based features

Based on the requirements above, the dashboard should be able to satisfy the needs of our client by having the following features:

- The dashboard must have a clean and professional look, without having too many distracting items.
- The dashboard needs to provide a graph of the data using the same and constant scale for every dataset. If it is impossible to plot the dataset on this constant scale, the user should be notified, and the scale changed accordingly, without distortion of the x-to-y axis proportion.
- The dashboard needs to present textual feedback based on the classification algorithms findings
- The dashboard needs to have a stand-alone implementation and should be preferably accessible by a browser window.
- A limit slider must be present in the dashboard, to change the classification algorithm during execution.

5 Realization

5.1 DATA VISUALIZATION TOOL SELECTION

In this section, multiple data visualization tools that seem to be viable option for the course of this project will be discussed and compared. All tools will be ranked based on criteria that are critical to the project, such as, modularity, ability to import and change data at will, ease to use, and accessible through the internet. The ability to be able to run with JavaScript is a must, which was taken into consideration when selection the tools below.

5.1.1 KENDO UI [31][32]

Kendo UI is a set of JavaScript libraries that includes a large array of components ranging from data grids and charts to schedulers, dropdowns and even buttons. Kendo UI was originally developed by the Telerik company, which is now a part of Progress. Kendo UI is a commercial library and there are versions available that support the Angular, React, and Vue frameworks as well as basic jQuery environments.

The fact that Kendo is based on libraries makes it a very flexible and versatile tool, however, to my findings, documentation is limited and/or behind a paywall. The fact that Kendo has a very commercial mindset makes using it harder than it should be.

The main benefit from Kendo is, it does not only include visualization tools, but also app building and web development, which makes it easier to tie the visualization into the entire project as a whole. Despite of this, I am not convinced that Kendo is the right choice for this project, as becoming proficient at Kendo's language may take a lot of time due to the limited documentation.

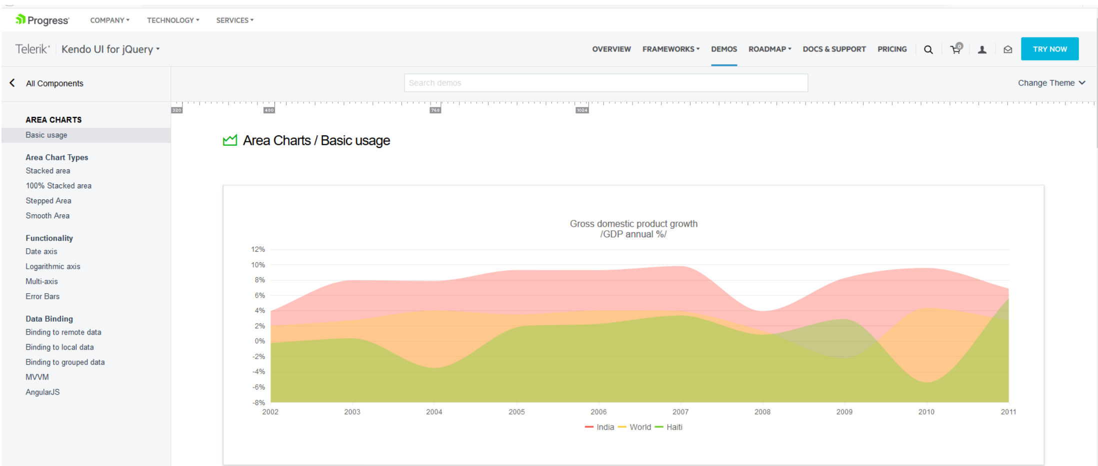


Figure 31 Kendo UI basic Area chart [33]

5.1.2 D3[32][34]

D3 stands for Data-Driven Documents and it is a JavaScript library for creating dynamic and interactive data visualizations. It was first released in 2011 and includes a very flexible and powerful set of features to help you build up various graphical data visualizations.
D3 is also easily incorporated into existing websites and/or frameworks, as it creates an SVG image it renders to the site. This has some major benefits.

The first benefit is that a framework can be build using any other tool the user feels comfortable and/or has experience with. Later on, the SVG image and accompanying code can easily be implemented later on.

The second benefit is resizing of the visualization. As D3 presents an image, as opposed to a combined section of elements, it can easily be fit into any space, and moreover, can be incorporated into bootstrap, which allows for dynamic resizing based on the screen size of the viewer.

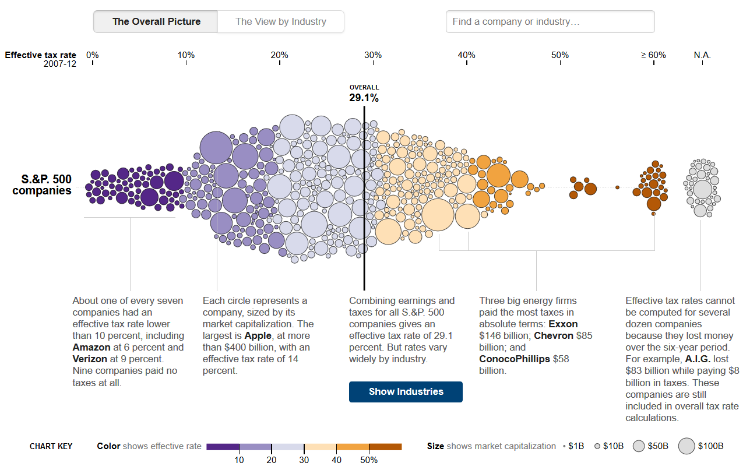
The third benefit of D3 is that it is not visible on the source code of the website. The only thing that links back to the visualization is a simple <svg> element tag. This is interesting to us, as the source code for the website is kept clean, as well as the code for the visualization itself. All JavaScript codes and html will be contained in different files. This makes the code for the overall project more readable and understandable.

D3 also offers a lot of flexibility to use different shapes and objects in graphs, and has a lot of ways to intertwine elements. This leads to very interesting takes on data visualization in general.

Overall, D3 is a very good fit for the project, as it is open source, and has a lot of documentation available. There is also a large active community on stackoverflow[35], who are eager to solve any problems anyone might encounter.

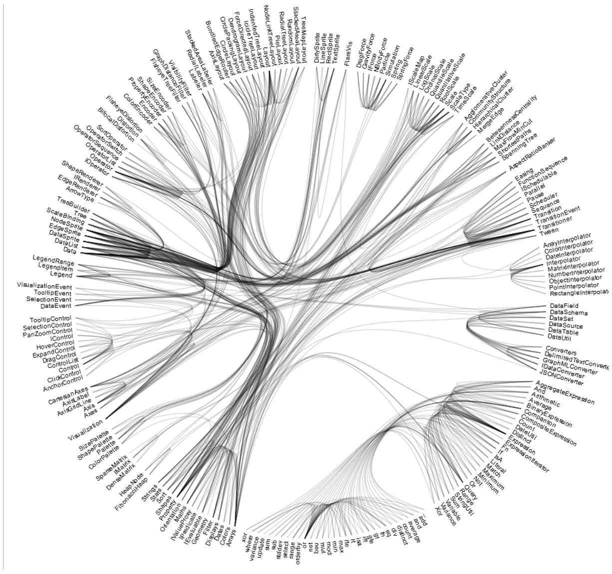
5.1.2.1 Examples of visualizations made in D3

Below, some examples of D3 visualizations can be found. This will show the versatility of D3 and some creative solutions to certain visualization problems.



> **"Across U.S. Companies, Tax Rates Vary Greatly**
>
> Last week, in a Congressional hearing, Apple got grilled for its low-tax strategy. But not every business can copy that approach. Here is a look at what S.&P. 500 companies paid in corporate income taxes — federal, state, local and foreign — from 2007 to 2012, according to S&P Capital IQ."

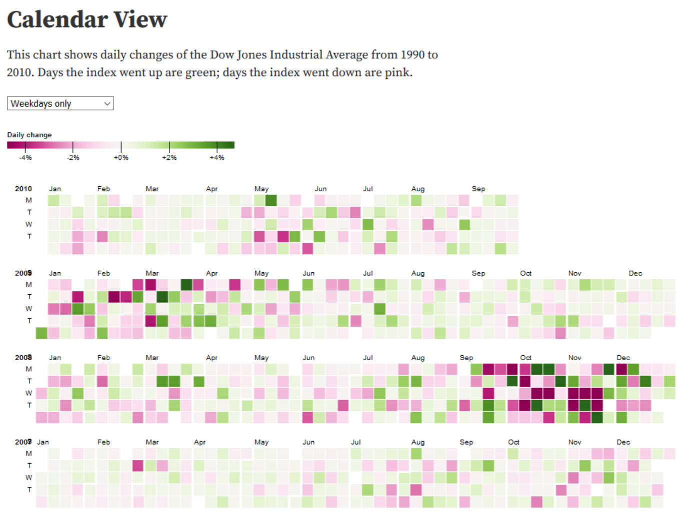
Figure 32 Example of D3 1: [36]



"**Hierarchical Edge Bundling**

This chart shows relationships among classes in a software hierarchy. Hover a class to reveal its imports (outgoing edges) and classes that import it (incoming edges)."

Figure 33 Example of D3 2:[37]



"**Calendar View**

This chart shows daily changes of the Dow Jones Industrial Average from 1990 to 2010. Days the index went up are green; days the index went down are pink."

Figure 34 Example of D3 3:[38]

5.1.3 React.GL[39]

React GL is a suite of tools created for large scale WebGL Data Visualization. The tools named under React GL are:

- Deck.gl
- Luma.gl
- React-map-gl
- Nebula.gl
- React-vis

A further look will be taken into react-vis, as that is the tool that suits our needs the most. React-vis is a more "down-to-earth" visualization tool, whereas the others mostly project data on maps, or provide ways to use WebGL to accelerate graphical representations, using the computer's graphics card to a higher potential in web displays. [40]

However, the implementation of React-Vis is a lot harder, as it requires its own engine, React.GL (based on node.js), to run. This complicates a lot of things for the framework the visualization will be housed in, as well as for the visualizations themselves. Also, implementing the classification algorithm will be a lot harder in this case. Nonetheless, React-Vis does support JavaScript, as its own engine is based of it, but the modularity of the entire project will take a big hit here, as files need to be in specific folders for them to work properly. This will greatly affect the ease of adaptation and further development of the project in a negative way.

React-vis does come with stunning graphics and snappy responses, also curtesy of its own engine. A few examples can be found below.

Figure 35 React-vis example[41]

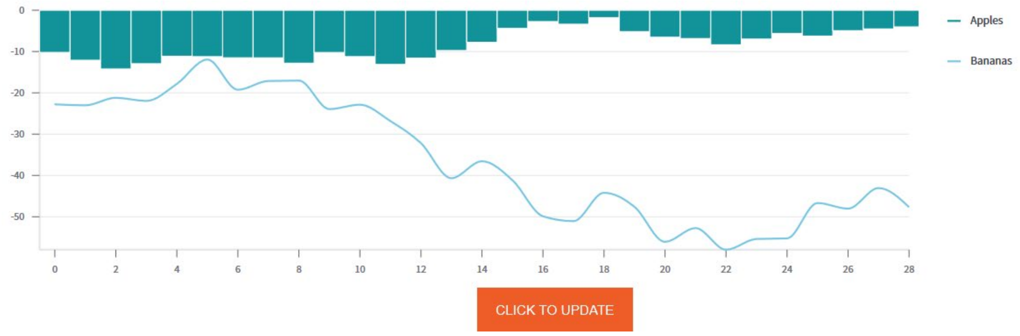


Figure 36 React-vis example after hitting update[41]

The examples above might look simple, but are the most similar to our use case, as it contains simple data points plotted on a single graph in different ways. It may seem underwhelming, but that is the core power of this kind of visualization.

5.1.4 Power BI[42]

Power BI is an interactive dashboard analysis tool developed by Microsoft. Elements can be created and linked to dataset to give great analytical insight in data. These elements can be uploaded to a website, to gain access

to these visualizations from anywhere in the world.

Figure 37 Power BI creation tool (adapted from [43]*)*

Power BI would be a great opportunity to build an excellent framework for this project, as it is very easy to use, and has lots of documentation and courses available. The ability to drag and drop elements in to place, and directly link data to them almost seems to easy. To add to this, the elements in Power BI use state of the art data visualizations, and hint to what kind of visualization should be used, based on the data format that is imported. Elements also mean that power BI is very modular, and could allow users to reshape their dashboard at will.

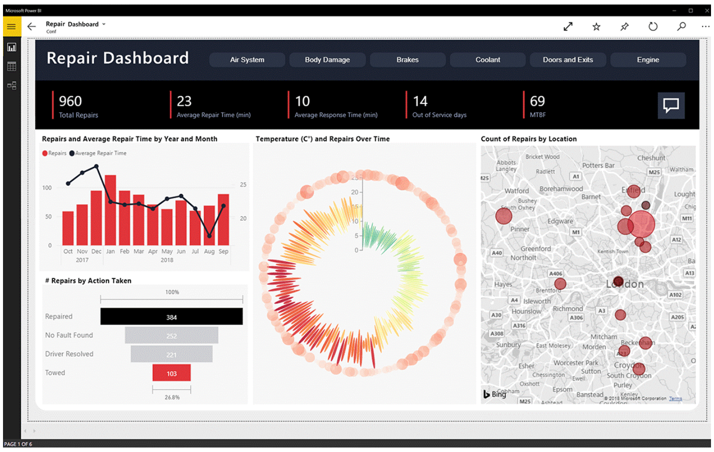


Figure 38 Examples of visualizations in Power BI [44]

There is a catch however, Power BI is a paid (subscription base) product, which would require our client to purchase a license.

Furthermore, the algorithm cannot easily be implemented, as this requires live data manipulation. This could be solved by installing a plugin that allows us to execute JavaScript inside the dashboard, but this will most likely result in a sloppy work around.

Power Bi does however, present great examples of how a professional dashboard should look like, and how interaction with the dashboard should be implemented.

5.2 TOOL SELECTION CONCLUSION

Tool	*Ease of use*	*JavaScript*	*Modularity*	*Data manipulation*
Kendo UI	+/-	+	+	+
D3	+	++	++	++
React-Vis	-	+	+	-/+
Power BI	++	-	++	-

Table 2: Rating of different visualization tools

Based on the reviews of all of the following tools above, the following choice was made for further developing the project.

D3

The main motivation for choosing this tools, is its modularity and responsiveness to a webpage, as it creates its own stand-alone visualizations without cluttering source code.

Furthermore, the fact that D3 is entirely based on java, makes implementing the algorithm a lot easier, as variables can easily be passed between scripts. This allows us to keep the source code for the classification algorithm clean and insightful, making further development easier.

In this chapter, the realization of the project will be discussed. Used tools for both the framework and the visualization will be discussed. A list of all the programs used for this project can be found in Appendix C: Online Resources.

5.3 SETTING UP THE TOOLS

To make sure the dashboard can be implemented in a standalone fashion and be accessible in a browser window, we need to set up a localhost server. This is required as PHP is used, which can only be executed on the server-side of a web application.

USBWebserver

The tool used for creating this localhost server is USBWebserver [45]. USBWebserver is an easy to use the program, hosting a local accessible Apache server with a MySQL database connected to it. As discussed before, we won't use the latter, as we will not be storing data on the server-side, but it comes with USBWebserver as a package.

When launching USBWebserver, the following interface is presented (Figure 39). The Apache server and MySQL database will start automatically. The web server can be accessed in a browser by using the localhost at port 8080 as an address. This can be done in a multitude of ways, but the two easiest will be listed.

- Using the address: localhost:8080
- Using the address: 127.0.0.1:8080

Figure 39 The USBWebserver interface

The USBWebserver works of a filesystem following a quite simple principle. When the program is first accessed there will be a few extra folders present to where the program was extracted to. These folders all contain files essentials to the program, but don't need to be touched as the webserver requires no further configuration. The entirety of this program is a plug and play solution.

To edit and upload files to the web server, the user has to access the root folder. This folder can be found in the same folder as the program or can be accessed via the webserver itself by pressing "Root dir", which stands for root directory.

Installing D3

In this root directory, a default index.html file can be found which can be discarded. The next step is to install D3. This is also a fairly easy process, as D3 is in essence just a collection of files. To install D3, download it from the website, and extract the .zip file. Copy all of its contents to the root folder. To initialize D3, a simple webpage containing an svg element is required. To check if your installation was successful, download an example script and also place it in the root folder. The source code needed to initialize both D3 and bootstrap can be found in Appendix E: Setting up the basics.

Installing Bootstrap

Bootstrap was also used during the development of this project. The main reason for this, is that bootstrap allows you to responsively resize elements as the window size changes. This ensures us of a correct display of the dashboard on any size monitor. To install bootstrap, a set of links containing paths to stylesheets and scripts can be copied from bootstrap download page. The source code to initialize both D3 and bootstrap can be found in Appendix E: Setting up the basics. Using the links from the bootstrap website, instead of downloading the package, allows our server to grab bootstrap live from their servers, minimizing program size, and keeping the number of files to a minimum.

5.4 CREATING THE DASHBOARD

The workload of creating the dashboard will be divided into 2 sections. One the framework of the entire dashboard, the other the data visualization and incorporation of the algorithm.

5.4.1 Creating the framework

To develop the framework, which is all coded by hand, a special text editor was used. This text editor makes editing code easier, as it color codes tags, and automatically indents code to give a clearer overview. It also provides some autofill features, to create a more enjoyable workflow. A code text editor is highly recommended for projects such as the development of this dashboard.

Creating the framework for the dashboard was not a hard task thanks to the implementation of bootstrap. Bootstrap allows you to divide the page into containers, which will be subdivided into rows containing columns.

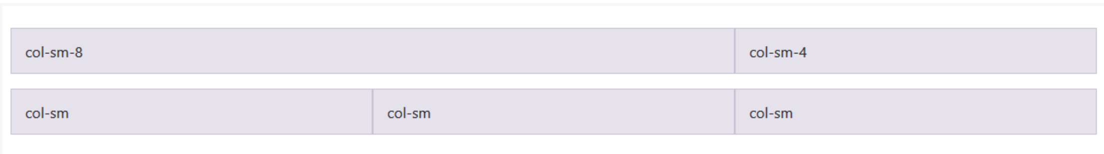


Figure 40 Bootstrap grid system [47]

Bootstrap will divide the page into a grid 12 columns, giving the user a wide range of options to work with. If a column width is not specified, bootstrap will automatically resize all the columns to the same size where they will fit the screen. All columns width are not defined in a definitive measure such as pixels but can be compared to percentages. A column width of 8 will result in a column spanning 75% of the available browser window width, as 12/8 is 75%. Columns can also be nested into other columns to gain even more control.

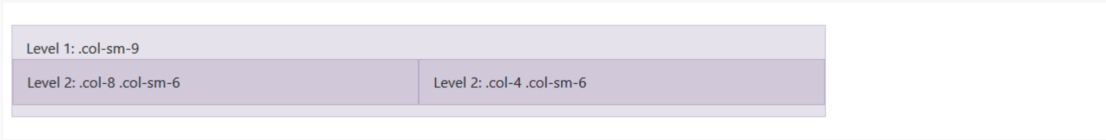


Figure 41 An example of nested columns [47]

Bootstrap also adds lots of other useful features, such as self-aligning items and content-aware elements. Another useful thing about bootstrap is that it comes with a style sheet, allowing us to use globally known button designs, instead of having to create our own.

The detailed source code of the entire framework can be found in Appendix F: HTML source code for the bootstrap grid. Below, the layout can be found in a grid view.

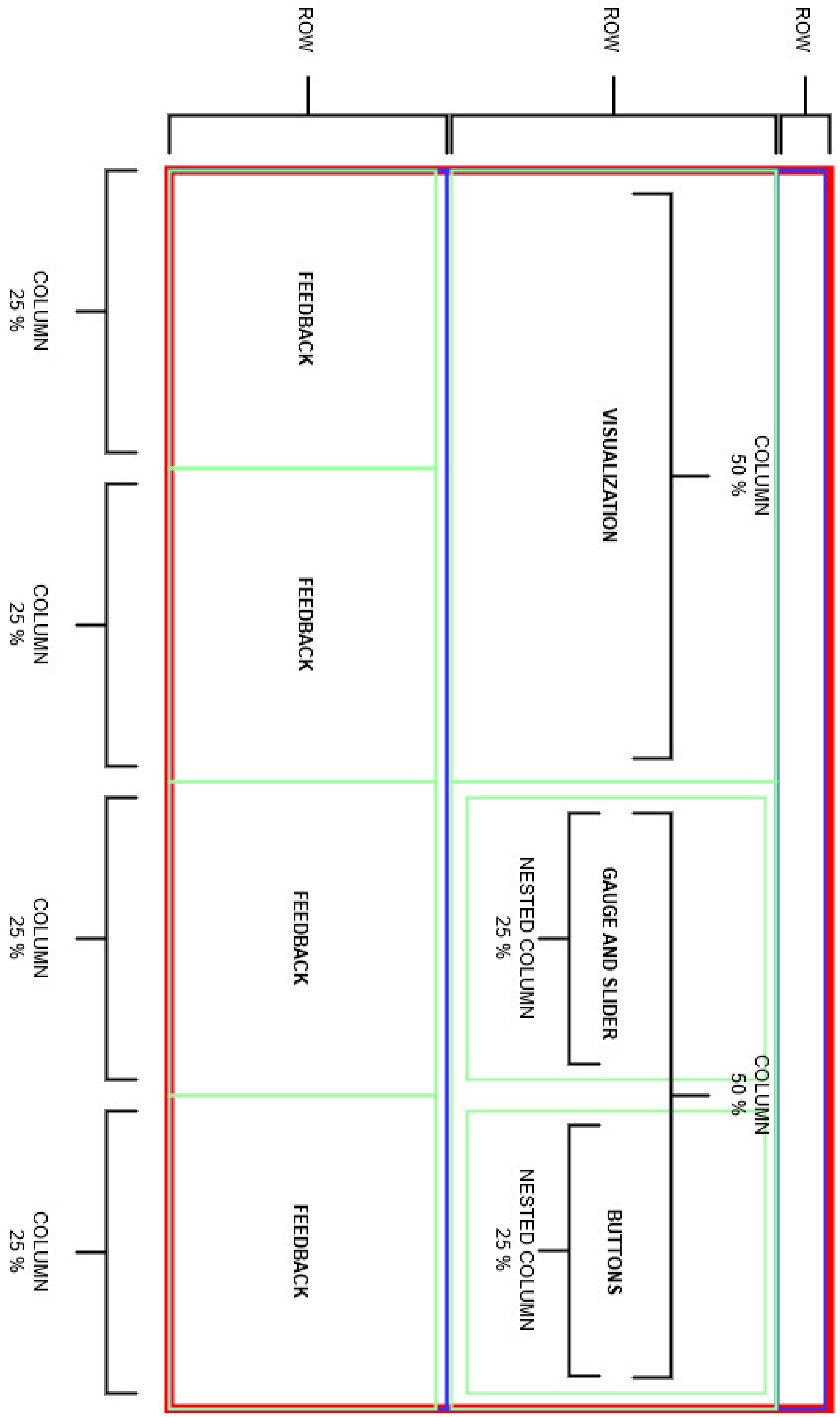


Figure 42: Grid view of the dasbhboard, as to be implemented in bootstrap. (created at draw.io)

5.5 CREATING THE DATA VISUALIZATION AND INCORPORATING THE ALGORITHM

In this section, we will discuss how the data visualization was created, and how the classification algorithm is incorporated in the dashboard.

5.5.1 Data visualization and sample data.

The data visualization itself will use sample data that was provided with J. de Haans classification algorithm. This data will be used to validate the correct execution of the algorithm. To simulate future use of the dashboard, sample data will not be available directly in the dashboard but must be uploaded. The sample data is contained in a MATLAB variable. To convert these variables into a usable file format for D3 to use, a conversion script was created. This script can be found in Appendix D: MATLAB to csv conversion script.

The output of this script is a comma-delimited file, using 2 columns. The columns have headers called time and value, corresponding to their variables.

	A	B
1	time	value
2	0.125	0
3	0.25	0
4	0.375	0.7
5	0.5	1.9
6	0.625	2.7
7	0.75	3.8
8	0.875	4.8
9	1	5
10	1.125	5.6
11	1.25	6.5
12	1.375	7.2
13	1.5	8.3
14	1.625	9
15	1.75	9.5
16	1.875	10.2
17	2	10.8
18	2.125	11.1
19	2.25	11.3
20	2.375	12.1
21	2.5	12.8
22	2.625	13.1
23	2.75	13.5

Figure 43: An example of what the converted data looks like when opened in Microsoft Excel

Another benefit of using CSV files is that they are easy to understand. These files can also easily be validated, as peculiar measurement values can be easily spotted. These files can also easily be altered to validate the robustness of the dashboard.

D3 will separate these columns into split variables. It achieves this by placing every value of each column in an array, with matching indexes, which allows D3 to match corresponding values when plotting them.

5.5.2 Incorporating the algorithm in a modular way.

To keep the source code for the framework and all other source code separate is an easy task. This is due to the fact that the framework is built in HMTL and the visualization and algorithm are written in JavaScript. However, as the algorithm needs to manipulate and evaluate data that is contained in the D3 source code, they need to be programmed in the same file. This is not in accordance with our requirement to implement the algorithm in a modular way. One workaround for this is to add ES6-modules. ES6-modules allow you to import scripts in a modular way, instead of having all code in one file. On top of this, it means we can load certain scripts only when they are required, reducing the load time of the dashboard. This means we can create different files for every function of the algorithm, and a different file for the visualizations as well.

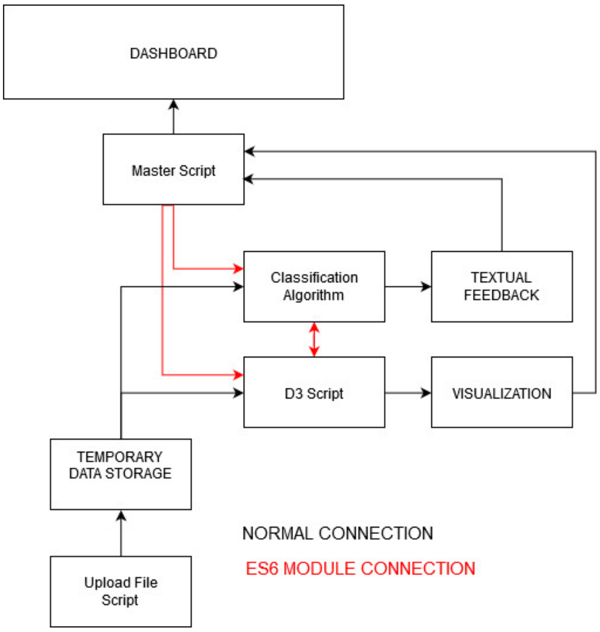


Figure 44 Code tree (simplified)

Using ES6 we can build a system of scripts from the ground up, keeping code for the visualization and classification algorithm separate. In Figure 44 a simplified version of this proposed code tree can be found. Red arrows indicate an ES6 module connection, allowing for interchangeable function use. Black lines are normal import/export connections of final values.

The dashboard block represents the dashboard, the part the user will actually see. The master script is the script that will forward all data/objects to the dashboard, where they will be represented.

The Classification Algorithm block represents all the files and function of the algorithm combined. Data will be fed into this block, starting from the Temporary Data Storage block. This data will be evaluated, and if necessary manipulated by the algorithm. The parameters found by the classification algorithm will be converted to textual feedback and send to the dashboard. If data was manipulated, it will be send to the D3 script to be visualized. Further data, such as peaks and interruption intervals will also be send to the D3 script to be visualized. As the algorithm works in a very specific order, the data between different functions will be also be shared within the Classification Algorithm block. A more detailed overview of the workings of the classification algorithm can be found in 5.6 Creating the classification algorithm.

All of the source code can be found in appendix H: Source Code for the Dashboard

5.6 CREATING THE CLASSIFICATION ALGORITHM.

In this section, the different part of the classification algorithm will be discussed, based on their functionality and how they were created in JavaScript. Please keep in mind that the algorithm operates in a very specific order, as a lot of functions use the results of previous functions. During the development of this project, the algorithm also was suspect to some changes, as more functionality has been added. A serious effort was made to incorporate these extra functions into the dashboard, but unfortunately, some of them might have to be referred to as future work. Furthermore, staccato peaks have been renamed to fluctuations.

In addition to all the information below, the source code of the algorithm can be found in appendix H: Source code for the algorithm.

5.6.1 Calculating Parameters.

The first and most basic function is to find all the basic parameters amongst the dataset. These parameters will be further used throughout the algorithm to calculate more complicated matters such as interruption intervals and fluctuations. These parameters can be found by doing low-level data manipulation and basic calculations.

The parameters that are calculated during this stage of the classification algorithm are:

- Timestep, used for calculating integrals. The timestep is found by taking the average time between measurements.
- Timing and values. These two parameters are created by manipulating the data imported into the classification algorithm. First, the data, which is in CSV format, is split into 2 rows, called Timing and Values, each containing their respective measurements. After this, the variables are cleaned. From the back of the list we start to remove data from the value list if the value of the measurement is either 0 or *null*, meaning there is no data available, or the measurements have produced no useful data. When we encounter a value that is either not *null,* or not a 0, we stop this process. After this, the timing list is cut to the same length as the updated value list.
- Ttot, abbreviation of TimeTotal, which stands for amount of time in seconds the measurement took. This variable is found by fetching the last measurement in the timing list and subtracting the first measurement.
- Vtot, abbreviation of VolumeTotal, which stands for the total amount of voided volume. This variable is found by multiplying every value during the measurements with timestep and adding them all together, creating a crude but sufficiently accurate integral.
- Qmax, the maximal flow in ml/s during the measurement, which is found by finding the maximum value in the values list. Furthermore, the corresponding time is added, creating a measurement object using the following structure: Qmax[0:value in ml/s, 1:time Qmax occurred at in s].
- Qmean, the average flow in ml/s. Found by adding all values together and dividing by the total amount of measurements.
- Qr, the ratio between Qmean and Qmax.
- AT, the time it takes to reach Qmax.
- DT, the time from Qmax to the end of the signal.
- DTAT, the ratio between DT and AT.

- TQ, the ratio between the total time and Qmax.
- Qmm, the ratio between Qmean and Qmax.

Note that the timing and value variable lists are created and manipulated in such a way that they have corresponding indices. This allows us to link a value to a measurement time using its index in the list.

5.6.2 Finding interruptions.

To find interruptions in the data, we must first define the criteria that an interruption must meet to be considered an actual interruption. This criteria is:

- The flow at a certain point is lower than the flow threshold.
  (At time of writing: 0.5ml/s, this can be changed in the threshold configuration file).
- The total voided volume after an interruption must exceed the volume threshold.
  (At time of writing: 5ml, this can be changed in the threshold configuration file).
  If this condition is not met between interruptions, 2 or more interruptions will be merged into 1.

Now that the criteria is defined, we can start looking at the method on how interruptions are found.

First of all, we find all values in the value list that are lower then the flow threshold, as all values in this list are in ml/s. The corresponding indexes of these values will be stored in a separate array variable, let's call this variable lowValues for now. Secondly, we convert the data stored in the lowValues array to intervals. An example of this is:

lowValues array:	[4,5,6,7,8,9,12,13,14,56,57,58,59]
Converted to intervals:	[4,9,12,14,56,59]

We achieve this by copying the first entry, and consecutively incriminatingly sifting through the array, checking if the next value is one higher then the previous one. If this is the case, we skip this value. If this is not the case, we copy the current and next value to a new array, called Intervals. Lastly we add the last value, and are left with an interval array based on the lowValues array.

Secondly, we apply the volume threshold. For this, we take the Intervals array and check in between the intervals. To explain this further, we take the example from above and elaborate.

Intervals array:	[4,9,12,14,56,59]
To check for volume:	0-4; 9-12; 14-56; 59-end;

In between these intervals, we take again a crude integral, as described in 6.3.3.1 Calculating Parameters (Vtot). We add all values contained in "to check for volume" per interval, and multiply them with the timestep for the current data. After this, we add all values, again per interval together. If the result does not exceed the volume threshold, it is deleted form the list, or merged together with the next interval, depending on if it is at the start or end of the list or not. If there is still a portion of the graph is still left after the last found interruption, and it does not exceed the volume threshold, it is deleted from the dataset entirely, as in this case, we can consider the start of the last interval as the end of the signal. This could leave us with the following, based on our ongoing example.

Final Intervals Array: [4, 14, 56, 59]

In this case, the volume in front of index 4 was larger than the threshold, and the volume after index 59 as well. The volume voided in between index 9 and 12 however, was not, so the interruption intervals 4-9 and 12-14 have been merged to 4-14.

After all of the above has been applied to the data, it is converted to time. Note that we executed this data manipulation on indexes, and not on values themselves. This allows us to easily convert the interval array into points in time. For this, we take the Final Intervals array and convert it back to a continues array, leaving us with:

Continues Final Intervals Array: [4,5,6,7,8,9,10,11,12,13,14,56,57,58,59]

Using this continues intervals array, we look up the corresponding timestamps in the timing list. Once the data has been converted to points in time, we forward it to D3 to be visualized. Additionally, we send textual feedback which has been created. This is done by dividing the amount of datapoints in the Final Intervals Array by 2, which gives us our amount of intervals, and again finding the corresponding timestamp to the data in the Final Intervals Array. We combine all this gather data into a message using the following format:

N intervals found:
1: from #s to #s
2: from #s to #s;
…..
N: from #s to #s

This message will be displayed in the dialog box, referred to in 3.3.7 Concept 4.

5.6.3 Calculating fluctuations. (Staccato Peaks)

Fluctuations are quite tricky to find, as they are defined by the peaks surrounding them. Fluctuations have also a quite strict, and honestly confusing at first criterium they need to meet, before being classified as an actual fluctuation. This criteria is called prominence. The prominence of a peak needs to be more then 20% of Qmax for a peak to classify as a fluctuation.

Prominence is calculated by finding the value of the highest valley between the peak that is evaluated and the 2 closest peak that are higher then the current peak or signal end. The latter however, is negligible, as we will always find a higher peak, referred to as Qmax. Qmax itself does not classify as a fluctuation. The way MATLAB calculates prominence is explained as follows (MATLAB documentation)[48]

To measure the prominence of a peak:

1. Place a marker on the peak.
2. Extend a horizontal line from the peak to the left and right until the line does one of the following:
    a. Crosses the signal because there is a higher peak
    b. Reaches the left or right end of the signal
3. Find the minimum of the signal in each of the two intervals defined in Step 2. This point is either a valley or one of the signal endpoints.
4. The higher of the two interval minima specifies the reference level. The height of the peak above this level is its prominence.

Point 2b can be neglected, as we will always find a peak that lies higher then the end of the signal, due to the nature of uroflow graphs. Detailed examples can be found in Figure 44: Prominence example, and its accompanying table 3: Prominence example.

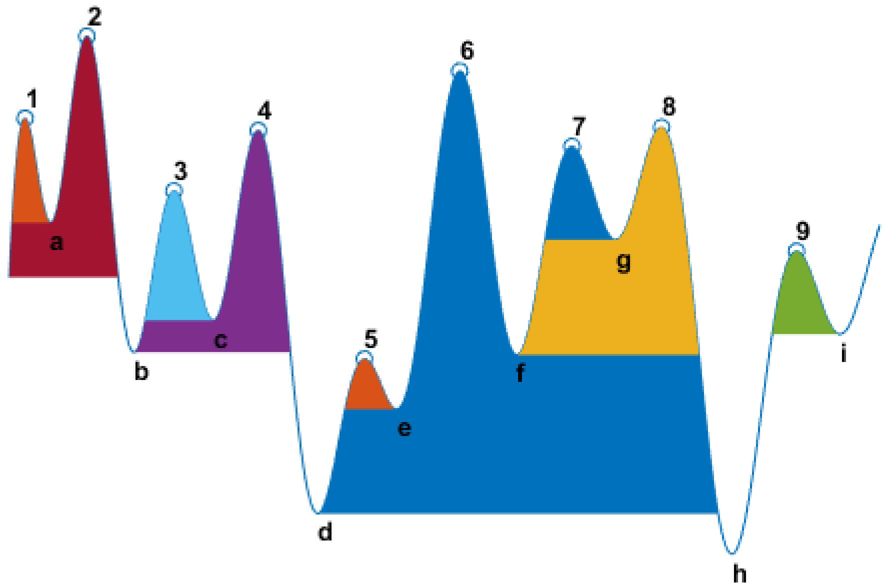


Figure 45: Prominence example[48]

Peak Number	Left Interval Lies Between Peak and	Right Interval Lies Between Peak and	Lowest Point on the Left Interval	Lowest Point on the Right Interval	Reference Level (Highest Minimum)
1	Left end	Crossing due to peak 2	Left endpoint	a	a
2	Left end	Right end	Left endpoint	h	Left endpoint
3	Crossing due to peak 2	Crossing due to peak 4	b	c	c
4	Crossing due to peak 2	Crossing due to peak 6	b	d	b
5	Crossing due to peak 4	Crossing due to peak 6	d	e	e
6	Crossing due to peak 2	Right end	d	h	d
7	Crossing due to peak 6	Crossing due to peak 8	f	g	g
8	Crossing due to peak 6	Right end	f	h	f
9	Crossing due to peak 8	Crossing due to right endpoint	h	i	i

Table 3: Prominence example[48]

Now that the strict definitions have been explained, the method of calculating these definitions can be discussed.

There are 3 main parts to calculating these fluctuations, namely finding all local maximums, all local minimums, and finding higher peaks than the current one we are evaluating. First we will take a look at finding all the peaks. To achieve this, we filter trough the value array, looking for entries where the next value is lower then the current value, and the previous value is equal or lesser then the current value. One thing that should be noted, is that we can only check from index 1 (the second entry) to the previous to last one, as we can not find a previous value for the first entry, and no next value for the last entry. Due to the nature of uroflow graphs, this does not pose a problem, as the first and last measurement will never contain a peak. All the corresponding indices to these peaks will be saved in an array called PeakArray.

The same thing will be done for finding all local minima, or valleys. However, instead of looking for smaller values, we now check the previous and next entries for higher values. All the corresponding indices will be stored in a ValleyArray. We can confirm correct execution of this part of the algorithm by checking if all valley indices are in between peak indices.

The next part is more complicated. In this part the actual prominence will be calculated. To do this, we iterate trough the array, fixing a different entry every time. This fixed entry will be stored in a variable called currentpeak. Next, we temporary split the array at the index of the currentpeak entry, creating 2 new arrays, one with all values left, and one with all values right of the current peak. This is necessary, as for all values that are left of the current peak entry, we want to loop backwards, finding the closest higher peak to the currentpeak, and for all values to the right of currentpeak, we want to loop forward to find the closest higher peak.

The selection criteria for higher peaks is as follows. If we encounter a value, whose entry corresponds to a higher value than the entry of current peak, we stop looking for new peaks, and save this found entry as leftPeak and rightPeak respectively. An example can be found below. For ease of understanding, the values have been included in the table, in the code these can be found by fetching the entry with index "Entry (index)" from the values list.

Index	0	1	2	3	4	5	6	7	8	9
Entry (index)	24	28	35	41	52	60	66	72	80	93
Value	10	9	7	8	11	14	13	10	8	9

Table 4: Example array containing peak locations and values

Let's say index 3 is currently evaluated, this peak has a value of 8, the array is then split into a left part, and a right part.

Index	0	1	2
Entry (index)	24	28	35
Value	10	9	7

Table 5: Left part of the peak Array

Index	0	1	2	3	4	5
Entry (index)	52	60	66	72	80	93
Value	11	14	13	10	8	9

Table 6: Right part of the peak Array

The left part will be evaluated backwards, and this will stop when a higher value is found, meaning leftPeak will result in Entry (index) 28, as 9 is a higher value then 8. The rightPeak variable will be set to Entry (index) 52 with a value of 11.

Now, we have the following information:

Variable	Left Peak	Current Peak	Right Peak
Entry (index)	28	41	52

Table 7: Information gathered on the current peak

Using this information, we find the lowest values, which are already saved into the ValleyArray between the Entry (index) of the leftPeak and currentPeak, and between currentPeak and rightPeak. After we have found these minimal values, we take the largest of these two, is will be our prominence reference. The actual prominence is calculated by subtracting the prominence reference from the current peaks value. If this result is higher than 20% of Qmax, the peak classifies as a fluctuation.

This process will be conducted for every peak. Every peak that does classify as a fluctuation will be saved in a separate variable, and converted to time. Here, the specific order of the algorithm comes into play, as we only will display fluctuations on the largest interval between interruptions. This means we need to fetch the Final Intervals Array, find the biggest interval, and only show fluctuations that are contained in said interval.

After all of the above has been executed, the fluctuation data is then converted to peaks, and forwarded to D3 and the dialog box, to be visualized and the findings explained.

5.6.4 Shape description

For the shape description, different parameters are evaluated to conclude the basic shape of the graph. The parameters that are evaluated in the current state of the project are:

- Interruptions, are interruptions present?
- Fluctuations, are fluctuations present?
- The total volume voided
- Qmax
- The ratio of time before to time after Qmax (DTAT)
- The ratio of average flow to Qmax (Qr)
- The ratio of signal time to Qmax (TQ)

Other parameters are calculated based on the parameters above, these parameters are: **Qref** and **dev**, where **Qref** is a reference value containing a deviation of **dev**.

Currently there are 4 sets of conditions the shape description module can apply to a graph. These shape descriptions are: **intermittent, fluctuating, bell-shaped,** and **high/low maximum**. Further shape descriptions that are to be applied as future work are **plateau** and **symmetry.**

The first thing the shape module does is look for interruptions and fluctuations. If these are present, the terms **intermittent** and **fluctuating** are respectively appended to the shape description. The other descriptors are trickier, as they require calculations that have not been done in any earlier stage of the algorithm.

For a curve to classify as **bell-shaped**, 2 of the following 3 criteria need to be met:

- Qr is larger than 0.63
- TQ is smaller than 1.28
- DTAT is in between 0.85 and 2.33

When 2 of these 3 criteria are met, the term **bell-shaped** is appended to the shape descriptor.

To determine if a curve has a **high or low maximum**, Qref and dev have to be calculated. These variables have different calculations based on the gender of the patient, but as in the first version of the classification algorithm, only female patient data was provided, it was opted to go with the calculation for female data. Implementation of the ability to differentiate between genders and calculate Qref and dev accordingly will be further referred to in future work. For now, Qref and dev are based on the following calculations:

$$Qref(female) = 6 + e^{0.511+0.505*\log(Vtot)}$$

$$dev(female) = 4 + (\frac{6}{500} * (Vtot - 100))$$

Vtot is the total voided volume, which if it exceeds 600, will be set to 600.

If the maximal flow in ml/s (Qmax) exceeds Qref+dev, the curve will be classified as having a high maximum. If Qmax is lower than Qref-dev, the curve will be classified as having a low maximum. If the value of Qmax is in between Qref-dev and Qref+dev, the curve has no deviation in Qmax worth mentioning.

5.6.5 Thresholds

The thresholds module is a simple module containing all the thresholds and limit values. The reason for the existence of the threshold module is to give the user the ability to easily change thresholds and limit values when desired, without looking through all modules to change the respective thresholds for the calculations they execute.

The thresholds that are contained in the thresholds file are:

- Flow threshold : set at 0.5 ml/s
- Volume threshold : set at 5 ml
- Peak threshold : set at 20% of Qmax

5.6.6 Classification engine

All of the modules described in 6.3.3 Creating the classification algorithm. come together in a main file called classificationEngine. This file is responsible for importing the modules and executing them. The classificationEngine also ensures that the modules are executed in the correct order and that the necessary data is passed on from module to module. At last, the classificationEngine also and forwards the results to D3 and the dashboard. The classification engine also ensures that the data visualization waits for additional data created by the algorithm.

5.7 FINAL RESULT

In this section, the final product will be discussed, and its functionality explained. The evaluation of the final product can be found in 6 Evaluation. During a small interim evaluation, it was decided that the significance level for flow measurements should be 1 decimal, and measurements for volume should be rounded to the nearest integer. This was implemented before evaluating the final product.

A full size image of all the figures contained in this section can be found in Appendix *: Uroflow Dashboard Images

5.7.1 Main overview

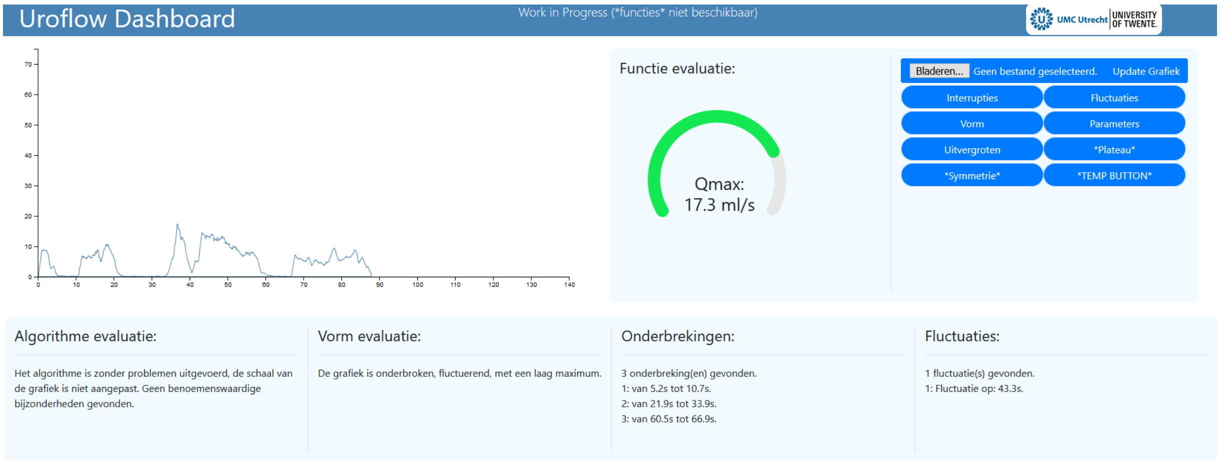


Figure 46: Uroflow Dashboard homescreen

The home screen that is presented when the user loads the Uroflow Dashboard application contains the following elements.

- The data in curve from (main visualization, line graph)
- A variable visualized in a circle gauge (secondary visualization)
- Buttons tied to a variety of functions
- 4 evaluation fields containing their respective information.

The circle gauge however, is not yet interactive, as no reference value could be found by the end of this project, further research to find such a reference is needed. A solution to this problem has been recommended in 7.3.5.3 Solution to not having a reference value for the circle gauge. For now, the circle gauge is used in a demo-like fashion, until reference variable is found and can be implemented.

Based on the function the user selects, the main visualization will update, and the secondary visualization is replaced with an info screen, earlier referred to as the dialog box. In this info screen, detailed information will displayed on how the calculation of the respective function was executed, and the results are explained based on the limit values set by the user.

5.7.2 The interruption function

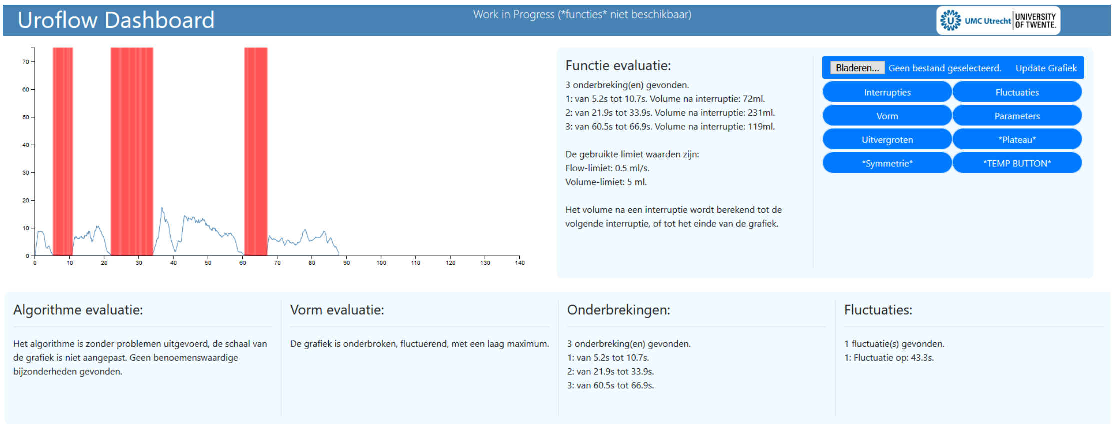


Figure 47: Uroflow Dashboard interruption function

As can be seen in the screenshot above, when the user select the interrupt function, the main visualization is updated containing red intervals, representing the interruptions found in the data. Furthermore, detailed explanations are presented in the info screen. This data contains things like the interval, from which to which second the interruption(s) lasted, and the voided volume in between the intervals or the interval and the end of signal. The info screen also displays the current set limit values, to prove these interruptions meet the current conditions to qualify as an actual interruption.

5.7.3 The fluctuation function

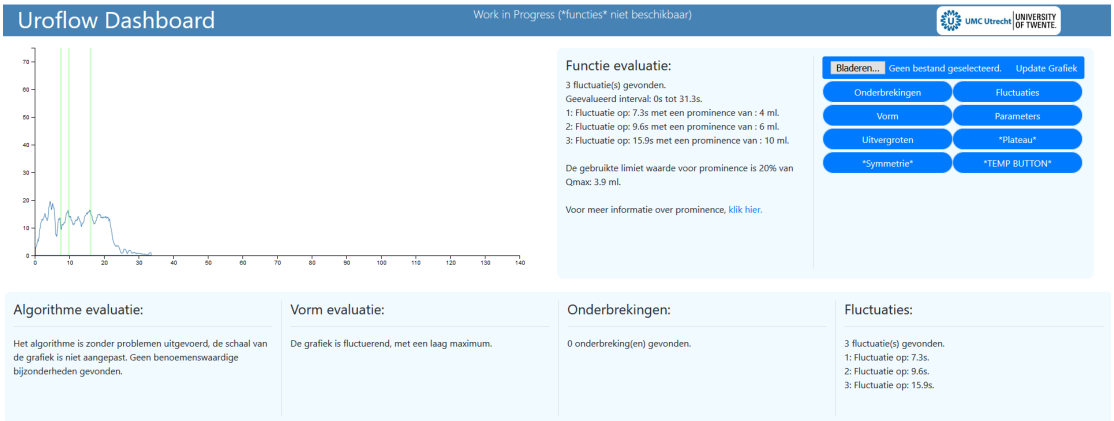


Figure 48: Uroflow Dashboard fluctuation function

The fluctuation function marks fluctuations in the graph a green line, and also provides detailed information on how the fluctuations where calculated, using what threshold. In this case, the threshold is extra important, as it is a dynamic value, representing a percentage of the maximal flow. Furthermore, detailed information is provided on which part of the curve was evaluated, as peaks only classify as fluctuations if they are present in the largest continues part of the curve.

5.7.4 Stacking mark ups.

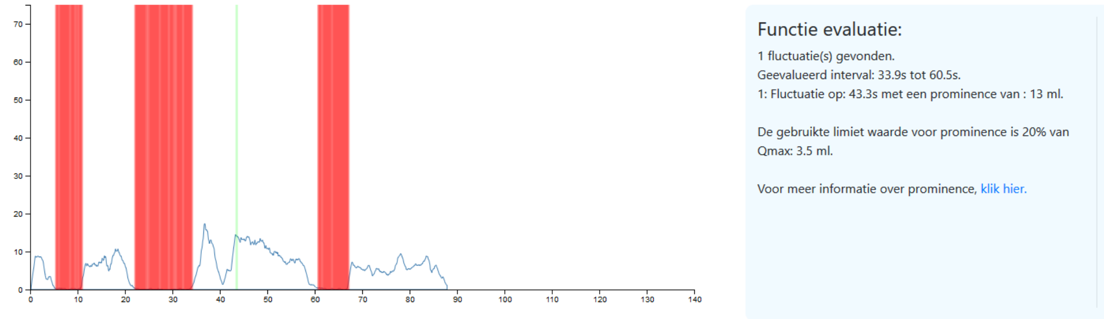


Figure 49: Stacked mark ups in the main visualization of the Uroflow Dashboard

The Uroflow Dashboard also contains the ability to stack markups, allowing the user to easily identify peaks and interruptions at the same time. Note however, that the info screen only displays detailed information for the function that was last pressed.

5.7.5 The parameter function

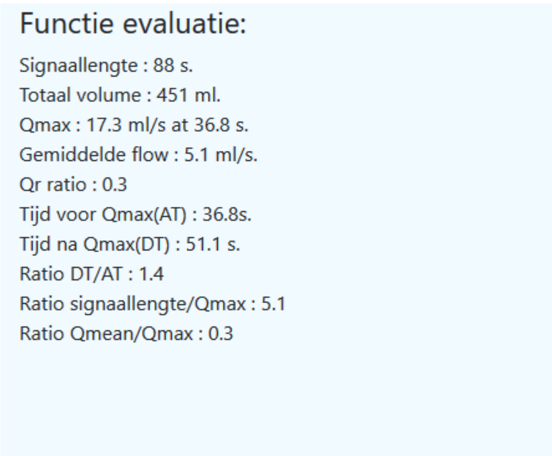


Figure 50: The info screen when the parameter function is applied

The parameter function presents all parameters related to the curve in the info screen. As no changes are made to the visualization, only a screenshot of the info screen is included.

No further calculations included, as the calculation of these parameters are basic mathematical computations, such as finding a maximum, a mean, adding and subtracting and multiplication and dividing.

No further screenshot of the parameter function is included in the appendix.

5.7.6 The zoom function

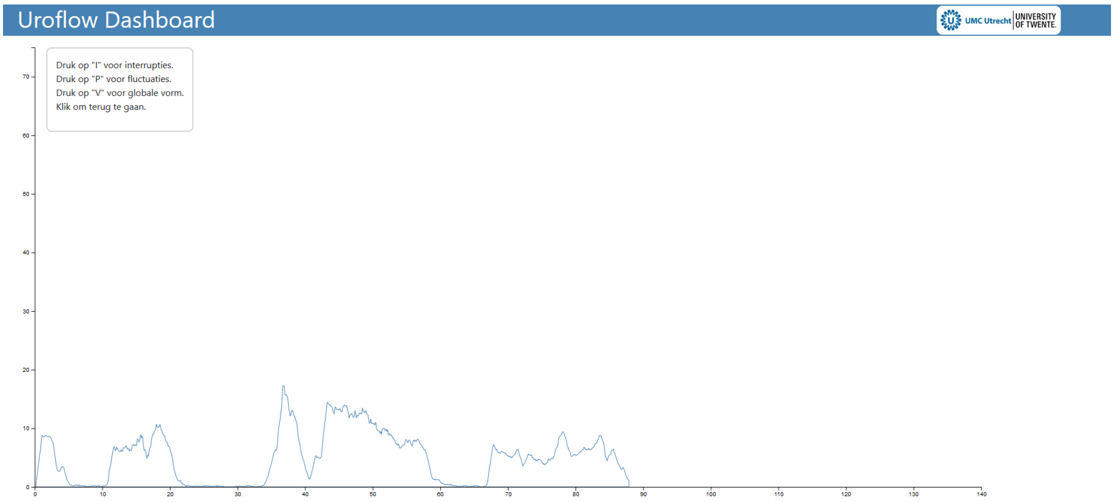


Figure 51: The zoom function

The zoom function enlarges the main visualization to a full screen view, and allows the user to enable or disable markups using the keyboard. This function was implemented to allow on more precise inspection of the graph itself, with or without markups. Clicking anywhere on the screen will revert the dashboard to the home screen. The user does still have the ability to pan and zoom on the graph, as a drag action is distinguished from a click action.

5.7.7 The shape function

The shape function provides no further detailed explanation, with the exception of also providing the shape description in the info screen. The shape description is also always visible in the bottom evaluation section, under the header “Shape evaluation”.

6 Evaluation

For the evaluation of this project, a threefold of the test was conducted. The first test is a user test based on the SUS (System Usability Scale)[49] to ensure our product is intuitive, easy to use and clear. The second method was a survey, with the same goal as the SUS test, but with more focused on feedback on design and layout.

6.1 SYSTEM USABILITY SCALE

The System Usability Scale is a quick test to rate the usability of the system. This test can be conducted using a large number of subjects in a rapid fashion. To speed up this process even more, the demo version of concept 4 accompanied by a survey to allow for online testing.

The SUS-test consists of 10 questions.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

These questions can be answered on a scale from 1-5, where 1 stands for strongly disagree, and 5 equals strongly agree. As one might have noticed, the SUS test contains positive and negative based questions, patterned in an even-odd fashion. To calculate the score of the SUS test, we take the responses on the even questions and subtract 1. This leaves us with a score of 0 ranging to 4. For the odd questions, we subtract the response from 5, again leaving us with a 0-4 score, but inverted.

Adding all the converted responses together gives us an index score, which has to be multiplied by 2.5. Please note that this is not a final grade, as the SUS test uses a curve based on the percentage of devices or products that have similar usability. An overview of how to properly grade a SUS test can be found in Table 3 SUS score to grade conversion.

SUS Score	Grade	Rating
80.3	A	Excellent
68-80.3	B	Good
68	C	Okay
51-68	D	Poor
< 51	F	Awful

Table 8 SUS score to grade conversion [50]

A more visual interpretation can be found below.

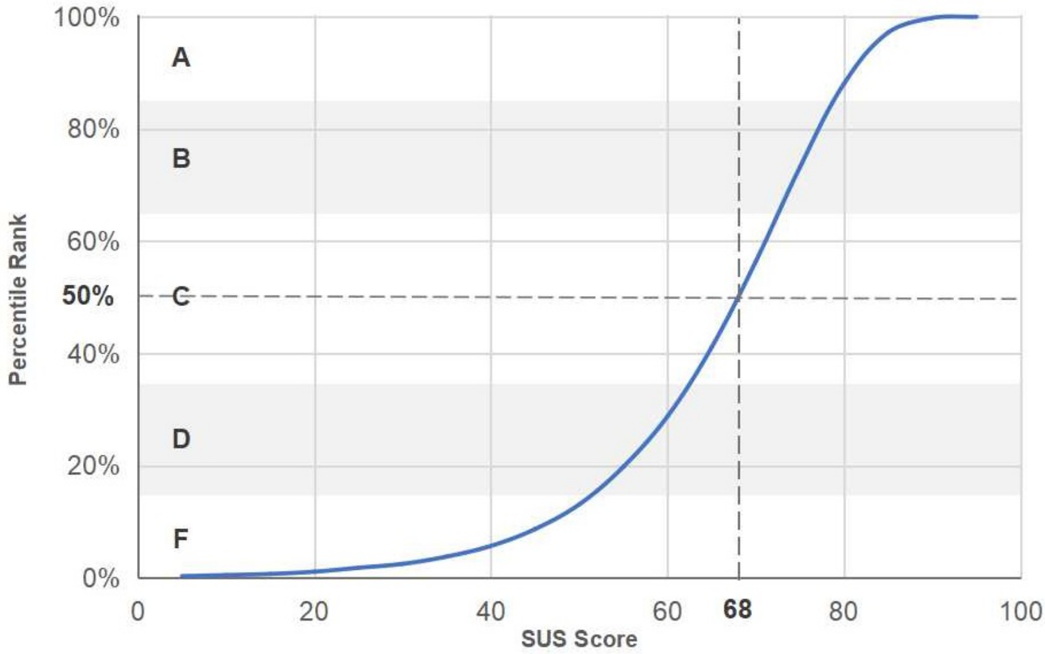


Figure 52 SUS score grading curve [51].

6.2 SURVEY ON LAYOUT AND DESIGN

The survey started out with a broad description of the project, giving the subjects a feel of what kind of product they would be evaluating. The survey consisted of 3 main categories. 1. Look and feel. 2. Clarity. 3. Evaluation. When applicable, screenshots were provided to give subjects a more in-depth insight into the dashboard.

The scope of the survey:

1. Look and feel:
    a. The interface comes across as…
        i. Messy and thrown together.
        ii. Substandard.
        iii. Alright.
        iv. Well thought out.
        v. Professionally build.
    b. The layout of the interface…
        i. Could certainly use improvement. (if so, please specify)
        ii. Is alright.
        iii. Feels like something I am used to.
        iv. Is something I would not have expected from this kind of application.
        v. Is organized.
        vi. Other…
    c. The elements on the interface…

        i. Are placed well.
        ii. Are in places I expect them to be.
        iii. Could certainly be switched around.
        iv. Feel like they do not belong in the places they are.

2. Clarity:
    a. The information displayed on the interface is..
        i. Clear and structured
        ii. Clear, but lacks structure
        iii. Feels distractive from parts that could be more important
        iv. Disorganized
    b. The interface itself presents information I would expect. (0-5)
    c. The interface is hard to look at after a while. (0-5)
    d. The sheer amount of information confuses me. (0-5)
    e. The graph is..
        i. Something I would expect from a medical application
        ii. Looks underwhelming, even with added mark ups.
        iii. Something I would not trust making my doctor a diagnosis with.
3. Evaluation
    a. What is your view on the following statements (strongly agree – strongly disagree)
        i. The system looks like it has added value for diagnosing.
        ii. The system looks like something I would trust, if my diagnosis was based on it.
        iii. The system seems practical.
        iv. I would use a similar system if one existed for my use case.

6.4 RESULTS

In this section, the findings of the user test will be evaluated and discussed. Furthermore, a short evaluation of the client and stakeholders will be presented.

6.4.1 SUS

The results of the SUS-test will be evaluated twofold. First, the average score per question will be evaluated to identify parts of the product we could improve on. Secondly, the SUS test will be evaluated as a whole.

I think I would use this system frequently (if I was an urologist).	3.5
I found the system unnecessarily complex.	3.25
I thought the system was easy to use.	3.25
I think I would need support of a technical person to be able to use this system.	3.75
I found the various functions of the system well integrated.	3.75
I thought there was too much inconsistency in the system.	3.5
I would imagine that most people would learn how to use this system very quickly.	3.25
I found the system very cumbersome (hard, a hassle) to use.	3.25
I felt very confident using the system.	3.75
I needed to learn a lot of things before I could get going with the system.	3.5

Table 9: SUS-test mean scores of individual questions

The mean scores rank from 0 to 4, where 0 is atrocious and 4 is excellent.

Based on the results above, there are not many parts of the system that can be approved upon in a favorable manner. One section that does stick out is the complexity of the system, and the users reporting they found the system hard to use.

A solution to this might be to implement a small tutorial option, where the system walks the user through the variety of functionalities, explaining what they do and how to use them.

Adding all the average scores together, and multiplying the result with 2.5 will give us a final result of 86. This indicates that our system is very easy to use, and when further developed, could be implemented using the current design, layout and functionality.

6.4.2 Survey on layout and design

The survey on layout and design presented similar results as the SUS-test. The overall clarity of the dashboard is scored relatively high, and users are easily accustomed to the layout. The overall result presents our dashboard as a “Clear, well designed and insightful product”, where most users responded that they would use a similar product if one existed for their use case. An overview of the results of the layout and design questionnaire can be found in appendix K: Layout and design questionnaire results.

One thing that did come forward from this questionnaire, is that elements should have the ability to be placed in different positions in the dashboard. This form of customization will be recommended in future work.

6.4.3 Stakeholder and client evaluation

Overall, the client and stakeholder (Prof. de Kort, Urology department UMC) are satisfied with the overall progress of this project. The Uroflow Dashboard does provide added value by creating more insight into the classification algorithm and allows the user to easily diagnose symptoms by importing data. The fact that the Uroflow Dashboard is created as a web application does also provide significant advantages based on implementation and overall accessibility, as well as support for a multitude of devices.

However, the main visualization of the curve could be improved upon, as they would prefer more visual feedback based on the overall shape description of the curve. A solution to this will be provided in 7.3.6 Incorporating feedback in the main visualization.

7 Discussion

7.1 RESULTS OF THE USER TESTS.

Based on the user test conducted in chapter 6 Evaluation, it is safe to say that the Uroflow Dashboard does have added benefits for diagnosing symptoms in uroflowmetry, is very usable and well designed.

The user tests however, were not conducted on a team of specialist, rather by people with no medical background. A consequence of this might be that the results are not a good reflection of the Uroflow Dashboard in a medical use case. On another note, however, if a person with no medical background is able to rate this product as very usable, this should transfer well to specialist who have more knowledge on the subject.

An argument to substantiate this is that both the client and the stakeholder also described the dashboard as clear and insightful and rated it very usable.

7.2 LIMITATIONS.

Due to the limited timespan of this project and the fact the classification algorithm was still subject to change when the Uroflow Dashboard was developed, not all of the functionality could be implemented. Functions that our client would or might like to see in the future are described in section 8.3 Future work.

Another limitation is the limited amount of sample data during the development of the Uroflow Dashboard. At first, only 100 sets of sample data, containing only data of female patients, was delivered. During the end of this research, a new version of the algorithm was supplied, containing 219 data samples of both man and women. Due to the limited sample size, evaluation of the correctness of the algorithm is difficult, and therefore we must stress the fact that the Uroflow Dashboard is still in development, and users should be cautions of the correctness of the results and calculations presented in the Uroflow Dashboard.

The new version of the classification algorithm also contained extra functionality was also added that has not yet been transferred to the dashboard.

Due to the absence of constant values in uroflowmetry, the circle gauge has not yet been implemented, as it requires a constant value as a reference. A solutions to this will be discussed in 8.3 Future work.

7.3 FUTURE WORK.

As development of this product is still in an early stage, only basic core features could be implemented. As the classification algorithm is developed further, the dashboard needs to be updated a well. Also, other functions that could increase usability, or expand the scope of the dashboard has not been added yet. This is mainly due to the limited time span of this research. Therefore, a number of extra options we deem viable and of benefit to our client will be listed below and expanded upon.

7.3.1 Customizability

It is a widely known statement that one product can not satisfy all users in the niche of that product, however hard we try. Different people just have different preferences. That is why we recommend to implement minor customizability in a future version of the project. The reason we opt for minor, is that one of the requirements of our clients was for the dashboard to be easy to interpret. Users should not get tangled up in a dashboard interface they don't understand anymore, while having no clue on how to reset it to the standard view.

Examples of minor customizability include, but are not limited to, the following options.

- Offer a limited selection of multiple layouts. No highly customizable layouts where the user can drag panels around, but basic predefined layouts. This enables the user to select a layout they are comfortable with, without the risk of them creating an interface they do not understand anymore or find hard to use.
- Customize the amount of feedback that is given to the user. It is duly noted that one of the requirements is to provide detailed textual feedback, however, we can imagine that some users, after having seen this feedback numerous times, can grow tired of the selective reading that ensues out of it. Therefore, we propose a menu where the user can select specific feedback that is available at all times, and what can be accessed by a detailed view.
- Implement different color pallets, to attend to users that might experience eyestrain or suffer from colorblindness.

7.3.2 Security

Security fell outside the scope of this research but is something we should seriously consider adding, as it paves the road for a lot of more possible functions which would require storing data on the dashboard. On top of that, this is a medical application, which is to be used in a medical setting, and should be secured no matter what. For now, the dashboard will only be implemented locally at the UMC, or even on the UMC's intranet. In this case, security is not a huge risk, as everything is locally stored. However, if more hospitals develop an interest in this project, it might be beneficial to switch to central hosting, instead of a myriad of stand-alone local deployments. In this case, hospitals need to be provided with login info to access the application, and user preferences need to be stored for a more pleasant experience. This can only be done if we have a means of identifying users, thus a login system would be required.

Another risk could be that a malicious intending individual has gained access to the UMC's intranet (if they opt for an intranet deployment, instead of a local deployment) and tries to intercept data between the dashboard server and the client's host. Currently, this data is unencrypted and could pose serious privacy and security risk. To prevent this, we recommend the implementation of at least TLS or its predecessor SSL, also known as the HTTPS protocol. This way, if someone does intercept data, it will be unusable without cracking these encryptions.

7.3.3 Data storage

In a future version of the dashboard, when a state-of-the-art security system is implemented, we might want to add the functionality to store data on the dashboard. This allows for easier diagnosing between devices, and the ability to share stored data with colleagues to ask their opinion on the matter. Other useful functions that require data storage is to easily compare current patient data with older measurements without having to open multiple views or reupload data. Instead, the user could simply select data sets out of a drop-down menu. To make this feasible, a well thought out storage system is required. A proposal for this system can be found in Figure 46.

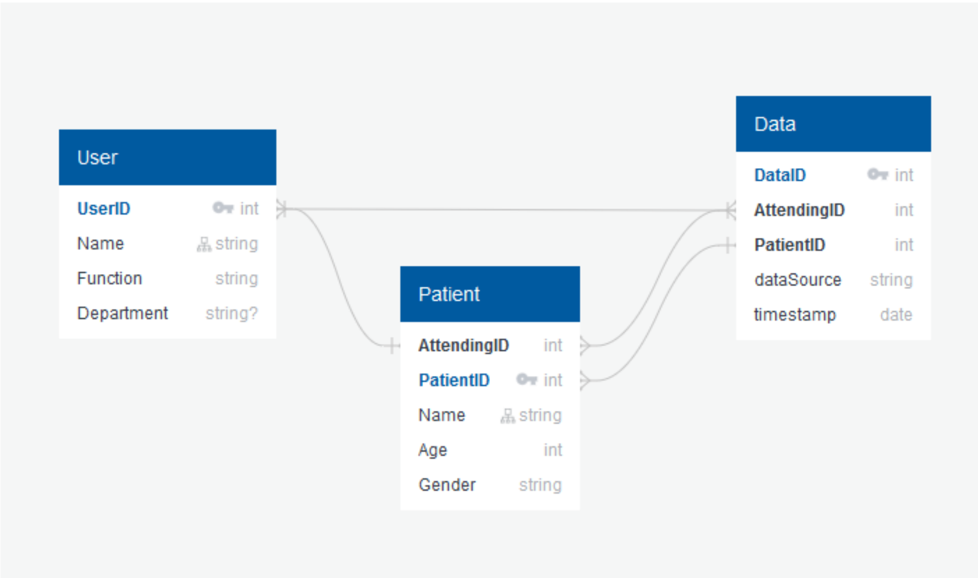


Figure 53 Proposed scheme for a database

In Figure 46, there are 3 tables. A User, a Patient, and Data. The User stores information about the user of the dashboard, in this case, an urologist or system administrator. Patients are linked to users using their AttendingID. This AttendingID will match an urologist UserID. Furthermore, data is linked both to patients and urologists using PatientID and AttendingID respectively. This system should allow the user to filter on patient data, or allow the system to show the user all data linked to his account, without first finding all connected patients.

The dashboard does already come with a MySQL database implemented, but it is currently not in use.

Another thing to keep in mind when storing data on the dashboard, is that it needs to be restricted to the attending urologist of patients. No cross-availability between users data should be allowed. This is to reduce ethical risk, as user are prevented to check on people they know without their consent, and increases privacy. This can be done using the AtteningID, and only allow access when the UserID of the user and the AttendingID on the data match.

Data storage also allows us to create more personalized experiences, as discussed in 8.3.1 Customizability. For customizability however, a less tight, easier to implement security system can be used.

7.3.4 Options to compare data

The ability to compare data could be very useful for both urologist and further development of this project. Urologist could learn more about different systems and at the same time evaluate the correctness of the classification algorithms findings.

However, to achieve this, again a state of the art security needs to be implemented first, as these kinds of functionalities are very privacy sensitive.

The patient would also need to consent to their data being used for aiding in the diagnosis in others. To motivate patients more to allow urologist to use their data, and to mitigate certain privacy risks, we could anonymize the data, only showing the data itself and the final diagnosis.

Options to compare data can be implemented in two ways. Allow for data to be imported from the database, or allow multiple uploads per session. One thing we need to keep in mind here, is that while the algorithm is modular, the visualization is not, and neither is the dashboard interface. To achieve this, the entire source code for both the visualization and dashboard interface need to be rewritten to, or either to accommodate for, the OOP protocol. This protocol is better know as Object Oriented Programming. The end result will be self-contained pieces of codes, known as classes, which allow multiple instances of that piece of code to run in parallel. In the end, all of the results of these instances could be joined into one single result that is exported to the dashboard.

7.3.5 Expanding and improving the algorithm

Due to general shortcomings in functionality and limitations and listed in 8.2 Limitations. the following expansions on the classifications algorithm and solutions to certain problems will be recommended.

7.3.5.1 Add gender differentiation

Implement a way for the algorithm to differentiate between male and female data, ensuring the correct evaluation for high or low maximum as described in 6.3.3.4 Shape description.

The correct calculation of Qref and dev for men equals:

$$Qref(male) = 6 + \left(2.37 + 0.18\sqrt{Vtot} - 0.014 * age\right)^2$$

$$dev(male) = 4 + \left(\frac{2}{500} * (Vtot - 100)\right)$$

Where Vtot is equal to the total volume voided, and the age is equal to the patients age. In the current version of the J. de Haan's classification algorithm, age is not supported as patient data, and is defaulted to a value of 40, effectively changing the formula for Qref to:

$$Qref(male) = 6 + \left(1.81 + 0.18\sqrt{Vtot}\right)^2$$

7.3.5.2 Adding plateau and symmetry functionality

As with the latest version of J de Haan's classification algorithm, the plateau functionality was introduced, and the symmetry functionality was severely changed. Due to time limitations, we were not able to adept these functions in time to implement them into the dashboard. Therefore, we recommend implementing these functions in a future version of the Uroflow Dashboard. To accommodate for the implementation of these functions, buttons describing the plateau and symmetry functions have already been implemented.

7.3.5.3 Solution to not having a reference value for the circle gauge.

As discussed before, there is no constant reference value available for any of the parameters in uroflowmetry. A work around for this is, in example for Qmax, to save the parameter to a database, and using the mean value of all entries for this parameter as a reference. Due to the fact that this solutions requires storing data on the server side of the Uroflow Dashboard, this functionality was not developed, as for now, there are no security systems in place.

When deploying this solutions, we have to keep in mind that the parameter that is stored in the database needs to be accompanied with an unique id based on the data set. If no unique id is provided, loading the same dataset over and over again will corrupt the mean value, and bring the mean value of the stored parameter closer to the same parameter in the data set. Including a unique id can prevent this, as we can check if there is already an entry in the database containing that same id. If this is the case, the parameter should not be stored in the database.

In the last stage of the development, it became apparent that Qref (as seen in section 7.3.5.1) can also be used a reference value. However, this value is not a constant, rather a reflection of the current dataset. A further note is that gender differentiation must be implemented first, for this method to work.

7.3.6 Incorporating feedback in the main visualization.

During a last evaluation interview with prof. de Kort (department of Urology UMC), another approach of incorporating feedback into the main visualization was proposed to incorporate in further developed versions of the project. The main goal of this new way of visualizing the curve is to nudge urologist into providing a universal shape description, even without added textual feedback of the classification algorithm.

The proposed solutions used a sort of framework in which the visualized curve will be contained. This framework will contain boundaries based on the results of the classification algorithm being executed on the visualized curve. This way, the user can easily see what boundaries are crossed, and what not, allowing for easy shape identification.

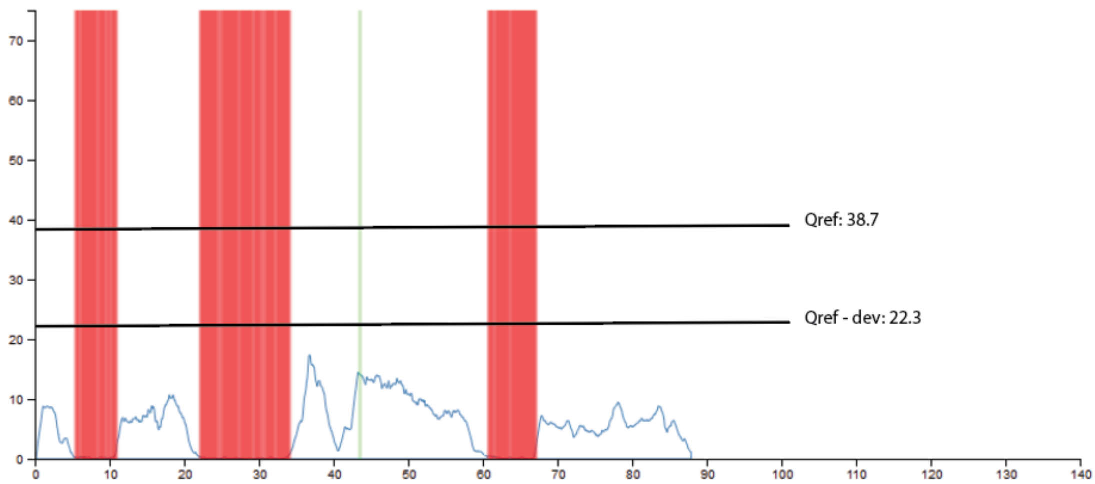


Figure 54: Shape Description framework example

An example of the beginning of such a framework can be found in Figure 54: Shape Description framework example. In this case, the Qref value has been included, which is used to determine if a graph has a high, regular or low maximum. Based on the example above, the user should be able to see in one glance that the curve has a low maximum, as its highest value lies beneath the Qref-dev line. Qref-dev stands for the calculated Qrev value – its accompanying deviation.

This framework could be further expanded using the plateau and symmetry functionality, as the interruption and fluctuation function already contribute to this framework.

7.4 ETHICAL EVALUATION.

An ethical evaluation report has been created during the development of this project. This detailed report can be found in Appendix I: Ethical evaluation. In this section, a small summary will be given of the detailed report.

During the ethical evaluation, several ethical and/or moral risk have been discovered, the major risks are:

- The Uroflow dashboard is being used as a diagnosing tool, instead of and aid for diagnosing. This causes urologists to slack at work, making their diagnosis less accurate.
- The algorithm is altered in such a way, that harm could be caused to patients.
- Urologist become too reliant on the system, and cannot identify symptoms manually if the service becomes unavailable.
- If patient data is stored on the server side of a future version of the Uroflow Dashboard, it could be leaked.
- Patients lose trust in their attendings, in this case urologist, if they rely too much on a digital tool, instead of their own insights.
- The classification algorithm can present a faulty result. If an urologist is not paying enough attention, this might cause harm to the patient.

Solutions or ways to mitigate the risk have been proposed to solve some of the problems listed above. In addition, legal standing grounds are also recommended and explained, stating we should do everything in our power to prevent data leaks, if in future versions data is to be stored on the dashboard, and take full responsibility until the perpetrator that caused the data leak is identified.

8 Conclusion

This research was aimed at developing an accessible and easy to deploy solution that visualizes symptoms in uroflowmetry curves and provides a way to universally describe flow patterns to aid urologist in the diagnostic process, while implementing J. de Haan's classification algorithm in a modular way.

This was achieved by implementing codesign based on 4 different concept iterations and evaluating the new design requirements that arose from these codesign sessions. Design choices of these concepts and the final product were based on findings in both the state-of-the-art and the literature review. Afterwards, the product was evaluated with our client and a stakeholder, as well as conducting a questionnaire on design and layout and a System Usability Scale test. Both the tests and feedback from the client and stakeholder showed promising results for the future.

As it is clear the product is still in its early stages, there is a lot that needs to be improved upon, such as improving the amount of feedback based on shape description the main visualization gives, as well as adding more functionality that is also found in the newer versions of J de Haan's classification algorithm. Furthermore, measures should be implemented to mitigate the ethical risks found during the ethical evaluation and to fortify the moral standing ground of the Uroflow Dashboard, ensuring it is, will or can - not (be) abused or misused in anyway that could harm a patient in a physical or any other way.

In conclusion, the project was a successful at building a well-established first prototype, which both the stakeholder and client would like to see further developed, incorporating the additional elements mentioned above.

9 References.

[1] J. van Haaren, “Automatic component based analysis of uroflowmetry curves in women,” 2019.

[2] C. D’Ancona *et al.*, “The International Continence Society (ICS) report on the terminology for adult male lower urinary tract and pelvic floor symptoms and dysfunction,” *Neurourol. Urodyn.*, vol. 38, no. 2, pp. 433–477, 2019.

[3] S. Hong *et al.*, “Spatiotemporal laser speckle flowmetry based on elastic-walled U-shaped tubing apparatus: optical method for urinary flow measurement,” *Opt. Eng.*, vol. 57, no. 10, p. 1, 2018.

[4] W. Schfer *et al.*, “Good Urodynamic Practices: Uroflowmetry, filling cystometry, and pressure-flow studies,” *Neurourol. Urodyn.*, vol. 21, no. 3, pp. 261–274, 2002.

[5] R. Li, Q. Zhu, M. Nibouche, and A. Gammie, “Urine flow rate curve shapes and their descriptors,” *Neurourol. Urodyn.*, vol. 37, no. 8, pp. 2938–2944, 2018.

[6] J. De Haan, “Clinical validation of algorithmic classification of female uroflowmetry curve patterns Author,” 2018.

[7] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, “Challenges in visual data analysis,” *Proc. Int. Conf. Inf. Vis.*, pp. 9–14, 2006.

[8] R. Kosara, H. Hauser, and D. L. Gresh, “An interaction view on information visualization. State-of-the-Art Report,” *Eurographics*, 2003.

[9] C. Nikolaou *et al.*, “Sextant: Visualizing time-evolving linked geospatial data,” *J. Web Semant.*, vol. 35, pp. 35–52, 2015.

[10] W. U. Lixin, J. Yu, Y. Yang, and Y. Jia, “Spatial big data organization, access and visualization with ESSG,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 40, no. 4W2, pp. 51–56, 2013.

[11] B. Craft and P. Cairns, “Beyond guidelines: What can we learn from the visual information seeking mantra?,” *Proc. Int. Conf. Inf. Vis.*, vol. 2005, pp. 110–118, 2005.

[12] A. Figueiras, “Towards the understanding of interaction in information visualization,” *Proc. Int. Conf. Inf. Vis.*, vol. 2015-Septe, pp. 140–147, 2015.

[13] D. A. Keim, “Information visualization and visual data mining,” *IEEE Trans. Vis. Comput. Graph.*, vol. 8, no. 1, pp. 1–8, 2002.

[14] J. S. Yi, Y. A. Kang, J. T. Stasko, and J. A. Jacko, “Toward a deeper understanding of the role of interaction in information visualization,” *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1224–1231, 2007.

[15] B. Shneiderman, “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations,” *Cr. Inf. Vis.*, pp. 364–371, Jan. 2003.

[16] J. P. Wanderer, S. E. Nelson, J. M. Ehrenfeld, S. Monahan, and S. Park, “Clinical Data Visualization : The Current State and Future Needs,” *J. Med. Syst.*, pp. 1–9,

2016.

[17] J. F. Rodrigues, A. J. M. Traina, M. C. F. De Oliveira, and C. Traina, “The spatial-perceptual design space: A new comprehension for data visualization,” *Inf. Vis.*, vol. 6, no. 4, pp. 261–279, 2007.

[18] C. A. Tam *et al.*, “Critical Analysis of the Use of Uroflowmetry for Urethral Stricture Disease Surveillance,” *Urology*, vol. 91, pp. 197–202, 2016.

[19] T. Blascheck *et al.*, “Exploration strategies for discovery of interactivity in visualizations,” *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 2, pp. 1407–1420, 2019.

[20] J. Preece, “Theories for Understanding Information Visualization,” in *The Craft of Information Visualization*, Elsevier, 2003, pp. 349–351.

[21] “A voice-activated uroflowmetry application - IT RATE.” [Online]. Available: https://itrate.co/portfolio-items/a-voice-activated-uroflowmetry-application/. [Accessed: 01-Nov-2019].

[22] “SPU: A portable uroflowmetry device named P-Flowdiary® combined with frequency volume charts is useful in clarifying lower urinary tract symptoms in children.” [Online]. Available: https://spuonline.org/abstracts/2017/MP28.cgi. [Accessed: 01-Nov-2019].

[23] “Uroflow Tracings.” [Online]. Available: https://uroflowtracings.com/web/WEBRun.php?page=88. [Accessed: 01-Nov-2019].

[24] “Minze Health - Hospiflow & Homeflow - Uroflowmetry and voiding diaries.” [Online]. Available: https://minzehealth.com/. [Accessed: 01-Nov-2019].

[25] D. Porru *et al.*, “Home and office uroflowmetry for evaluation of LUTS from benign prostatic enlargement,” pp. 45–49, 2005.

[26] S. Arora and R. Ahlawat, “Chapter-11 Urology for Nephrologist,” no. November 2017, 2016.

[27] A. . Mader and W. Eggink, “A design process for Creative Technology,” *Proc. 16th Int. Conf. Eng. Prod. Des. Educ.*, pp. 568–573, 2014.

[28] “MoSCoW prioritization technique - very useful to prepare for Agile exams.” [Online]. Available: https://www.volkerdon.com/pages/moscow-prioritisation. [Accessed: 08-Jan-2020].

[29] “Download free Adobe XD | UX/UI design and collaboration tool.” [Online]. Available: https://www.adobe.com/nl/products/xd.html?sdid=91BF525M&mv=search&ef_id=CjwKCAiAu9vwBRAEEiwAzvjq-_OGCt2BuiYNoR7nOlhXlqPalGCt9joEoP9RTlfIMgllZRvg_AWxpRoCt6IQAvD_BwE:G:s&s_kwcid=AL!3085!3!394597829420!e!!g!!adobe xd. [Accessed: 09-Jan-2020].

[30] “Bootstrap · The most popular HTML, CSS, and JS library in the world.” [Online]. Available: https://getbootstrap.com/. [Accessed: 09-Jan-2020].

[31] "jQuery UI Components - Build Better Apps Faster - Progress Kendo UI." [Online]. Available: https://www.telerik.com/kendo-jquery-ui. [Accessed: 04-Nov-2019].

[32] "Comparing D3 to Kendo UI for JavaScript Data Visualization." [Online]. Available: https://www.telerik.com/blogs/javascript-charts-comparing-d3-to-kendo-ui-for-data-visualization. [Accessed: 04-Nov-2019].

[33] "Kendo UI jQuery-based Area Charts Demo." [Online]. Available: https://demos.telerik.com/kendo-ui/area-charts/index. [Accessed: 03-Jan-2020].

[34] "D3.js - Data-Driven Documents." [Online]. Available: https://d3js.org/. [Accessed: 04-Nov-2019].

[35] "Stack Overflow - Where Developers Learn, Share, & Build Careers." [Online]. Available: https://stackoverflow.com/. [Accessed: 03-Jan-2020].

[36] "Across U.S. Companies, Tax Rates Vary Greatly - Interactive Feature - NYTimes.com." [Online]. Available: https://archive.nytimes.com/www.nytimes.com/interactive/2013/05/25/sunday-review/corporate-taxes.html. [Accessed: 03-Jan-2020].

[37] "Hierarchical Edge Bundling / D3 / Observable." [Online]. Available: https://observablehq.com/@d3/hierarchical-edge-bundling. [Accessed: 03-Jan-2020].

[38] "Calendar View / D3 / Observable." [Online]. Available: https://observablehq.com/@d3/calendar-view. [Accessed: 03-Jan-2020].

[39] "react-vis." [Online]. Available: https://uber.github.io/react-vis/. [Accessed: 04-Nov-2019].

[40] "WebGL - Wikipedia." [Online]. Available: https://nl.wikipedia.org/wiki/WebGL. [Accessed: 03-Jan-2020].

[41] "react-vis." [Online]. Available: https://uber.github.io/react-vis/examples/showcases/plots. [Accessed: 03-Jan-2020].

[42] "Power BI | Microsoft Power Platform." [Online]. Available: https://powerbi.microsoft.com/nl-nl/. [Accessed: 04-Nov-2019].

[43] "Wat is Power BI | Microsoft Power BI." [Online]. Available: https://powerbi.microsoft.com/nl-nl/what-is-power-bi/. [Accessed: 08-Jan-2020].

[44] "BI2.png (PNG-afbeelding, 1260 × 789 pixels)." [Online]. Available: https://www.dynamicpeople.nl/wp-content/uploads/2019/10/BI2.png. [Accessed: 08-Jan-2020].

[45] "USBWebserver – Download USBWebserver." [Online]. Available: https://www.usbwebserver.net/webserver/. [Accessed: 10-Jan-2020].

[46] "Sublime Text - A sophisticated text editor for code, markup and prose." [Online]. Available: https://www.sublimetext.com/. [Accessed: 11-Jan-2020].

[47] "Grid system · Bootstrap." [Online]. Available: https://getbootstrap.com/docs/4.0/layout/grid/. [Accessed: 11-Jan-2020].

[48] "Prominence - MATLAB & Simulink - MathWorks Benelux." [Online]. Available: https://nl.mathworks.com/help/signal/ug/prominence.html. [Accessed: 20-Jan-2020].

[49] "MeasuringU: Measuring Usability with the System Usability Scale (SUS)." [Online]. Available: https://measuringu.com/sus/. [Accessed: 12-Jan-2020].

[50] "How to Measure Product Usability with the System Usability Scale (SUS) Score." [Online]. Available: https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f. [Accessed: 12-Jan-2020].

[51] "MeasuringU: 5 Ways to Interpret a SUS Score." [Online]. Available: https://measuringu.com/interpret-sus-score/. [Accessed: 12-Jan-2020].

[52] "cell2csv - File Exchange - MATLAB Central." [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/7601-cell2csv. [Accessed: 10-Jan-2020].

[53] "1578420464477.png (PNG-afbeelding, 1062 × 542 pixels) - Geschaald (90%)." [Online]. Available: https://s3.amazonaws.com/thinkific/file_uploads/104413/images/9ed/427/422/1578420464477.png. [Accessed: 08-Jan-2020].

10 List of figures.

11 List of tables

Appendix A: First explorative interview.

In this appendix, a summary will be given of the first explorative interview. Parts that contain background information but have no added value for the project as a whole are abbreviated and left out:

At first, it seemed that only a visualization was required, which could be created by extracting all the data from the classification algorithm in MATLAB, and presenting it in an intuitive and understandable way for both the urologist and patient, without impairing the urologist ability to diagnose based on said representation of the data. The main focus however, should be on the urologist. The characteristics, in this case basic parameters, of a curve are easy to resolve from the data and are currently displayed next to the graph (in the MATLAB version). Examples of such characteristics are Qmax, signal time, etc. This is done automatically by the classification algorithm in MATLAB. Another thing the algorithm does, is describe the shape of the curve, such as containing serrated edges, and interruptions. This is also of great importance to the patient, as questions like; “For how long have you been experiencing problems?”, “What will the next diagnosis be?” and “What kind of medication needs to be administered?” can be asked based on this. What came to mind when looking at the serrated edges, is that the measurement frequency is of great importance. If the time between measurements is too large, peaks might disappear or merge. A high measurement resolutions is required, both for the correctness of the data, and to provide the ability to zoom in on a segment of the graph. An interactive graph is preferred. Job de Haan’s task during this project is to define general terms for parameters in the algorithm, so every urologist can copy the results and discuss them, without creating confusion, and creating unity within curve description amongst urologists. In essence, Jobs algorithm will define the shape of the curve and its parameters, and my task is to visualize the curve accordingly.

Experiments using machine learning to classify the curves have also been conducted, but were shortly after stopped. The main reason for this was that the attending urologist lost insight and control in the diagnosis the machine learning algorithm presented, as there was no clear way to validate the output, or figure out what was wrong with the algorithm when the output was not valid. The leads to the requirement that the project should give the urologist a clear insight into the choices the algorithm has made. This is another reason why the algorithm diverted from machine learning, as the choices the algorithm made could not be backed up by hard data. The current version of the algorithm uses a limit value approach, which allows us to explain the choices of the algorithm.

… Explanation of the machine learning algorithm that is no longer used has been left out, as machine learning for this algorithm is deprecated….

The curve in the data visualization should consist of a clear line, while still presenting features that are of essence to the curve. The scale of the axis should not change, and correct time and values should be displayed at all times, even when zooming and or panning on the graph.

The core concept of the project becomes more clear, make the classification algorithm more insightful for urologist, using visualizations and explanations. The current version of the algorithm in MATLAB presents to much data at once, the idea of functions is presented. This will lead to the implementations of buttons during the project, resulting in specific data only being presented when required.

The development of an diagnostic aid is stressed, as opposed to the development of a diagnostic tool. The project should not present a diagnosis, but draw the attention to found symptoms and/or problems.

The idea of extracting data from MATLAB, including symptoms and results is presented. MATLAB should provide a file with all the information required for a data visualization. This feature was never implemented in the algorithm, which was one of the reasons, next to increasing the usability and deployment possibilities, to recreate the algorithm and incorporate it into the Uroflow Dashboard.

In the current version of the algorithm, there are function buttons. Pressing, for instance, the fluctuation button will show fluctuations, but now how the fluctuation is calculated. Showing the calculation of fluctuations, and other points of interest for that matter, is important, and the goal of the project. The question the users should ask themselves is: "Why does the classification algorithm show this as a symptom", where the goal is that the product to be developed should have the ability to answer this question of the user.

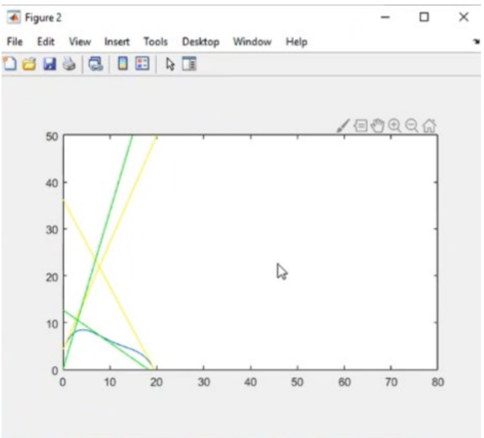
… Explanation of how uroflowmetry works, such as the measurement frequency…

The goal is to (re)create the algorithm in such a way that it explains itself. This should be incorporated in the final product with an accompanying data visualization.

In the end, the product to be developed should provide a way for an urologist to describe a flow shape in an universal way, to create a common understanding of these to be universal terms between urologist. An additional usage the product might have is to teach urologist in training how to describe flow shapes in general, using this universal standard provided by the algorithm.

Appendix B: Skype interview on extra functionality

During a skype interview with J. De Haan, we discussed extra functionality that could be incorporated in (a future version of) the Uroflow Dashboard. Functionalities that came to mind were mostly based on new ways to calculate the symmetry of the graph, something that is not available in the current state of the Uroflow Dashboard.



A new way to calculate these symmetries would be to introduce differentiated curves, based on the to evaluated curves, and use slope coefficients to draw slope angle lines. Based on where these lines would cross, above the middle of the graph or not, symmetry would be determined.

An example can be seen to the left, based on the order of the polynomial of the derivative curve, the symmetry will differ. Evaluating the green lines would result in a not symmetrical curve, where evaluating the yellow lines would result in a symmetrical curve, as the intersection is close to the median measurement time in the graph.

Other things that were discussed during this interview were changes in the calculations of limit values. For instance, during the start of this project, there were 3 different ways to calculate the threshold a peak should overshoot to classify as a fluctuation. This was changed to one static method.

Furthermore, the concept of prominence was introduced, which forced us to evaluate peaks based on surrounding peaks, instead of the peak to be evaluated on its own.

Appendix C: List of online resources.

Concept 4 live demo:

https://xd.adobe.com/view/4c12ca4c-3d41-4a25-46a1-46bcb3896ca6-3a23/screen/cccbde4b-ab92-4f05-9162-67e3e3f76587/Web-1920-5

Tool used for drawing the proposed database structure:

https://app.quickdatabasediagrams.com/#/

Tool used for drawing other diagrams:

https://draw.io

Text editor used for developing the project:

https://sublimetext.com

USBWebserver:

https://usbwebserver.net/webserver

Bootstrap:

https://getbootstrap.com/

D3:

https://d3js.org/

Appendix D: MATLAB to csv conversion script

In this appendix, the MATLAB to csv conversion script will be presented and explained. General knowledge of some of the variables and plugins present in the script will be provided.

Variable/plugin	Description
Headers	A 1x2 string array used to put headers above the data itself, to ensure D3 can read the file properly
Testcurves100N	A 100X1 array containing 100 nX2 arrays of varying length n. These last arrays are where the sample data is stored
Cell2csv	An expansion plugin on MATLABS csvwrite() command. Cell2csv also allows for strings to be interpreted and written. This is necessary, as the headers are strings. Cell2csv can be found here: https://nl.mathworks.com/matlabcentral/fileexchange/7601-cell2csv [52]

The script itself:

```
headers = ["time", "value"]
for i = 1:length(testcurves100N)
        testcurves100N{i,1} = [headers; testcurves100N{i,1}]
end
for i = 1:length(testcurves100N)
        cell2csv("curve"+i+".csv", testcurves100N{i,1})
end
```

In the first for loop, which cycles from 1 to the number of data points in the variable testcurves100N, the headers will be added in front of all the data contained in the arrays in testcurves100N. The variable will be overwritten containing the headers.

In the second for loop, which also cycles from to the end of testcurves100N, every array will be grabbed and exported under the name curve + i.csv. i is the number of the array in testcurves100N. As an example, the 7th array in testcurves100N is exported. The resulting output will be curve7.csv

Appendix E: Setting up the basics.

To initialize both bootstrap and D3 in our project, we need a basic code framework contained in a page. Within this code framework, our project can be created, utilizing the functionality of D3 and bootstrap together.

```
<html>
<head>
<meta charset=”utf-8”>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script src=”anexamplescript.js”></script>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
</head>
<body>

<svg width=”900” height=”900></svg>

</body>
</html>
```

When copying the code above to a new html file, the user is free to add anything they would like between the <body></body> tags. Everything that is added between these tags can be incorporated into, or use functionality from bootstrap and D3.

Appendix F: HTML source code for the bootstrap grid.

In this appendix, the source code for the bootstrap grid is presented. Note, only the body can be found here, as that is the part that constructs the grid. As al bootstrap elements will resize automatically to accommodate their contents, the grid appears to be squished vertically. This will resolve itself later on, as content is added.

```
<body>
    <div class="container-fluid">
        <div class="row ">
            <div class="banner">
                BANNER FIELD
            </div>
        </div>
        <div class="row ">
            <div class="col-md-6 ">
                GRAPH FIELD
            </div>
            <div class="col-md-6 ">
                <div class="row ">
                    <div class="col-md-6  ">
                        NESTED: GAUGE AND EXPLANATION FIELD
                    </div>
                    <div class="col-md-6  ">
                        NESTED: BUTTON FIELD
                    </div>
                </div>
            </div>
        </div>
        <div class="row ">
            <div class="col-md-3  ">
                FEEDBACK 1/4
            </div>
            <div class="col-md-3  ">
                FEEDBACK 2/4
            </div>
            <div class="col-md-3  ">
                FEEDBACK 3/4
            </div>
            <div class="col-md-3  ">
                FEEDBACK 4/4
            </div>
        </div>
    </div>
</body>
```

Appendix G: Interactivity overview Concept 4.

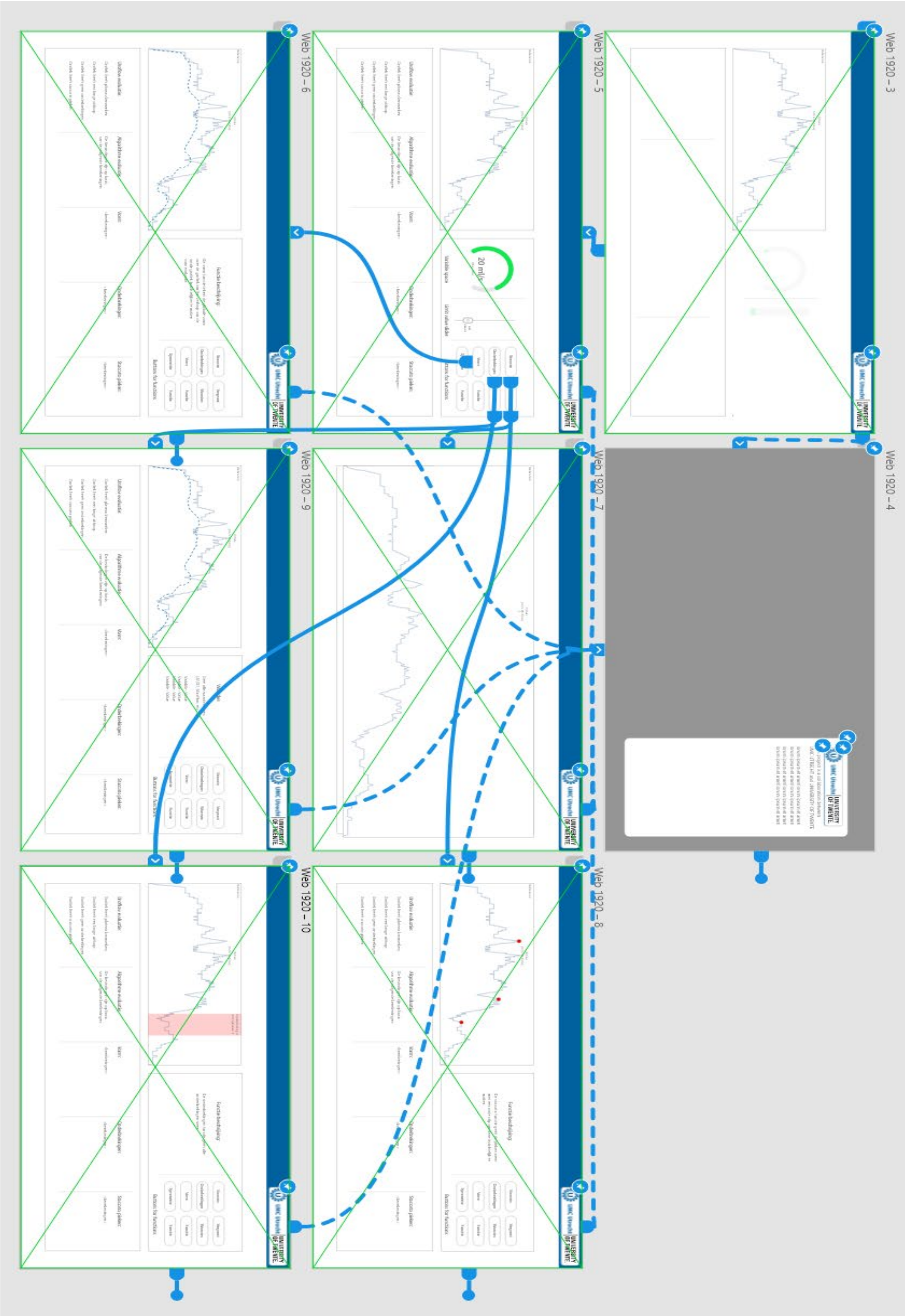


Figure 55 Concept 4 interactivity overview (Adobe XD)

Appendix H: Source code

In this appendix, all the source code for the project will be listed. The appendix will be divided into parts, namely dashboard and algorithm. Furthermore, the algorithm sections will be split corresponding to the functionality said part of the algorithm provides.

SOURCE CODE FOR THE ALGORITHM

UPLOAD.JS

The following pieces of code allows for uploading data to the dashboard, and acts as a temporarily storage buffer to load data into D3 and the algorithm. The upload script consist out of both a Javascript piece to fetch the data, and a php piece to save the data in a temporary buffer.

Upload.js(JAVASCRIPT)

```
const url = 'process.php';
const form = document.querySelector('form');
form.addEventListener('submit', e => {
  e.preventDefault();
  const files = document.querySelector('[type=file]').files;
  const formData = new FormData();
  for (let i = 0; i < files.length; i++) {
    let file = files[i];
  console.log(file);
    ext = file.name;
    dotloc = ext.lastIndexOf(".");
    extSize = ext.length;
    subLength = extSize - dotloc;
    ext = ext.substr(-subLength);
    console.log(ext);
    const newFile = new File([file], 'data' + ext, {type: file.type});
    console.log(newFile);
    formData.append('files[]', newFile);
  }
  fetch(url, {
    method: 'POST',
    body: formData
  }).then(response => {
    console.log(response);
  });
  setTimeout(() => {window.location.reload(true);}, 1000);
});
```

Process.php(PHP)

```
<?php

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
   if (isset($_FILES['files'])) {
      $errors = [];
      $path = 'data/';
	      $extensions = ['pdf', 'csv', 'json'];

      $all_files = count($_FILES['files']['tmp_name']);

      for ($i = 0; $i < $all_files; $i++) {
			$file_name = $_FILES['files']['name'][$i];
			$file_tmp = $_FILES['files']['tmp_name'][$i];
			$file_type = $_FILES['files']['type'][$i];
			$file_size = $_FILES['files']['size'][$i];
			$file_ext = strtolower(end(explode('.', $_FILES['files']['name'][$i])));
$file = $path . $file_name;
if (!in_array($file_ext, $extensions)) {
	$errors[] = 'Extension not allowed: ' . $file_name . ' ' . $file_type;
}
if ($file_size > 2097152) {
	$errors[] = 'File size exceeds limit: ' . $file_name . ' ' . $file_type;
}
if (empty($errors)) {
	move_uploaded_file($file_tmp, $file);
}
}
if ($errors) print_r($errors);
   }
header("Refresh:0");}
```

GRAPH.JS (D3)

In this section, the code for the visualization can be found. Note that D3 is also run as a main script and contains functions used for toggling parts of the visualization on and off, which are combined with the functionality of displaying more detailed data.

```
var svg = d3.select("svg"),
    margin = {top: 20, right: 20, bottom: 20, left: 40},
    margin2 = {top: 230, right: 20, bottom: 30, left: 40},
    width = +svg.attr("width") - margin.left - margin.right,
    height = +svg.attr("height") - margin.top - margin.bottom,
    height2 = +svg.attr("height") - margin2.top - margin2.bottom;

//var parseDate = d3.timeParse("%b %Y");

var rectData = new Array();
var peakData = new Array();
var meanData = new Array();

var x = d3.scaleLinear().range([0, width]),
    x2 = d3.scaleLinear().range([0, width]),
    y = d3.scaleLinear().range([height, 0]),
    y2 = d3.scaleLinear().range([0, 0]);

var xAxis = d3.axisBottom(x),
    xAxis2 = d3.axisBottom(x2),
    yAxis = d3.axisLeft(y);

var brush = d3.brushX()
    .extent([[0, 0], [width, height2]])
    .on("brush end", brushed);

var zoom = d3.zoom()
    .scaleExtent([1, Infinity])
    .translateExtent([[0, 0], [width, height]])
    .extent([[0, 0], [width, height]])
    .on("zoom", zoomed);

var area = d3.area()
    .curve(d3.curveMonotoneX)
    .x(function(d) { return x(d.time); })
    .y0(height)
    .y1(function(d) { return y(d.value); });

var shape = d3.line()
    .curve(d3.curveBundle.beta(0.4))
    .x(function(d) { return x(d.time); })
    .y(function(d) { return y(d.value); })

var area2 = d3.area()
    .curve(d3.curveMonotoneX)
    .x(function(d) { return x2(d.time); })
    .y0(height2)
    .y1(function(d) { return y2(d.value); });

svg.append("defs").append("clipPath")
    .attr("id", "clip")
    .append("rect")
    .attr("width", width)
    .attr("height", height);

var focus = svg.append("g")
```

```
    .attr("class", "focus")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var context = svg.append("g")
    .attr("class", "context")
    .attr("id", "contextID")
    .attr("transform", "translate(" + margin2.left + "," + margin2.top + ")");

d3.csv("data/data.csv", type, function(error, data) {
  if (error) throw error;
  window.data = data;

  var waitForGlobal = function(key, callback) {
  if (window.dataInterrupt) {
    callback();
  } else {
    setTimeout(function() {
      waitForGlobal(dataInterrupt, callback);
    }, 100);
  }
};

 waitForGlobal("jQuery", function() {
  threshold = window.dataInterrupt;
  console.log("DATA FOUND");
  peakData = window.dataPeak;
  rectData = window.dataInterrupt;
  if(rectData){
  calculateSVG(data);
  }
 });
});

var threshold = 0;

function calculateSVG(data){
  checkCookie();

  console.log("Running D3");

  var temp = 0;
  var samp = data[data.length-1].time;
  var maxD = d3.max(data, function(d) { return d.value;});
  console.log(data[data.length-1].time);
  if(samp< 50){
    x.domain([0,50]);
  }
  else{
   x.domain(d3.extent(data, function(d) { return d.time;}));
  }
  console.log(x.domain());
  if(x.domain()[1] < 75){
   y.domain([0, x.domain()[1]]);
  }
  else{}
  if(d3.extent(data, function(d) { return d.value}) > 75){
    console.log("DOMAIN UPDATED!");
    y.domain(d3.extent(data, function(d) { return d.value}));
  }
  else{
    y.domain([0,75]);
```

```
 }
 x.domain([0,140]);
 x2.domain(x.domain());
 y2.domain(y.domain());

 for (var i = 0; i < data.length-8; i = i + 8) {
   var meanObject = {
     "time" :  (data[i].time + data[i+1].time + data[i+2].time + data[i+3].time + data[i+5].time + data[i+5].time +
data[i+6].time + data[i+7].time)/8,
     "value" : (data[i].value + data[i+1].value + data[i+2].value + data[i+3].value + data[i+4].value + data[i+5].value
+ data[i+6].value + data[i+7].value)/8
   };
   meanData.push(meanObject);
 }
 console.log("MEAN DATA");
 console.log(meanData);
 console.log(data)

 focus.append("path")
     .attr("clip-path", "url(#clip)")
     .datum(data)
     .attr("class", "area")
     .attr("d", area);

 focus.append("path")
     .attr("clip-path", "url(#clip)")
     .datum(meanData)
     .attr("class", "line")
     .attr("stroke", "rgb(255,0,0)")
     .attr("stroke-width", 2)
     .attr("fill", "none")
     .attr("d", shape);

 focus.selectAll("myRects")
     .data(rectData)
     .enter()
     .append("rect")
       .attr("class", "rects")
       .attr("fill", "rgba(255,0,0,0.2)")
       .attr("stroke", "red")
       .attr("stroke", "none")
       .attr("x", function(d) {return x(d)})
       .attr("y",0)
       .attr("width",3)
       .attr("height", height);

   focus.selectAll("myRects2")
     .data(peakData)
     .enter()
     .append("rect")
       .attr("class", "rects2")
       .attr("fill", "rgba(0,255,0,0.2)")
       .attr("stroke", "blue")
       .attr("stroke", "none")
       .attr("x", function(d) {return x(d)})
       .attr("y",0)
       .attr("width",3)
       .attr("height", height);

 focus.append("g")
     .attr("class", "axis axis--x")
     .attr("transform", "translate(0," + height + ")")
```

```
      .call(xAxis);

  focus.append("g")
      .attr("class", "axis axis--y")
      .call(yAxis);

  context.append("path")
      .datum(data)
      .attr("class", "area2")
      .attr("d", area2);


  context.append("g")
      .attr("class", "axis axis--x")
      .attr("transform", "translate(0," + height2 + ")")
      .call(xAxis2);

  context.append("g")
      .attr("class", "brush")
      .call(brush)
      .call(brush.move, x.range());


  svg.append("rect")
      .attr("class", "zoom")
      .attr("width", width)
      .attr("height", height)
      .attr("transform", "translate(" + margin.left + "," + margin.top + ")")
      .call(zoom);

  svg.append("rect2")
      .attr("class", "zoom")
      .attr("width", width)
      .attr("height", height)
      .attr("transform", "translate(" + margin.left + "," + margin.top + ")")
      .call(zoom);

  fixStyleZoom();
  fixStyle();
  var shapeEval = document.getElementById("shapeEval");
  shapeEval.innerHTML = window.shapeDesc;
  var interEval = document.getElementById("interEval");
  interEval.innerHTML = window.infoInterrupt;
  var flucEval = document.getElementById("flucEval");
  flucEval.innerHTML = window.infoPeak;
}


function brushed() {
  if (d3.event.sourceEvent && d3.event.sourceEvent.type === "zoom") return; // ignore brush-by-zoom
  var s = d3.event.selection || x2.range();
  x.domain(s.map(x2.invert, x2));
  focus.select(".area").attr("d", area);
  focus.selectAll(".rects")
    .attr("x", function(d){return(x(d))});
  focus.selectAll(".rects2")
    .attr("x", function(d){return(x(d))});

  focus.select(".axis--x").call(xAxis);
  svg.select(".zoom").call(zoom.transform, d3.zoomIdentity
      .scale(width / (s[1] - s[0]))
      .translate(-s[0], 0));
```

```
}

function zoomed() {
  if (d3.event.sourceEvent && d3.event.sourceEvent.type === "brush") return; // ignore zoom-by-brush
  var t = d3.event.transform;
  x.domain(t.rescaleX(x2).domain());
  focus.select(".area").attr("d", area);
  focus.select(".line").attr("d", shape);
  focus.select(".axis--x").call(xAxis);
  context.select(".brush").call(brush.move, x.range().map(t.invertX, t));
  focus.selectAll(".rects")
    .attr("x", function(d){return(x(d))});
  focus.selectAll(".rects2")
    .attr("x", function(d){return(x(d))});
}

function type(d) {
  d.time = +d.time;
  d.value = +d.value;
  return d;
}


function HideContext(){
  changeCont = document.getElementById("contextID");
  Contbtn = document.getElementById("contextbtn");
  if(changeCont.style.display == "none"){
    changeCont.style.display = "inline-block";
    Contbtn.innerHTML = "Hide Context";
    Contbtn.className = "btn btn-light";
  }
  else{
    changeCont.style.display = "none";
    Contbtn.innerHTML = "Show Context"
    Contbtn.className = "btn btn-primary";
  }
}

var shapeLine = document.getElementsByClassName('line');

function fixStyle(){
  //document.getElementById("infoScreen").innerHTML = "";
  var rect = document.getElementsByClassName('rects');
  for (var i = rect.length - 1; i >= 0; i--) {
    rect[i].style.display = "none";
  }
  var rect = document.getElementsByClassName('rects2');
    for (var i = rect.length - 1; i >= 0; i--) {
    rect[i].style.display = "none";
  }
    shapeLine[0].style.display = "none";
}

function fixStyleZoom(){
  var rect = document.getElementsByClassName('rects');
    for (var i = rect.length - 1; i >= 0; i--) {
    rect[i].style.display = "none";
  }
  var rect = document.getElementsByClassName('rects2');
    for (var i = rect.length - 1; i >= 0; i--) {
    rect[i].style.display = "none";
  }
```

```
}

function fixDialog(){
  document.getElementById("infoScreen").innerHTML = "";
}

 function showInterrupt(){
  fixDialog();
  console.log("Clicked intterupt");
    var infoScreen = document.getElementById("infoScreen");
    if(infoScreen.innerHTML == window.infoInterDet){
        infoScreen.innerHTML = "";
    }
    else{
      infoScreen.innerHTML = window.infoInterDet;
    }
  var rect = document.getElementsByClassName('rects');
  console.log(rect);
  if(rect.length > 0){
    console.log(rect[0].style.display);
    if(rect[0].style.display == "inline-block"){
      console.log("Hiding");
      for (var i = rect.length - 1; i >= 0; i--) {
        rect[i].style.display = "none";
      }
      fixDialog();
    }
    else if(rect[0].style.display == "none"){
      console.log("Showing");
      for (var i = rect.length - 1; i >= 0; i--) {
        rect[i].style.display = "inline-block";
      }
    }
  }
}

function showInterruptZoom(){
    var rect = document.getElementsByClassName('rects');
  console.log(rect);
  if(rect.length > 0){
    console.log(rect[0].style.display);
    if(rect[0].style.display == "inline-block"){
      console.log("Hiding");
      for (var i = rect.length - 1; i >= 0; i--) {
        rect[i].style.display = "none";
      }
      fixDialog();
    }
    else if(rect[0].style.display == "none"){
      console.log("Showing");
      for (var i = rect.length - 1; i >= 0; i--) {
        rect[i].style.display = "inline-block";
      }
    }
  }
}

function showPeaks(){
  fixDialog();
  console.log("Clicked intterupt");
    var infoScreen = document.getElementById("infoScreen");
    if(infoScreen.innerHTML == window.infoPeakDet){
```

```
      infoScreen.innerHTML = "";
    }
    else{
     infoScreen.innerHTML = window.infoPeakDet;
    }
  var rect = document.getElementsByClassName('rects2');
  console.log(rect);
  if(rect.length > 0){
    console.log(rect[0].style.display);
    if(rect[0].style.display == "inline-block"){
     console.log("Hiding");
     for (var i = rect.length - 1; i >= 0; i--) {
       rect[i].style.display = "none";
     }
     fixDialog();
    }
    else if(rect[0].style.display == "none"){
     console.log("Showing");
     for (var i = rect.length - 1; i >= 0; i--) {
       rect[i].style.display = "inline-block";
     }
    }
  }
}

function showPeaksZoom(){
    var rect = document.getElementsByClassName('rects2');
  console.log(rect);
  if(rect.length > 0){
    console.log(rect[0].style.display);
    if(rect[0].style.display == "inline-block"){
     console.log("Hiding");
     for (var i = rect.length - 1; i >= 0; i--) {
       rect[i].style.display = "none";
     }
     fixDialog();
    }
    else if(rect[0].style.display == "none"){
     console.log("Showing");
     for (var i = rect.length - 1; i >= 0; i--) {
       rect[i].style.display = "inline-block";
     }
    }
  }
}

function showShape(){
  var infoScreen = document.getElementById("infoScreen");
     if(infoScreen.innerHTML == window.shapeDesc){
       infoScreen.innerHTML = "";
    }
    else{
     infoScreen.innerHTML = window.shapeDesc;
    }
  if(shapeLine[0].style.display == "none"){
    shapeLine[0].style.display = "inline";
    console.log("Showing Shape");
  }
  else if(shapeLine[0].style.display == "inline"){
    shapeLine[0].style.display = "none";
  }
}
```

```
function showShapeZoom(){
  if(shapeLine[0].style.display == "none"){
    shapeLine[0].style.display = "inline";
    console.log("Showing Shape");
  }
  else if(shapeLine[0].style.display == "inline"){
    shapeLine[0].style.display = "none";
  }
}

function notAvail(){
  var unAvailablemsg = "Deze functie is helaas nog niet beschikbaar in de huidige versie van het Uroflow Dashboard";
  var infoScreen = document.getElementById("infoScreen");
      infoScreen.innerHTML = unAvailablemsg;
}

function enlarge(){
  window.location.replace("./Zoom.html");
}

function showParameters(){
  console.log("Showing parameters");
  var paraObject = window.paraData;
  var x;
  var message = "";
  var infoScreen = document.getElementById("infoScreen");
  infoScreen.innerHTML = "";
  for(x in paraObject){
    var data = x + " : " + paraObject[x];
    infoScreen.innerHTML += data + "<br>";
  }
}

function uptimeData(){
  d3.csv("data/data.csv", function(error, data){
    data.forEach(function(d){
      d.time = +d.time;
      d.value = +d.value;
    });

        calculateSVG();
  x.domain(d3.extent(data, function(d) {return d.time; }));
  y.domain([0, d3.max(data, function(d){ return d.value;})]);

  var svg = d3.select("body").transition();
  svg.duration(750)
    .attr("d", area(data));
});
}

function setCookie() {
  console.log("setting cookie");
  var d = new Time();
  d.setTime(d.getTime() + (1 * 60 * 60 * 1000));
  var expires = "expires="+d.toUTCString();
  document.cookie = "visited=true;" + expires + ";path=/";
}

function getCookie() {
  var name = "visited=";
```

```
  var ca = document.cookie.split(';');
  for(var i = 0; i < ca.length; i++) {
    var c = ca[i];
    while (c.charAt(0) == ' ') {
      c = c.substring(1);
    }
    if (c.indexOf(name) == 0) {
      return c.substring(name.length, c.length);
    }
  }
  return "";
}

function checkCookie() {
  var visit = getCookie("visited");
  console.log("checking cookie");
  if (visit !== "") {
  } else {
 //   if (visit !== "" && visit !== null) {
      alert("Disclaimer:  Het Uroflow Dashboard is een diagnostisch hulpmiddel en nog in ontwikkeling. Geen rechten kunnen ontleend worden aan de resultaten en/of berekeningen die door het Uroflow Dashboard gepresenteerd worden.");
      console.log("cookie : No previous visit");
      setCookie();
   // }
  }
}
```

CLASSIFICATIONENGINE.JS

The classification engine is the main part that is directing the execution of the classification algorithm, ensuring all modules are imported and executed in the correct order and are provided with all their necessary data. The classification engine also forwards the data to the dashboard and D3.

```
import getParam from './getparameters.js';
import limitValues from './thresholds.js';
import interruptV2 from './interruptV2.js';
import Peaks from './findpeaks.js';
import ShapeDescriptor from './ShapeDescriptor.js';

var thresholds = 0;
var param = 0;
var interruptions = 0;

if(window.attachEvent) {
		console.log("Read CSV is running..");
   window.attachEvent('onload', runScript);
} else {
   if(window.onload) {
      var curronload = window.onload;
      var newonload = function(evt) {
         curronload(evt);
         getData(evt);
      };
      window.onload = newonload;
   } else {
      window.onload = runScript;
   }
}

function runScript(){
		console.log("--Classification engine is running--");
		var parameters = new getParam(window.data);
		parameters.spliceData();
		parameters.cleanValueArray();
		parameters.constructParameters();
		var thresholds = new limitValues(parameters.Qmax);
		var interrupts = new interruptV2(parameters.timestep, parameters.values);
		interrupts.findInterrupt(thresholds.fTh, thresholds.vTh);
		var interruptions = parameters.convertToTime(interrupts.finalArray);
		var pks = new Peaks(thresholds.peakTh, parameters.values, parameters.timing, parameters.Qmax);
		var peaks = pks.findPeaks(interrupts.sendFirstInterrupt());
		var fluctData = parameters.convertToTime(peaks);
		var paraObject = parameters.exportParameters();
		var shapedescriptor = new ShapeDescriptor(interrupts.interPresent(), pks.flucPresent(), parameters.Qr,
parameters.DTAT, parameters.Vtot, parameters.TQ, parameters.timing, parameters.Qmax[0]);
		var shapeDesc = shapedescriptor.createDesc();


		//Send data
		window.infoInterrupt =
convertToInfo(["interrupt",parameters.convertToTime(interrupts.finalIntervalsArray)]);
		window.infoInterDet = convertToInfo(["interdet",
parameters.convertToTime(interrupts.finalIntervalsArray), interrupts.volumeFound, thresholds.fTh,
thresholds.vTh]);
		window.dataInterrupt = interruptions;
		window.infoPeak = convertToInfo(["peak", parameters.convertToTime(peaks)]);
```

```
		window.infoPeakDet = convertToInfo(["peakdet", parameters.convertToTime(peaks),
pks.prominenceArray, parameters.convertToTime(pks.maxInterval), thresholds.peakTh]);
		window.dataPeak = fluctData;
		window.paraData = paraObject;
		window.shapeDesc = shapeDesc;
		window.algoData = "control";
}

function convertToInfo(_data){
		var content = "";
		if(_data[0] == "interrupt"){
				var interruptsFound = Math.floor(_data[1].length/2);
				var info = interruptsFound + " onderbreking(en) gevonden. <br>";
				var message = info;
				for (var i = 0; i < _data[1].length-1; i = i+2) {
						content = (i/2 + 1) + ": van " + _data[1][i] + "s tot " + _data[1][i+1] + "s. <br>";
						message = message + content;
				}
				return message;
		}
		else if(_data[0] == "interdet"){
				var interruptsFound = Math.floor(_data[1].length/2);
				var info = interruptsFound + " onderbreking(en) gevonden. <br>";
				var message = info;
				for (var i = 0; i < _data[1].length-1; i = i+2) {
						content = (i/2 + 1) + ": van " + _data[1][i] + "s tot " + _data[1][i+1] + "s. Volume na
interruptie: " + _data[2][i/2] + "ml.<br>";
						message = message + content;
				}
				message = message + "<br> De gebruikte limiet waarden zijn: <br> Flow-limiet: " + _data[3] +
" ml/s. <br> " + "Volume-limiet: " + _data[4] + " ml. <br>";
				message = message + "<br> Het volume na een interruptie wordt berekend tot de volgende
interruptie, of tot het einde van de grafiek.";
				return message;
		}
		else if(_data[0] == "peak"){
				var peaksFound = _data[1].length;
				var info = " fluctuatie(s) gevonden. <br>";
				var message = peaksFound + info;
				for (var i = 0; i < _data[1].length; i++) {
						content = i+1 + ": Fluctuatie op: " + _data[1][i] + "s.<br>";
						message = message + content;
				}
				return message;
		}
		else if(_data[0] == "peakdet"){
				var peaksFound = _data[1].length;
				var info = " fluctuatie(s) gevonden. <br>";
				var interval = "";
				if(_data[3][1] == null){
						interval = "Geevalueerd interval: " + _data[3][0] + "s tot signaaleinde. <br>";
				}
				else{
						interval = "Geevalueerd interval: " + _data[3][0] + "s tot " + _data[3][1] + "s. <br>";

				}
				var message = peaksFound + info + interval;
				for (var i = 0; i < _data[1].length; i++) {
						content = i+1 + ": Fluctuatie op: " + _data[1][i] + "s met een prominence van : " +
Math.round(_data[2][i][1]) + " ml.<br>";
						message = message + content;
				}
```

```
			message = message + "<br>De gebruikte limiet waarde voor prominence is 20% van Qmax: " 
+ round2(_data[4]) + " ml. <br>";
			message = message + "<br>Voor meer informatie over prominence, <a 
href=\"https://www.mathworks.com/help/signal/ug/prominence.html\" target=\"_blank\"> klik hier.</a>";
			return message;
		}
}

function round2(_var){
		_var = (Math.round(_var*1000)/1000);
		return _var;
}
```

GETPARAMETERS.JS

In this section of the appendix, the script for retrieving the parameters from the data can be found.

```
export default class getParam{
	constructor(data){
		//DATA
		this._data = data;

		//DATA ARRAYS
		this.timing = [0];
		this.values = [0];

		//CLASSIC PARAMETERS AND MATHEMATICAL VARIABLES
		this.timestep = 0;
		this.Ttot = 0;
		this.Vtot = 0;
		this.Qmax = [0];
		this.QmaxTime = 0;
		this.Qmean = 0;

		//AD PARAMETERS
		this.Qr = 0;
		this.AT = 0;
		this.DT = 0;
		this.DTAT = 0;
		this.TQ = 0;
		this.Qmm = 0;
	}

	spliceData(){ //seperates the D3 data into 2 index matching arrays
		var dataset = this._data;
		for (var i = 0; i < dataset.length; i++) {
			this.timing.push(dataset[i].date);
			this.values.push(dataset[i].value);
		}
	}

	cleanValueArray(){ //Removes all 0 and null values from the end of the array, leaving 0 that could be
interrupts
		var values = this.values;
		var control = values;
		for (var i = values.length - 1; i >= 0; i--) {//loop back to front
			if(values[i] == 0 || values[i] == null){	//Check for 0 or null
				values.pop(i);				//Pop the element
			}
			else{			//If we encounter a non 0 or non null element
				break;	//Stop
			}
		}
		this.values = values;
	//Overwrite orignal array with cleaned array
		var timing = this.timing;
		timing = timing.slice(0,values.length);			//Slice the timing array to match
the value array and to match them up
		this.timing = timing;
	//Overwrite with updated array
	}

	updateValueArray(){
		var arb = this.arb;
```

```
			var cutto = arb[arb.length-2];
			console.log("ARB CHECK");
			console.log(arb);
			this.values = this.values.slice(0,cutto);
			this.timing = this.timing.slice(0,cutto);
			var arbremoved = arb.pop();
			var newarb = arb;
			this.arb = newarb;
			console.log(this.values);
		}
		constructParameters(){
			//Classic parameters
			this.Ttot = Math.round(this.timing[this.timing.length -1] - this.timing[0]);		//Get the total
time, round to nearest integer
			this.timestep = this.Ttot/this.timing.length;
					//Calculate avarage timestep between measurments
			for (var i = 0; i < this.values.length; i++) {
					//Get all values of volume
				this.Vtot = this.Vtot + this.values[i] * this.timestep;
				//Voided volume is value per measurement * duration (timestep)
			}
			this.Qmax = [Math.max(...this.values), this.values.indexOf(Math.max(...this.values))];
		//Find the max value in values and its corresponding index
			this.QmaxTime = this.timing[this.Qmax[1]];
			//Find the time in seconds to the corresponding index of Qmax
			var tempQmean = 0;
					//Temp Qmean to store data in
			for (var i = 0; i < this.values.length; i++) {
							//loop through values
				tempQmean = tempQmean + this.values[i];
				//Add all values to Qmean (creating SUM)
			}
			this.Qmean = tempQmean/this.values.length;
			//Divide Qmean by amount of measurements (CREATING MEAN)
			console.log("Voided volume: " + this.Vtot + ". Calculated over signal length of " + this.Ttot + "
seconds, with timestep " + this.timestep);
			console.log("Qmax: " + this.Qmax + " ml/s at " + this.QmaxTime + " seconds");
			console.log("Qmean: " + this.Qmean + " ml/s over signal lenght of " + this.Ttot + " seconds");
			// A/D parameters
			this.Qr = this.Qmean/this.Qmax[0]; //Divide Qmean by the value of Qmax
			this.AT = this.timing[this.Qmax[1]] - this.timing[0];
			this.DT = this.timing[this.timing.length-1] - this.timing[this.Qmax[1]];
			this.DTAT = this.DT/this.AT;
			this.TQ = this.Ttot/this.Qmax[0];
			this.Qmm = this.Qmean/this.Qmax[0];
			window.timestep = this.timestep;
		}
		convertToTime(_data){
			var tempTimeArray = [];
			var data = _data;
			for (var i = 0; i < data.length; i++) {
				var tempTime = this.timing[data[i]];
				tempTimeArray.push(tempTime);
			}
			return tempTimeArray;
		}
}
```

INTERRUPTV2.JS

```
export default class interruptV2{
	constructor(_timestep, _values){
		this.noFlowArrayIntervals = new Array();
		this.noFlowArray = new Array();
		this.timestep = _timestep;
		this.values = _values;
		this.finalArray = new Array();
		console.log("Using timestep: " + this.timestep + " \nto apply vTh.")
		this.finalIntervalsArray = new Array();
		this.volumeFound = new Array();
	}

	findInterrupt(_fTh, _vTh){
		var values = this.values;
		var fTh = _fTh;
		var vTh = _vTh;
		var tempArray = new Array();
		var flowStarted = false;
		for (var i = 0; i < values.length; i++) {
			if(!flowStarted && values[i] > _fTh){
				flowStarted = true;
			}
			if(flowStarted){
				if(values[i] < _fTh){
					tempArray.push(i);
				}
			}
		}
		this.noFlowArray = tempArray;
		var intervals = createIntervals(tempArray);
		this.noFlowArrayIntervals = intervals;
		var finalIntervals = applyvTh(this.noFlowArrayIntervals, this.values, this.timestep, vTh);
		var _vFound = finalIntervals[1];
		finalIntervals = finalIntervals[0];
		this.volumeFound = _vFound;
		this.finalIntervalsArray = finalIntervals;
		this.finalArray = fillIntervals(finalIntervals);
	}

	sendFirstInterrupt(){
		if(this.finalIntervalsArray.length > 0){
			return this.finalIntervalsArray;
		}
		else{
			return values.length-1;
		}
	}

	interPresent(){
		if(this.finalArray.length > 0){
			return true;
		}
		else{
			return false;
		}
	}

}
```

```
var volumeFound = new Array();
function createIntervals(_noFlow){
	var RevNoFlow = new Array();
	var indexTracker = 0;
	var tempArray = new Array();
	for (var i = 0; i < _noFlow.length; i++) {
		var j = i + 1;
		if(j == _noFlow.length){
			j = j-1;
		}
		if(_noFlow[j] - _noFlow[i] > 1 || _noFlow[j] - _noFlow[i] == 0){

			if(indexTracker !== 0){
				indexTracker++;
			}
			var tempArray = _noFlow.slice(indexTracker, i+1);
			RevNoFlow.push(_noFlow[indexTracker], _noFlow[i]);
			indexTracker = i;
		}
	}
	//RevNoFlow = RevNoFlow.slice(0,RevNoFlow.length-2);
	return RevNoFlow;
}

function applyvTh(_noFlowIntervals, _values, _timestep, _vTh){
	var vThapplied = new Array();
	vThapplied.push(_noFlowIntervals[0]);
	for (var i = 1; i < _noFlowIntervals.length -2; i = i+2) {
		var currentCheck = [_noFlowIntervals[i], _noFlowIntervals[i+1]];
		console.log("checking: ")
		console.log(currentCheck);
		if(calculateVolume(currentCheck, _values, _timestep, _vTh)){
			vThapplied.push(currentCheck[0], currentCheck[1]);
		}
	}
	if(calculateVolume([_noFlowIntervals[_noFlowIntervals.length-1], _values.length-1], _values,
_timestep, _vTh)){
		vThapplied.push(_noFlowIntervals[_noFlowIntervals.length-1]);
	}
		if(vThapplied.length % 2 == 1){
			console.log("Dumped last value!");
			console.log("New vThapplied array : ");
			//var DumpLast = vThapplied.pop();
		}
	console.log("INTERRUPT OVERVIEW");
	console.log(vThapplied);
	return [vThapplied, volumeFound];
}


function calculateVolume(currentCheck, _values, timestep, vTh){
	var volume = 0;
	console.log(currentCheck[0] + " === " + currentCheck[1]);
	for (var i = currentCheck[0]; i < currentCheck[1]; i++) {
		volume = volume +  (_values[i] * timestep);
	}
	console.log("Volume found : "+ volume);
	if(volume < vTh){
		console.log("Data should be removed!");
		volume = 0;
		return false;
```

```
		}
		else{
			console.log("Data can stay");
			volumeFound.push(Math.round(volume));
			volume = 0;
			return true;
		}
}

function fillIntervals(finalIntervals){
		var filledInterval = new Array();
		for (var i = 0; i < finalIntervals.length -1; i = i + 2) {
			for(var j = finalIntervals[i]; j <= finalIntervals[i+1]; j++){
				filledInterval.push(j);
			}
		}
		return filledInterval;
}
```

FINDPEAKS.JS

```
export default class Peaks{
	constructor(_peakTh, _values, _timing, _Qmax){
		this.peakTh = _peakTh;
		this.values = _values;
		this.timing = _timing;
		this.Qmax = _Qmax;
		this.peakArray = new Array();
		this.fluctuations = new Array();
		this.prominenceArray = new Array();
		this.maxInterval = new Array();
	}

	findPeaks(_FirstInterrupt){
		var values = this.values;
		var allPeaks = new Array();
		var allLows = new Array();
		var closestHigherPeak = new Array();
		for (var i = 1; i < values.length-1; i++) {
			if(values[i] >= values[i+1] && values[i] >= values[i-1]){	//Go through all values
and find all peaks
				allPeaks.push(i);
									//Save to peaks
			}
		}
		for (var i = 1; i < values.length-1; i++) {
			if(values[i] <= values[i+1] && values[i] <= values[i-1]){	//Go through all values
and find all minimum
				allLows.push(i);
									//Save to lows
			}
		}
		for (var i = 0; i < allPeaks.length; i++) {
				// go through all peaks We dont have to observe first and last peak


	// as they have a mimimum of the peak next to them, with a lower prominence then signal end
			closestHigherPeak.push(findClosestHigherPeak(i, allPeaks, values));		//Go
through all peaks, and find the closest peak left and right of current peak


	//saves data as [[leftHigherpeak, currentpeak, rightHigherpeak],[leftHigherpeak, currentpeak,
rightHigherpeak], ...]
		}
		this.fluctuations = applyProminence(closestHigherPeak, allLows, values, this.peakTh,
this.timing, this.Qmax);
		this.fluctuations = applyInterval(_FirstInterrupt, this.values.length, this.fluctuations);
		this.prominenceArray = correctProminence(this.fluctuations);
		this.maxInterval = maxInterval;
		return this.fluctuations;
	}

	flucPresent(){
		if(this.fluctuations.length > 0){
			return true;
		}
		else{
			return false;
		}
	}
}
```

```
function correctProminence(_fluctuations){
	var tempArray = new Array();
	var prominenceIndex = new Array();
	for (var i = 0; i < prominenceArray.length; i++) {
		prominenceIndex.push(prominenceArray[i][0]);
	}
	for (var i = 0; i < _fluctuations.length; i++) {
		if(prominenceIndex.indexOf(_fluctuations[i]) > -1){
			tempArray.push(prominenceArray[prominenceIndex.indexOf(_fluctuations[i])]);
		}
	}
	return tempArray;
}
var maxInterval = 0;
function applyInterval(_FirstInterrupt, valuesLength, _fluctuations){
	maxInterval = [0, _FirstInterrupt[0]];
	for (var i = 1; i < _FirstInterrupt.length-2; i = i + 2) {
		if(_FirstInterrupt[i+1] - _FirstInterrupt[i] > maxInterval[1] - maxInterval[0]){
			maxInterval = [_FirstInterrupt[i], _FirstInterrupt[i+1]];
		}
	}
	if(valuesLength - _FirstInterrupt[_FirstInterrupt.length-1] > maxInterval[1] - maxInterval[0]){
		maxInterval = [_FirstInterrupt[_FirstInterrupt.length-1], valuesLength];
	}
	console.log("MAX interruption interval: " + maxInterval);
	var newFluct = new Array();
	for (var i = 0; i < _fluctuations.length; i++) {
		if (_fluctuations[i] > maxInterval[0] && _fluctuations[i] < maxInterval[1]) {
			newFluct.push(_fluctuations[i]);
		}
	}
	return newFluct ;
}

function findClosestHigherPeak(peak, _allPeaks, _values){
	var currentPeak = peak;
											//Current peak we are
looking at
	var allPeaks = _allPeaks;
										//all peaks
	var values = _values;
											//all valiues
	var leftOfPeak = allPeaks.slice(0,currentPeak);
							//Slice the array, save all peaks left of current peak
	var rightOfPeak = allPeaks.slice(currentPeak+1, allPeaks.length);
		//Same for right of the current peak
	var leftPeakIndex = 0;
	var rightPeakIndex = 0;
	for (var i = leftOfPeak.length-1; i >= 0; i--) {
						//For left, we loop backwards, to find the closest first
		if(values[leftOfPeak[i]] > values[allPeaks[currentPeak]]){
			//If we found a peak, is it higher?
			leftPeakIndex = leftOfPeak[i];
									//save the corresponding index in values of
this peak
			break;

	//stop looking for new peaks, we found closest higher
		}
	}
```

```
	for (var i = 0; i < rightOfPeak.length; i++) {
								//For right, we loop forward to find closest
		if(values[rightOfPeak[i]] > values[allPeaks[currentPeak]]){
			//Is the peak higher?
			rightPeakIndex = rightOfPeak[i];
										//Save the corresponding index in values of
this peak
			break;
		}
	}
	var peakIndices = [leftPeakIndex, allPeaks[currentPeak], rightPeakIndex];
	//compile data as [leftpeak, currentpeak, higherpeak]
	return peakIndices;
}
var prominenceArray = new Array();
function applyProminence(_closestHigherPeak, _allLows, _values, _peakTh, _timing, _Qmax){
							//Look for lows inbetween peaks, dont have to look at the whole data
	var peakInterval = _closestHigherPeak;
	var allLows = _allLows;
	console.log(allLows);
	var values = _values;
	var lowestleft = 0;
	var lowestright = 0;
	var finalPeaks = new Array();
	console.log("FINDING LOWS");
	for (var i = 0; i < peakInterval.length; i++) {
		var leftpeak = peakInterval[i][0];
		var currentpeak = peakInterval[i][1];
		var rightpeak = peakInterval[i][2];
		var foundLowsleft = new Array();
		var foundLowsRight = new Array();
		if(leftpeak !== 0){
	//			console.log("Looking for left between: " + leftpeak + " - " + currentpeak);
			for (var j = leftpeak; j <= currentpeak; j++) {
				if(allLows.indexOf(j) > -1){
					foundLowsleft.push(j);
				}
			}
		}
	//	console.log(foundLowsleft);
		if(rightpeak !== 0){
	//			console.log("Looking for right between: " + currentpeak + " - " + rightpeak);
			for (var k = currentpeak; k <= rightpeak; k++) {
				if(allLows.indexOf(k) > -1){
					foundLowsRight.push(k);
				}
			}
		}
		var minLeft = convertToValues(foundLowsleft, values);
		var minRight = convertToValues(foundLowsRight, values);
		minLeft = Math.min(...minLeft);
		minRight = Math.min(...minRight);
		if(minLeft == "Infinity") minLeft = 0;
		if(minRight == "Infinity") minRight = 0;
	//	console.log("Calculating highest low");
	//	console.log(minLeft + " --- " + minRight);
		var prominenceRef = Math.max(minLeft, minRight);
		var prominence = values[currentpeak] - prominenceRef;
		//console.log("Peak value = " + values[currentpeak] + ". prominenceRef = " + prominenceRef
+ ". prominence = " + prominence + ". Occurs at : " + _timing[currentpeak]);
		if(values[currentpeak] - prominenceRef > _peakTh){
			finalPeaks.push(currentpeak);
```

```
				prominenceArray.push([currentpeak, prominence]);
			}
		}

		console.log("---PROMINENCE---");
		console.log(prominenceArray);


		var finalPeaksTimeCorrected = new Array();
		if(finalPeaks.length > 1){
			for (var i = 0; i < finalPeaks.length -1; i++) {
				if(_timing[finalPeaks[i+1]] - _timing[finalPeaks[i]] > 1){
					finalPeaksTimeCorrected.push(finalPeaks[i]);
				}
			}
			//FIX for last value
			if(_timing[finalPeaks[finalPeaks.length-1]] - _timing[finalPeaks[finalPeaks.length-2]] > -1){
				finalPeaksTimeCorrected.push(finalPeaks[finalPeaks.length-1]);
			}
		}
		else {
			finalPeaksTimeCorrected = finalPeaks;
		}
		console.log("REMOVING QMAX");
		for (var i = 0; i < finalPeaksTimeCorrected.length; i++) {
			console.log(_timing[finalPeaksTimeCorrected[i]] - _timing[_Qmax[1]]);
			if(_timing[finalPeaksTimeCorrected[i]] - _timing[_Qmax[1]] < 0.5){
				if(_timing[_Qmax[1]] - _timing[finalPeaksTimeCorrected[i]] < 0.5){
					console.log("removing peak at: " + _timing[finalPeaksTimeCorrected[i]]);
					var dump = finalPeaksTimeCorrected.splice(i,1);
				}
			}
		}
		return finalPeaksTimeCorrected;
	}

	function convertToValues(_data, _values){
		var toValues = new Array();
		for (var i = 0; i < _data.length; i++) {
			toValues.push(_values[_data[i]]);
		}
		return toValues;
	}
```

SHAPEDESCRIPTION.JS

```
export default class ShapeDescriptor{
	constructor(_int, _fluct, _pQr, _pDTAT, _pVTot, _pTQ, _timing, _Qmax){
		this.int = _int;
		this.fluct = _fluct;
		this.pQr = _pQr;
		this.pDTAT = _pDTAT;
		this.pVTot = _pVTot;
		this.pTQ = _pTQ;
		this.timing = _timing;
		this.Qmax = _Qmax;
		this.Qref = 0;
		this.dev = 0;
	}
	calcMaxRef(){ //Uses Woman as reference, as gender data is not yet implemented.
		if(this.pVTot > 600){
			this.pVTot = 600;
		}
		var Qref = 6+Math.exp(0.511+0.505*Math.log(this.pVTot));
		var sd = 0.005*this.pVTot + 2;
		Qref = Qref - 6;
		var dev = 4+(6/500*(this.pVTot-100));

		this.Qref = Qref;
		this.dev = dev;
	}

	createDesc(){
		//calcMaxRef();
		if(this.pVTot > 600){
			this.pVTot = 600;
		}
		var Qref = 6+Math.exp(0.511+0.505*Math.log(this.pVTot));
		var sd = 0.005*this.pVTot + 2;
		Qref = Qref - 6;
		var dev = 4+(6/500*(this.pVTot-100));

		this.Qref = Qref;
		this.dev = dev;

		var message = "De grafiek is";
		var bellShapeCrit = "Klokvorming bepaald door :<br>xh ";
		if(this.int){
			message = message + " onderbroken,";
		}
		if(this.fluct){
			message = message + " fluctuerend,";
		}
		if(this.pQr > 0.63 && this.pDTAT > 0.85 && this.pDTAT < 2.33){
			message = message + " klokvormig,";
			bellShapeCrit += "Qmean > 0.63, DTAT tussen 1.85 en 3.13.";
		}
		else if(this.pQr > 0.63 && this.pTQ < 1.28){
			message = message + " klokvormig,";
			bellShapeCrit += "Qmean > 0.63, ratio signaallengte/Qmax < 1.28.";
		}
		else if(this.pTQ < 1.28 && this.pDTAT > 0.85 && this.pDTAT < 2.33){
			message = message + " klokvormig,";
			bellShapeCrit += "ratio signaallengte/Qmax < 1.28, DTAT tussen 1.85 en 3.13.";
		}
		else{
```

```
                bellShapeCrit = "";
        }

        if(this.Qmax > this.Qref + this.dev){
                message =  message + " met een hoog maximum.";
        }
        else if(this.Qmax < this.Qref - this.dev){
                message =  message + " met een laag maximum.";
        }
        else{
                message = message + " met een normaal maximum.";
        }

        message += "<br><br>" + bellShapeCrit;
        console.log(message);
        return message;
    }


}
```

THRESHOLDS.JS

```
export default class limitValues{
		constructor(_Qmax){
				this.fTh = 0.5;							//Flow treshold 0.5ml/s
				this.vTh = 5;							//Volume treshold for interruption
evaluation 5ml/s
				this.peakTh = _Qmax[0]*0.2; //Peak trheshold 20% of Qmax
				console.log("Using a peak trheshold of: " + this.peakTh + "(" + _Qmax[0] + " x 0.2");
		}
}
```

SOURCE CODE FOR THE DASHBOARD

INDEX.HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="./css/graphCss.css">
<link rel="stylesheet" type="text/css" href="./css/indexCss.css">

<script src="https://d3js.org/d3.v4.min.js"></script>
<!--<script type="text/javascript" src="graph.js"></script>-->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
</head>
<body class="body-bg">
<div class="container-fluid">
	<div class="row">
			<div class="col banner">
				<h1 class="bannerheader">Uroflow Dashboard</h1>
			</div>
			<div class="col banner">
			<p class="lead verticalAlign">Work in Progress (*functies* niet beschikbaar)</p>
			</div>
			<div class="col-2 banner">
				<div class="logo">
				</div>
			</div>
	</div>
	<div class="row">
		<div class="col-md-6">
			<div class="">
				<svg width="900" height="400"></svg>
			</div>
		</div>
		<div class="col-md-6">
			<div class="row element-bg" style="height: 400px">
				<div class="col-md-6 lineRightXL">
					<div>
						<h4 >Functie evaluatie:</h4>
						<div id="infoScreen">
						</div>
					</div>
				</div>
				<div class="col-md-6">
					<form method="post" enctype="multipart/form-data">
						<div class="btn-group" role="group">
			<input type="file" accept=".csv" name="files[]" class="btn btn-primary " text="">
  <input class="btn btn-primary " type="submit" value="Update Grafiek" name="submit" id="submit_re">
						</div>
				</form>
					<script src="./js/upload.js"></script>
						<div class="btn-group" role="group">
```

```
<button class="btn btn-primary btngrid" onclick="showInterrupt()">Interrupties</button>
        <button class="btn btn-primary btngrid" onclick="showPeaks()">Fluctuaties</button>
                                </div>
                                <div class="btn-group" role="group">
        <button class="btn btn-primary btngrid" onclick="showShape()">Vorm</button>
        <button class="btn btn-primary btngrid" onclick="showParameters()">Parameters</button>
                                </div>
                                <div class="btn-group" role="group">
        <button class="btn btn-primary btngrid" onclick="enlarge()">Uitvergroten</button>
        <button class="btn btn-primary btngrid" onclick="notAvail()">*Plateau*</button>
                                </div>
                                <div class="btn-group" role="group">
        <button class="btn btn-primary btngrid" onclick="notAvail()">*Symmetrie*</button>
        <button class="btn btn-primary btngrid" onclick="">*TEMP BUTTON*</button>
                                </div>
                        </div>
                </div>
        </div>
        <div class="element-bg col-md-12">
        <div class="row">
                <div class="col-md-3 lineRight">
                        <div class="">
                                <h4>Algorithme evaluatie:</h4>
                                <hr>
<p>Het algorithme is zonder problemen uitgevoerd, de schaal van de grafiek is niet aangepast.
                                Geen benoemenswaardige bijzonderheden gevonden.</p>
                        </div>
                </div>
                <div class="col-md-3 lineRight">
                        <div class="">
                                <h4>Vorm evaluatie:</h4>
                                <hr>
                                <p id="shapeEval"></p>
                        </div>
                </div>
                <div class="col-md-3 lineRight">
                        <div class="">
                                <h4>Onderbrekingen:</h4>
                                <hr>
                                <p id="interEval"></p>
                        </div>
                </div>
                <div class="col-md-3">
                        <div class="">
                                <h4>Fluctuaties:</h4>
                                <hr>
                                <p id="flucEval"></p>
                        </div>
                </div>
        </div>
        </div>
    </div>
</div>

</body>
<script type="text/javascript" src="./graph.js"></script>
<script type="module" src="./algorithme/classificationEngine.js"></script>
</html>
```

ZOOM.HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="./css/graphCss.css">
<link rel="stylesheet" type="text/css" href="./css/indexCss.css">

<script src="https://d3js.org/d3.v4.min.js"></script>
<!--<script type="text/javascript" src="graph.js"></script>-->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
</head>
<body class="body-bg">
<div class="container-fluid">
	<div class="row">
			<div class="col banner">
				<h1 class="bannerheader">Uroflow Dashboard</h1>
			</div>
			<div class="col-2 banner">
				<div class="logo">
				</div>
			</div>
	</div>
	<div class="row">
		<div class="col-12 justify-content-center">
			<svg width="1700" height="800"> </svg>
		</div>
		<div class="instructions">
			<p> Druk op "I" voor interrupties. <br>
				Druk op "P" voor fluctuaties.  <br>
				Druk op "V" voor globale vorm. <br>
				Klik om terug te gaan.
			</p>
		</div>
	</div>
</div>
</body>
<script type="text/javascript" src="./graph.js"></script>
<script type="module" src="./algorithme/readCSV.js"></script>
<script type="text/javascript">
	window.addEventListener("click", function (){
		window.location.replace("./bootstrap.html");
	});

	document.onkeypress = function(evt) {
   evt = evt || window.event;
   var charCode = evt.keyCode || evt.which;
   var charStr = String.fromCharCode(charCode);
   console.log(charStr);
   if(charStr == "i" || charStr == "I"){
	console.log("Showing interrupts")
```

```
            showInterruptZoom();
  }
  if(charStr == "p" || charStr == "P"){
            console.log("Showing peaks")
            showPeaksZoom();
  }
  if(charStr == "v" || charStr == "V"){
            console.log("Showing peaks")
            showShapeZoom();
  }
            };
</script>
</html>
```

Appendix I: Ethical evaluation.

Table of contents

Introduction

In this report we hope to gain further insights in possible ethical risk and moral problems that may present themselves in either the creation, testing and deployment of the Uroflow Dashboard project at the UMC Utrecht. Furthermore, possible solutions will be introduced, may such problems arise during this research.

UROFLOW DASHBOARD PROJECT

The Uroflow dashboard project is a joint collaboration between the University of Twente and the University Medical Center Utrecht, UMC for short. In this project, a dashboard is developed to aid urologist in finding, identifying and diagnosing lower urinary tract symptoms.

To find lower urinary tract symptoms, patients are required to void their bladder in a specialized toilet, which will measure urine flow as opposed to time. The resulting dataset can easily be turned into a line graph, with time in seconds at the X-axis, and voiding in ml/s on the Y-axis. This is a first visualization stage to easily identify unnatural high peaks and/or interruptions in the voiding routine of the patients.

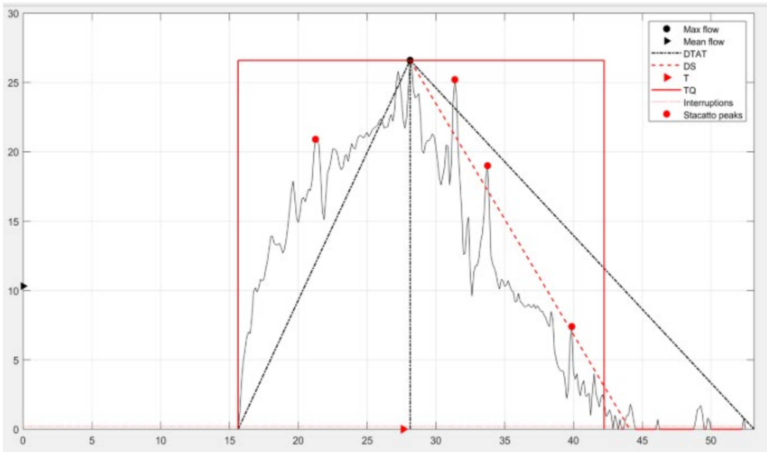


Figure 56: Output of the LUTS identifying algorithm (van Haaren, 2019)

The dataset containing the persons voiding routine will also be passed through a state-of-the-art algorithm developed at the UMC, which will further identify characteristics such as the maximum voided volume per second, further referred to as Qmax, the symmetry of the voiding graph, the location of Qmax as opposed to the rest of the graph, skewness, acceleration and deceleration time of the void and a few others.

All of these findings will be presented in a dashboard, created by the University of Twente. In this dashboard, both explanatory and explorative data will be available. A snapshot of the dashboard can be found in figure 2.

For explorative data a graph is presented where peaks and interruptions are highlighted, and all characteristic data will be listed in a side window, as well as the locations and durations of other points of interest, such as peaks, interruptions and Qmax. Furthermore, smaller visualizations of variables like acceleration and deceleration time will be present.

For explanatory data all the findings of the algorithm will be explained and argued, for further evaluation of the attending urologist to either concur or to reject the findings. Furthermore, a select amount of variables from the graph, such as skewness, will be explained in words. It might say that the graph is skewed to the right for example.

Another item present in the dashboard, is a slider that grants the user the ability to alter the algorithm, as the algorithm is yet in development and needs more fine tuning. This slider will effect limit values on which the algorithm decides if something is either normal or a symptom and should be selected for further evaluation.

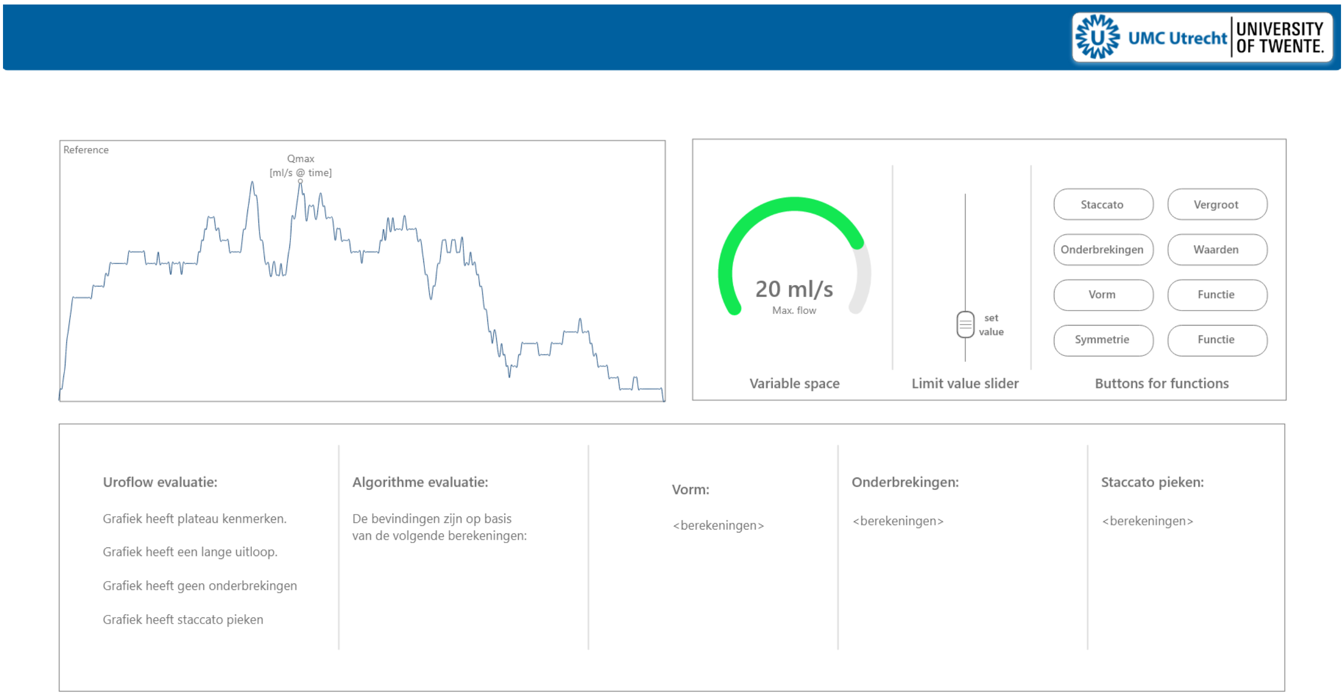


Figure 57: A prototype snapshot of the dashboard (Adobe XD)

METHOD

To expose possible ethical risk and moral problems, a 7-step system will be implemented. This system is adapted from Ethics in Technology Toolkit (Vallor, Mckenna, & Mckenna, 2018). To further clarify this 7-step system, a small introduction will be given for every tool, introducing what will be discussed, what the tools purpose is, and how the tool will apply to our specific case.

The 7 tools found in the Ethics for Technology toolkit are:

- Ethical risk sweeping.
- Ethical pre- and post-mortems
- Expansion of the ethical circle
- Case-based analysis
- The ethical benefits of creative work
- Analyzing possible abuse of the system
- Ethical feedback and iteration.

Ethical Risk Sweeping

Ethical risk can be found by thinking of scenarios and their outcomes. The risk we take in those scenarios should in no way cause (considerable) harm to either the image of our own project/company, human beings, other entities or businesses, institutions or the environment. Scenarios that will lead to acute moral controversy are also included in this section.

The tool will be applied as earlier described above. A certain scenario will be taken into mind, and a multitude of possible outcomes will be considered, and certain steps will be presented to avoid scenarios that lead to controversy. In the case that avoidance of negative outcomes is not possible, solutions will be brought forward to confine the negative effects to a minimum.

Ethical pre- and post-mortems

Ethical pre- and post-mortems can be defined by looking at earlier research we have conducted or other projects we have completed. Based on the experience we have gained from previous work, a number of solutions to issues that have been found in the past can be implemented, preventing those issues from reoccurring.

Furthermore, a look will be taken into the current state of the project to try to identify possible ethical or systematical problems that might lead to catastrophic failure, whether it be by a single event or a cascade of smaller minor deficits.

Expansion of the ethical circle

Expanding the ethical circle is a tool to get more and deeper insights in our casus. Other people might have other insights we have overlooked. One way to achieve this is to get opinions from experts on the grounds of the dashboard project, or to introduce the opinion of stakeholders. This way, we can cover certain situations more effectively, and most important, more thoroughly.

Case-based analysis

The aim of this tool is to look similar work that might be a good reflection of our current project, and do research on ethical problems that may have occurred during the development and/or implementation of said similar project. This way, we can avoid pitfalls others have already discovered for us.

Furthermore, analogical reasoning will be applied to our mirror case to explore coexisting risks, opportunities, solutions and mitigation strategies.

The ethical benefits of creative work

The ethical benefits of creative work can be found by conduction a select amount of sub-questions.

- Why are we doing this, and for what good ends?
- Will our clients be better off with our service?
- Has the ethical benefit of this technology remained at the center of our work and thinking?
- What are we, as creators of this project, willing to sacrifice to get this right?

All of these sub questions force us into one direction. We need to be careful not to purposely overlook some of ethical risk associated by our project in order to gain short term success, as opposed to doing a thorough analysis to create a future proof product with long term benefits.

Analyzing possible abuse of the system

Analyzing possible abuse of the system is of great importance. Our product should in no way be received as a means to do direct or indirect harm to others whilst it is used either as intended, or with malicious intend for self-gain.

Secondly, our product should have certain safety measures that prevent wrongful use, be it in- or unintended.

Ethical feedback and iteration

Ethical feedback and iteration are an unmeasurable valuable asset to our project, as ethical research is never finished. New insights may arise later on, or a change in society or law may require us to rethink our ethical and moral strategies, as well as the possible protocols embedded in our product to make s both conform to society and law.

Applying the toolkit.

In this section, all of the aforementioned tools will be applied to our current state of research and development of the Uroflow Dashboard project. Ethical risk sweeping, pre- and post-mortems, and analyzing possible abuse of the system will be done on a scenario basis, meaning all findings are hypothetical, but can form real threats in the near future. If a tool is found to be too deficient to apply to a single case, it will be applied over a multitude of scenarios as a whole.

SCENARIOS

Below, a select amount of possible scenarios will be presented and the ethical risk based on those scenarios will be evaluated.

Scenario 1: Patient data gets stolen.

One huge ethical concern in almost all projects concerning data and computing is privacy. In this case we presume that the Uroflow dashboard contains a data storage functionality for comparisons between earlier measurements, or even a possibility to compare data of different patients that are presenting similar symptoms.

Storing data, of course, comes with the risk of it being stolen, which can be used to blackmail certain patients, or sell personal information to advertisers. This is a possible abuse of the system by third parties and should be prevented at all cost, as this may not only hurt the individual who's data has been stolen, but also the image of the UMC as a whole, where the project is to be implemented.

One way to obviate this is to have no data storage at the dashboard end at all, while keeping the comparison functionality. The difficulty with this solutions is that it requires the attending urologist import the data every time he or she wants to inspect it. Normally, this would not be an issue, but having to import different files over and over again to make a what should be easy comparison can become very frustrating over time. This solution would be a Hail-Mary option if no other solution is available or other solutions fail to implement.

A second solution, based on earlier projects we have be completed, would be to deploy the dashboard on the UMC's intranet and restrain public access. This will mitigate the situation somewhat, but we need to be realistic of the fact that if someone intrudes the intranet, our project and all the data it contains is exposed as well. Furthermore, our data can also be stolen by a corrupt employee of the UMC, which is why we strongly recommend to encrypt all, if any, patient data and limit access to the patients own urologist.

This way, if data gets leaked, based on the information that pops up, the source of the data leak can easily be identified and dealt with accordingly. The ability to deal with these data leak sources will greatly reduce the loss of faith and damage done to the image of the UMC, as, while the UMC is still responsible, someone can be held accountable. The main gist of this solution is, if we can not prevent it, we must have someone we can held accountable if a situation occurs. This will also discourage possible perpetrators from going through with their action, knowing they can be traced. (Beu & Buckley, 2004)

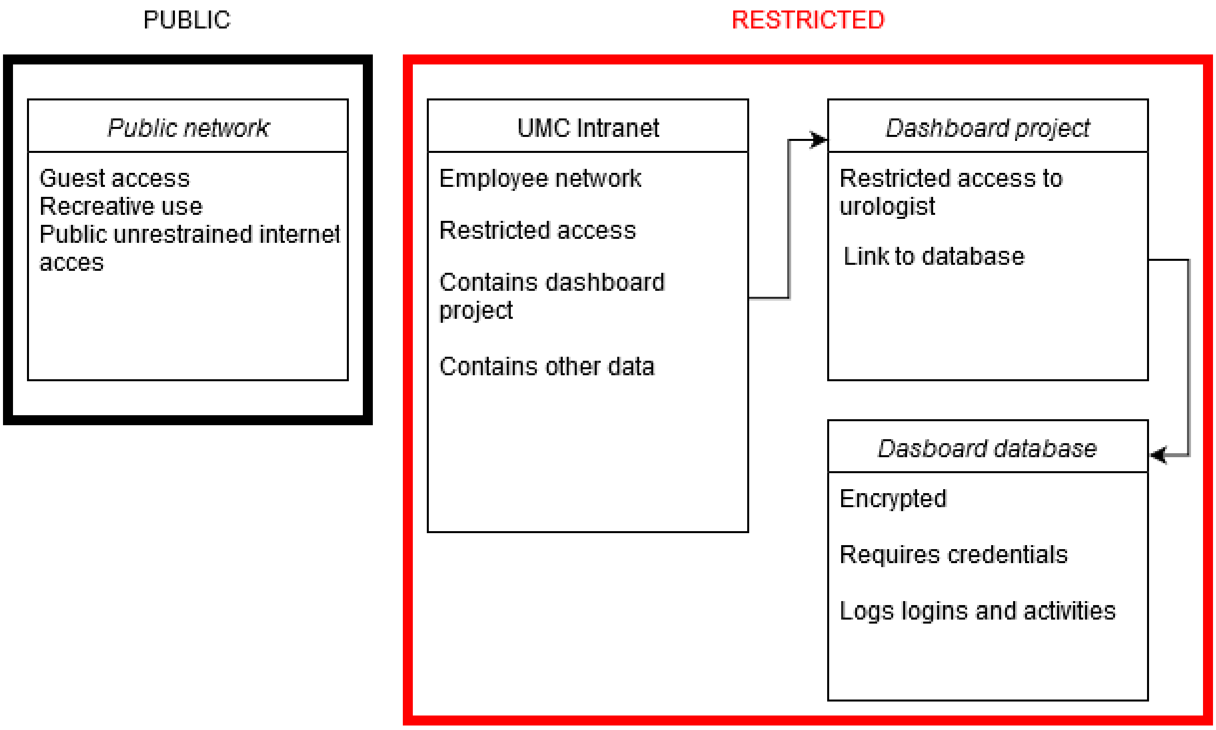


Figure 58: System diagram of the 2nd proposed solution

Scenario 2: The dashboard becomes the urologist.

One thing that we should make very clear to any potential user of this project is that the Uroflow Dashboard is an aid for diagnosing, not a tool to diagnose. As with any Clinical Information Technology (CIT), or Information Technology in general, errors can be thrown, leading to faulty calculations and/or outcomes. Becoming too dependent on the dashboard can be described as unintended use, or even abuse of the system, as it might provide a gateway for urologist to slack off at work.

Eventual errors.

We, as the creators of the Uroflow Dashboard should prevent errors from happening at any cost, and take measures to catch errors if they occur. A very clear indicator needs to be implemented to notify urologist that, when we even have the slightest amount of doubt that a faulty calculation might have happened, the resulting output might be faulty and the user should check data and values. If this check is not done, and the urologist is not skeptical about the dashboards output, and blindly follows along, this might do some serious harm to patients. This could lead back to us, as our product gave faulty information. To cover for these potential lawsuits, it is of great importance that we take a legal standing point regarding this basis. To repeat, it should be very clear that the dashboard is an aid, not a tool, to diagnose, and should in no way be leading over the attending urologist own opinion.

Ratings of physicians

Another reason why we need to be very clear the dashboard is an aid, and not a tool is not only for our own interest, but also in the interest of the image of the UMC. A study conducted in 2008 researches the rating of physicians who rely on expert opinions as opposed to those who rely on tools or CIT's. The study concludes that physicians who rely tools, rather then own or expert opinion are rated less favorable. (Probst, Shaffer, Lambdin, Arkes, & Mitchel, 2008)

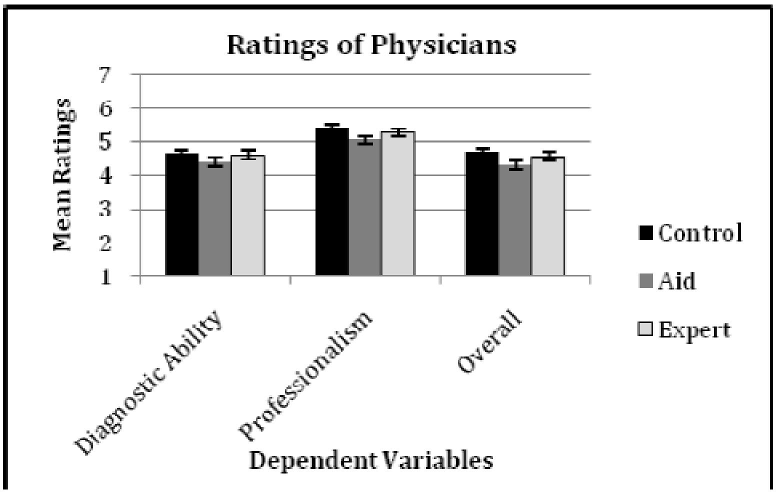


Figure 59 The physician making the unaided diagnosis was rated more positively than the physician using the computer based aid on three variables (diagnostic ability: z=-1.75, p <.05, overall satisfaction: z= -1.71, p<.05, and professionalism: z=-2.77, p<.05). The physician who consulted a human expert had ratings that were not significantly different than the physician making the unaided diagnosis.(Probst et al., 2008)

According to the findings above, it might also be a very real scenario that patients lose trust in their physician as they rely on a CIT, such as the dashboard, too much. This could be based on the patient feeling anyone could be telling them their diagnosis, rather than an expert.

Luckily, there is still a large amount of healthy skepticism amongst physicians regarding the implementation of CIT's. A study conducted in 2007 writes the following:

"On a more positive note, the healthy dose of skepticism remaining among current and potential users of new CIT may help drive both product and process improvements around CIT that can, in turn, improve the quality of care supported by the new technologies. Across focus groups, physicians' willingness to provide input about what technology features are most helpful and what capabilities they would like was striking. Including interested physicians in technology-aided process improvements and in developing decision support tools such as order sets or HHC databases will likely continue to help increase physicians' usage and acceptance of such technologies. Given that CIT implementation appears to proceed in a never-ending cycle of product enhancements, exploring and accommodating physicians' perspectives and interests could conceivably help both product developers and organizations involved in technology implementation in order to ensure that physicians' needs are best met and to maximize the opportunity to reduce physician skepticism." (McAlearney, Chisolm, Schweikhart, Medow, & Kelleher, 2007)

From this, we can conclude that urologist won't be dependent of immediately, but we still need to be aware of the fact that this can be a very real possibility the longer the tool is implemented. We should use the timespan between first implementation and familiarization, where the amount of skepticism and additional input for new/requested features is at its highest, to improve our product and remove as much of early adaptor errors as possible.

Service unavailable

A third reason we should avoid too much dependence is that there is a very real possibility that the Dashboard might become unavailable during times. This might be because of crashes, or system updates. If the urologist are too dependent on the Dashboard, this will result in them not being able to diagnose people the "old-fashioned" way. This can be very harmful to patients who need immediate care.

We as a company should still implement ways to prevent system unavailability. One way to achieve this is to implement a Common Address Redundancy Protocol (CARP). (“Common Address Redundancy Protocol - Wikipedia,” n.d.; McBride, 2003)

When CARP is implemented, multiple hosts on the UMC’s intranet will run the Uroflow Dashboard program, one being the default Master Server, on an active link. When the Master server fails, requests to the Dashboard will automatically be redirected over a passive link to a second host, or so called slave, ensuring the service stays available. To achieve this, synchronization between the master and slave servers is of great importance. This would require multiple hard copies of the database linked to the Dashboard interface.

The synchronization part could also be mitigated by storing no data on the dashboard server, as described in scenario 1, solution 1.

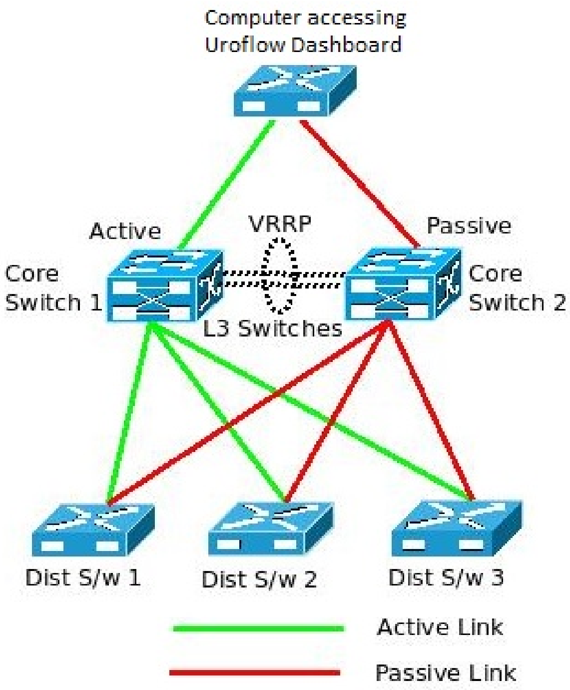


Figure 60: CARP protocol, adapted from (“Debian redundatn gateway VRRP setup with Keepalived,” n.d.)

Scenario 3: The algorithm is altered.

As mentioned before in the introduction, the classification algorithm is still in development. One way to accommodate this in the current version of the dashboard is the introduction of a limit slider (figure 2). However, this does come with certain dangers, as changing the limit slider, changes what classifies as a symptom. If the user does not realize changes have been made, whether accidental or intentional, this could have some serious consequences for the diagnosis.

A solution to this is to have the slider reset to a default value every time the dashboard is accessed, and change the graph and calculation in real time, when the slider is moved. This way, it would be hard to miss that something is changing, removing the accidental part from the equation.

Furthermore, the source code for the project should be locked up tight, ensuring no unauthorized changes can be made to the core of the project. Changes with malicious intend can have a serious impact on the system. Possible motivation for malicious changes can be:

- Copying and selling personal data without a trace
- Revenge
- Sadistic intentions
- Feeling in control

Preventing unauthorized changes can be done in a multitude of ways.

One is to have every file, or rather the entire folder, encrypted and password protected, requiring personal credentials, and limited to a select amount of people who need that level of authorization.

A second or additional layer of protection is a code change voting system. Every time a source code file is changed and committed (cleared for implementation by the creator, or person who changed the code), a copy of the new file will be send to a select amount of people. These people can be experts in their field, system administrators, or the UMC's ICT department. Only after the majority, or even all of these people, have approved of the changes to the source file, will the updated source file be implemented in the project.

CASE-BASED ANALYSIS

Minze has developed a special toilet accessory that can be fitted on any toilet and is able to take uroflow measurements. Accompanying this toilet accessory are 2 types of dashboards. The Minze Hospiflow and the Minze Homeflow. These two work closely together, as data from the Homeflow can easily be passed on to the users physicians Hospiflow.

For the sake of mirroring our current case, a more detailed report will only be given on Hospiflow. Hospiflow integrates patient details and uroflow data into one clear overview. It contains history, detailed data and uroflow graphs all in one overview.

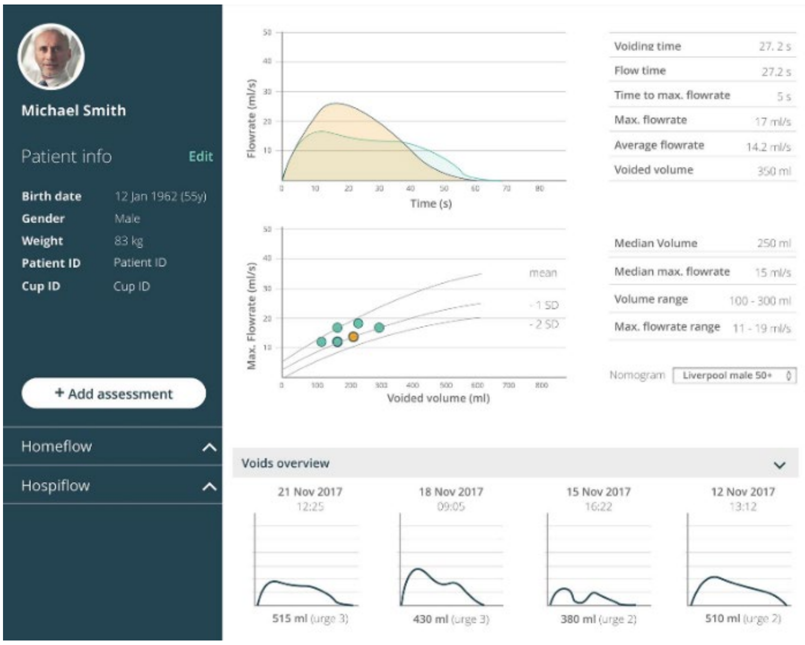


Figure 61: Minze Hospiflow (“Minze Health - Hospiflow & Homeflow - Uroflowmetry and voiding diaries,” n.d.)

As Minze has only launched their first commercialized products in May 2018 (“Minze Health, innovatieve tool voor plasproblemen - De Specialist,” n.d.), there is not a lot of information available on (ethical) problems that have arisen since its implementation.

There is, however, more detailed information about their measuring tool, regarding approvals from the Belgium Classification Organization for medical equipment. There is a small note regarding corresponding software, hinting to the interface also being approved.

No other tools implementing a dashboard and/or classification algorithm for medical use have been found. This is a warning to us that we have to be very careful with choices we make, as there are no previous products we can learn from. This will become very important later during ethical feedback and iteration, as we need to find our own risk mitigation strategies during the development and implementation of the dashboard.

THE ETHICAL BENEFITS OF CREATIVE WORK.

In this section, a more detailed look will be taken into our own motivation for creating this product. Is this something the world really needs, and has a good use for? To do this, the following sub-questions will be answered, and several points of attention will be brought to light.

Why are we doing this, and for what good ends?

The main motivation for this project is to aid urologist during a diagnosis of a patient with lower urinary tract symptoms (LUTS). Arising from this is a motivation to help human beings live a better life, by giving them better medical care. Our purpose is to make life easier for both the urologist and the patient.

Will our clients be better off with our service?

In a perfect world, based on our motivation given above, yes. Realistically speaking, there are a lot of factors we need to take into account that our software is not altered or abused to cause harm to others. To expand on this, we need to ask ourselves the question if the benefits outweigh the possible risk that harm might be done, and the size of that ham.

Based on ethical risk sweeping, we are confident to say, that is, if urologist use the dashboard as intended, and necessary safety measures are put in place, that our product can be very beneficial to both our clients and their patients. But make no mistake, this is if the dashboard is used as **intended, iterated upon,** and **measures to prevent abuse** are in place.

Has the ethical benefit of this technology remained at the center of our work and thinking?

During the development and evaluation of the Dashboard, we have tried to translate values into design requirements as much as possible. This process is described by Ibo van der Poel, in his paper "Translating values into design" (Poel, 2013).

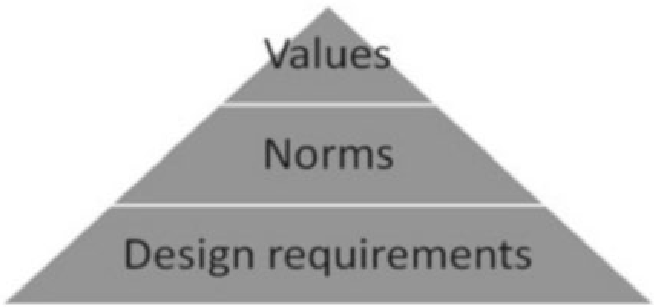


Figure 62: The three basic layers of a values hierarchy. Note that each of the layers may itself be hierarchically layered (Poel, 2013)

An example of this is the value patient well-being, from which the norms privacy, security and professional attitude arise. Based on these norms, the following design requirements are to be implemented.

Privacy

Only grant access to an urologist own patient data, the urologist will not be able to see data from other patients who are not his clients or directly connected to him by means of a session or consult. Furthermore, no data shall be stored at the dashboards end or it be severely protected and encrypted, requiring personal credentials to access this data. A permanent log-file shall be kept of any access to any file whatsoever, all by it through the dashboard interface, or direct access by means of using the database.

Security

The dashboard must include features to ensure every aiding diagnoses is right. In the slightest amount of doubt, the attending urologist is to be directly notified that there might be a possibility of faulty calculation, and should be extra careful if he or she decides to proceeds with a diagnosis based on the dashboard.

Professional attitude

The attending urologist needs to be made aware that the dashboard is just an aid, and not a tool to do a diagnosis for him or her. This notification must enforce a professional attitude from the attending to the patient, explicitly mentioning to them what the Dashboard is, how they use it, and even use the Dashboard to explain the patients symptoms to the patients.

What are we, as creators of this project, willing to sacrifice to get this right?

Based on the extent of what we are actually dealing with here, it is of great importance that we as the creators of this project ensure that everything is up to standards. The main reason for this is the fact that we can have a major impact on another human beings health, thus their lives.

On top of this, we should have a strong legal standpoint, ensuring that we, as the creators can not be at fault for mistakes urologist make based on our dashboard, with the exception of major faulty software. However, we must also emphasize here that under no circumstance, a client might feel accused by us, as the creators, for not using our software correctly and feel insulted. In this case, it would be better to take a strong stand and mediate between the patient and the attending.

EXPANDING THE ETHICAL CIRCLE

Expansion of the ethical circle is of vital importance to this project. Although a multitude of scenarios have been discussed in this research, even obvious risk may have been overlooked. This is why we suggest having regular interviews with stakeholders and other experts to find more ethical pitfalls. A few examples, but not limited to are:

Expert opinions. Have urologist from the UMC, or even from other organizations, think up a multitude of scenarios and risk tied to this project. There will be, with no doubt, duplicate synopsis, but the odds of finding an overseen risk greatly outweighs reading through all these duplicates. We need to make sure we cover all our basis.

ICT experts. Have experts in computer technology check our security and give advice based on the storage of data, type of encryption, implementing redundancy, etc.

This might seem like something that should be done in development, but as the information age progresses, new secure system arise, and older encryptions are broken or bypassed.

A prime example of the importance of a secure system is ransomware. Ransomware encrypts files and makes them unusable, until a ransom is paid, effectively keeping the files hostage.

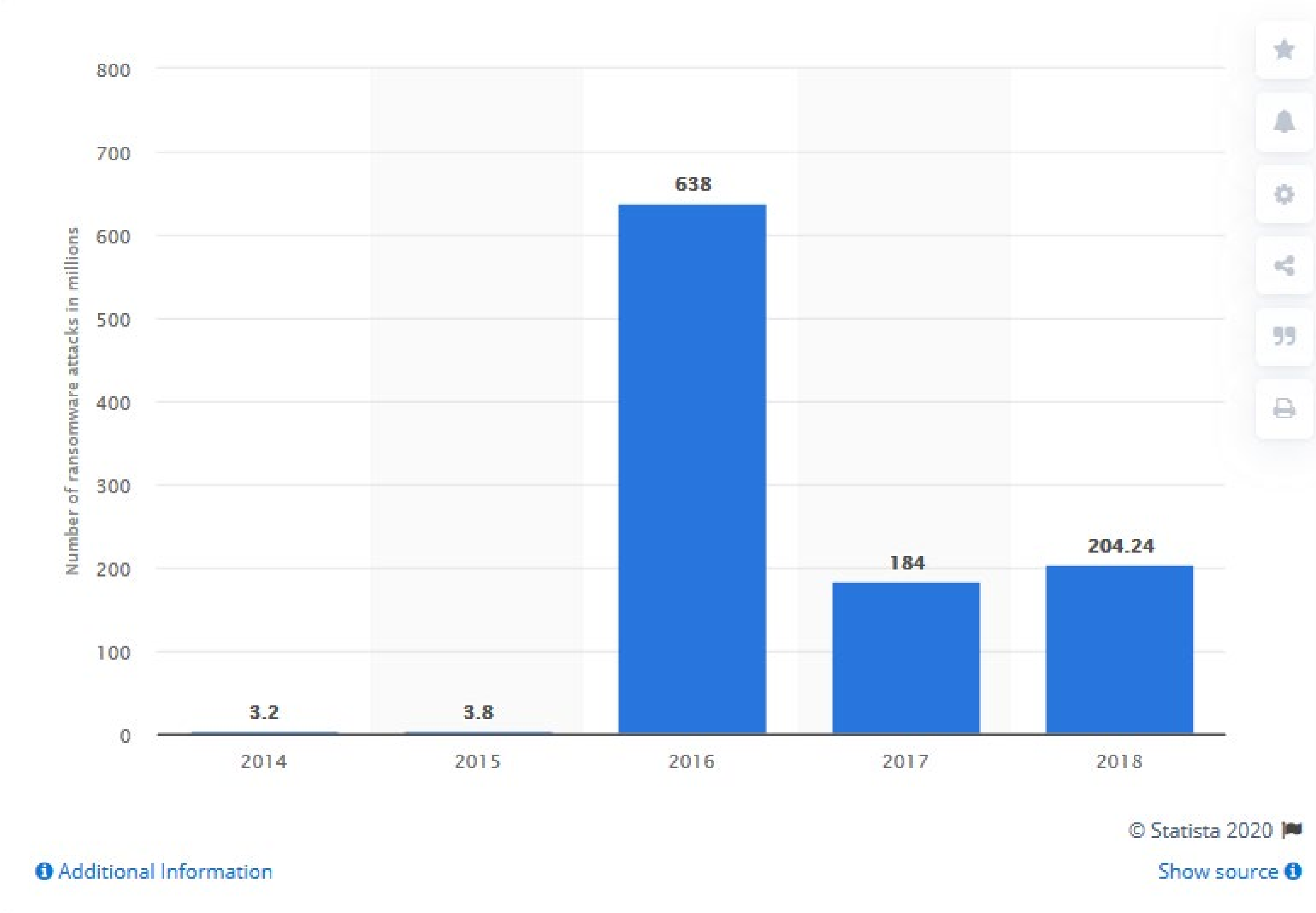


Figure 63: Amount of ransomware attacks in millions("• Number of ransomware attacks per year 2018 | Statista," n.d.)

Hospitals have often been victims of ransomware attacks, or even specially targeted. One reason for this might be a higher pressure form society to pay the, often higher than usual, fee to decrypt the files. The hospitals often give in, because during the time the files are being held hostage, no patients can be treated.

Another way to expand the ethical circle might be to ask a select amount of patients about their opinion on the use of the Dashboard as a diagnostical aid. Examples of such questions are:

- Do you feel comfortable knowing your attending uses a technological aid to make your diagnoses?
- Do you foresee any risk in using such as system as this?
- What is your view on the use of information technology in hospitals?

ETHICAL FEEDBACK AND ITERATION

It's a well know statement that ethical engineering is a never finished task, due to changing social and political climates. When a social value shifts, so does its ethical counterpart. This is why it is vital to this project to have multiple feedback moments and iterations based on said feedback moments. To add to this, CIT's are categorized under ICT, which is a very rapidly changing discipline. This is why we propose the following ethical feedback loop.

Meetings at a regular interval shall be held between the R&D team, design team and an ethics team. Here, newly arisen ethical risk shall be discussed. These new ethical risk can be a product of a changing climate in society, based on new features to be implemented, or from expansion of the ethical circle.

The ethics team is tasked with finding out what the root of the new found risk are and will forward this information to the R&D team. The R&D team will try to tackle these risk in the core of the project. If this fails, the design team will step in and try to fix the issue in the interface.

If both the R&D and the design team are unable to fix the problem, the ethics team will run this by supervisors and ask for the design speciation's to be altered, or for certain features to be left out.

On top of this, peer review sessions will be introduced with other companies in the same field, experts on the field of information security and urology, and medical personnel in general. This is an addition on expansion of the ethical circle. During these meetings the group will check the current state of the project and try to spot any ethical risk, and check if previous ethical risk are dealt with accordingly .The findings of these focus groups should be well documented and presented at the meetings with the ethical teams to find solutions.

Furthermore, we should not neglect any moral or ethical problem an employee working at the project has. If someone who is developing the product we are trying to sell, there is a guarantee a (potential) client has the same issue as well. We need to encourage these employees to stand up and voice their opinion.

All of the above mentioned should be held at least three times before a new version of the product is granted the “OK” for release.

We are confident by implementing this feedback cycle, we have the ability to find and fix any major moral and/or ethical issue, before an update gets pushed to client.

Conclusion

After careful execution of this research, we are confident to say that the project is viable in terms of ethical risk and moral issues. All major risks found during the research of this project have been looked into and a fitting solutions has been presented.

We must make sure that patient can not be harmed by intended or unintended use or abuse of our system. Safety measures must be implemented completely and tested thoroughly before the complete systems is implemented. This applies to both measures at the interface itself, as well as measures to protect the source code of the project.

We must make sure that patient privacy is taken very seriously. Not a single person who has no real need to check a patients files must have authorization to those files taken away immediately. Measures must be put in place to prevent data leaks, or a breach of the database, resulting in the patients details going public.

We must have a strong legal standpoint on the grounds that we are in no way responsible for a wrong diagnoses based on our aid. After all, the attending has full responsibility over the choices he or she made. Nevertheless, we must do everything in our power to prevent wrong diagnoses, informing attending of possible faulty calculation.

We must take (legal) responsibility if a data leak happens, and do everything to find the perpetrator to protect both our own and the UMC's image.

We must make attending urologist realize that Uroflow Dashboard is an aid and not a tool for slacking, forcing them to have a professional attitude towards our product, as well as the patient.

We hope to have informed the readers of this report of major risks and possible ethical issues that may arise during the development and/or implementation of "Uroflow Dashboard", and that those risk will be acted upon accordingly, and proposed solutions to be implemented as the developers see fit.

Bibliography

Number of ransomware attacks per year 2018 | Statista. (n.d.). Retrieved January 6, 2020, from https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/

Beu, D. S., & Buckley, M. R. (2004). Using accountability to create a more ethical climate. *Human Resource Management Review*, *14*(1), 67–83. https://doi.org/10.1016/j.hrmr.2004.02.004

Common Address Redundancy Protocol - Wikipedia. (n.d.). Retrieved January 6, 2020, from https://en.wikipedia.org/wiki/Common_Address_Redundancy_Protocol

Debian redundatn gateway VRRP setup with Keepalived. (n.d.). Retrieved January 6, 2020, from http://blog.alexolivan.com/the-return-of-the-linux-router-from-pfsense-to-debian-part-4-from-carp-to-vrrp/

McAlearney, A. S., Chisolm, D. J., Schweikhart, S., Medow, M. A., & Kelleher, K. (2007). The story behind the story: Physician skepticism about relying on clinical information technologies to reduce medical errors. *International Journal of Medical Informatics*, *76*(11–12), 836–842. https://doi.org/10.1016/j.ijmedinf.2006.09.021

McBride, R. (2003). *"CARP."* Mailing list ARChives. Retrieved from http://marc.info/?l=openbsd-misc&m=106642790513590

Minze Health, innovatieve tool voor plasproblemen - De Specialist. (n.d.). Retrieved January 6, 2020, from https://www.despecialist.eu/nl/nieuws/beroepsnieuws/minze-health-innovatieve-tool-voor-plasproblemen.html

Minze Health - Hospiflow & Homeflow - Uroflowmetry and voiding diaries. (n.d.). Retrieved November 1, 2019, from https://minzehealth.com/

Poel, I. van de. (2013). Translating Values Into Design, *15*(January 2013). https://doi.org/10.1007/978-94-007-7762-0

Probst, C. A., Shaffer, V. A., Lambdin, C., Arkes, H. R., & Mitchel, A. (2008). Ratings of Physicians Relying on Experts Versus Physicians Relying on Decision Aids, 91–92.

Vallor, S., Mckenna, R., & Mckenna, D. (2018). Ethics in Technology Practice - Toolkit. Retrieved from https://www.scu.edu/media/ethics-center/technology-ethics/Ethics-Toolkit.pdf

van Haaren, J. (2019). Automatic component based analysis of uroflowmetry curves in women.

Appendix J: Uroflow Dashboard Images

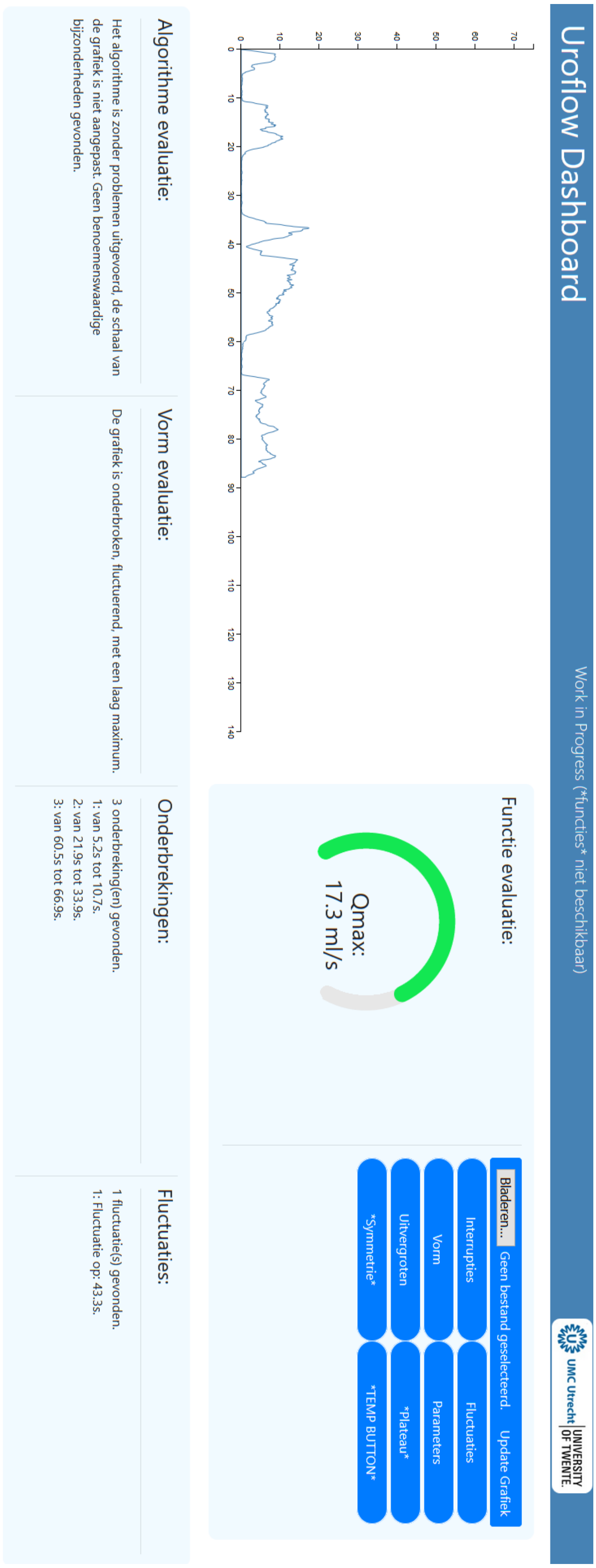


Figure 64: Uroflow Dashboard home screen

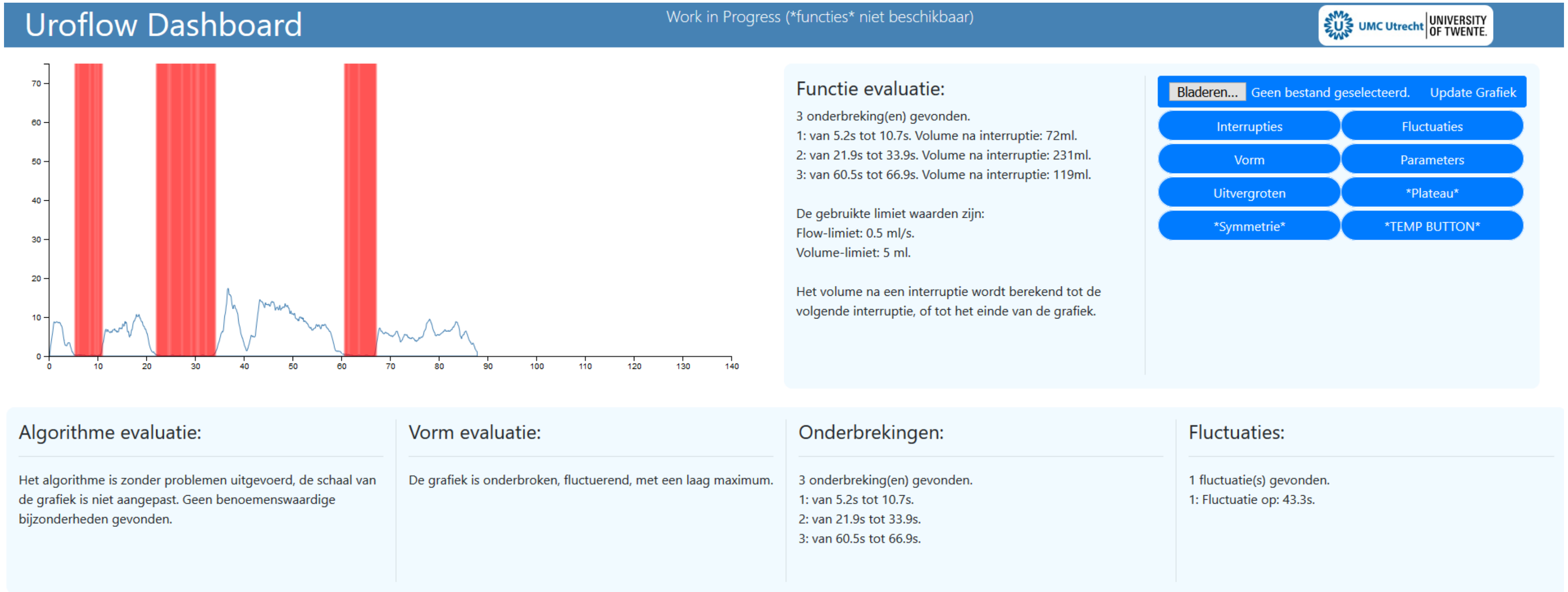


Figure 65: Uroflow Dashboard interruptions function

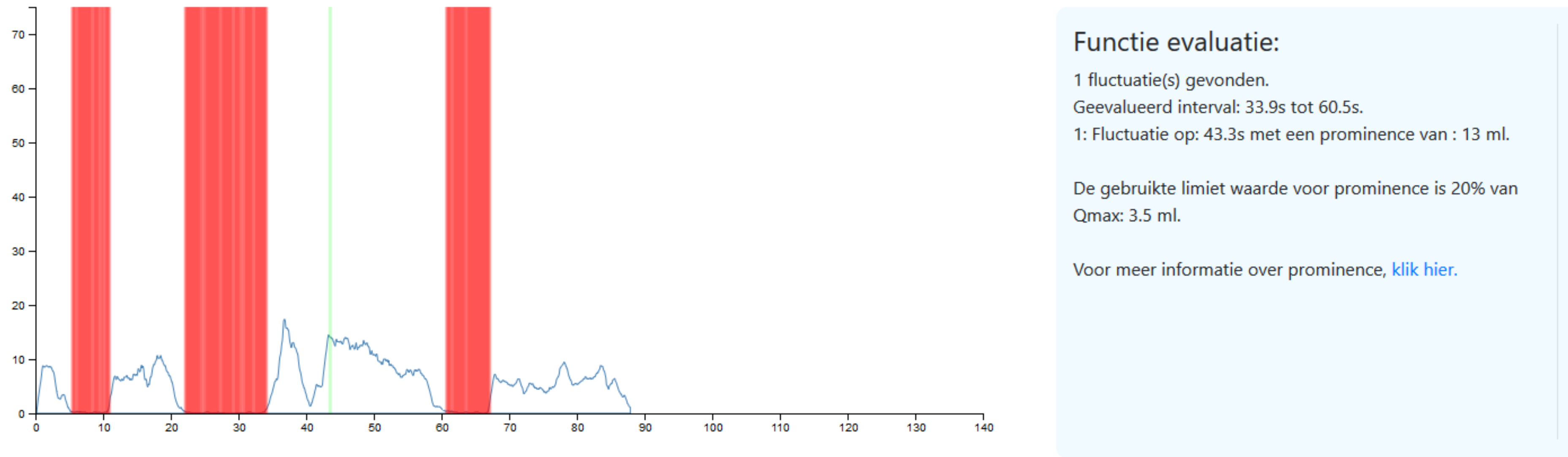


Figure 66: Stacking mark ups in the Uroflow Dashboard

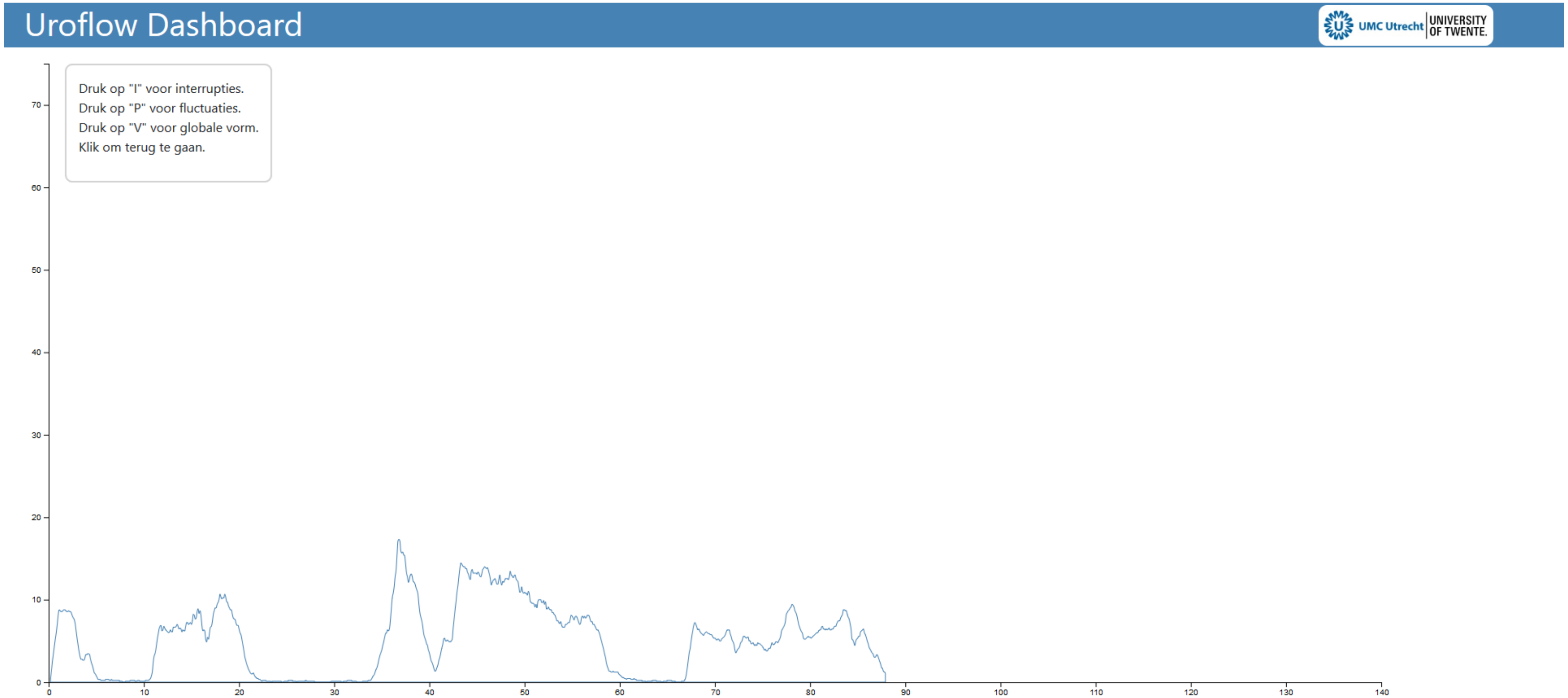
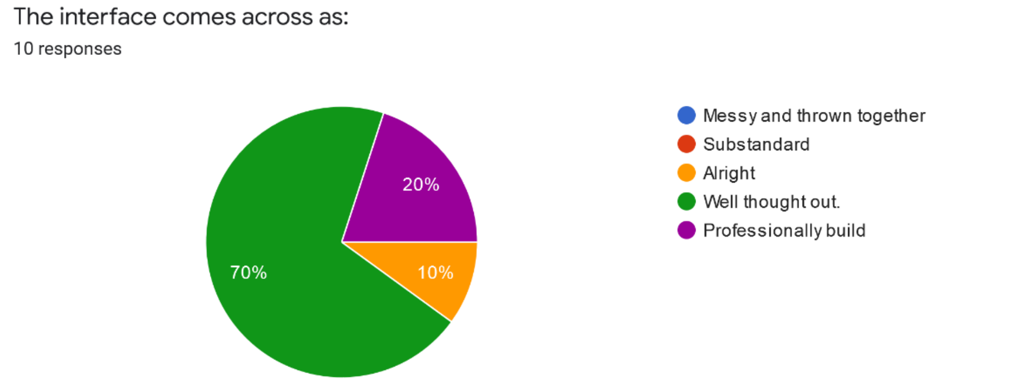
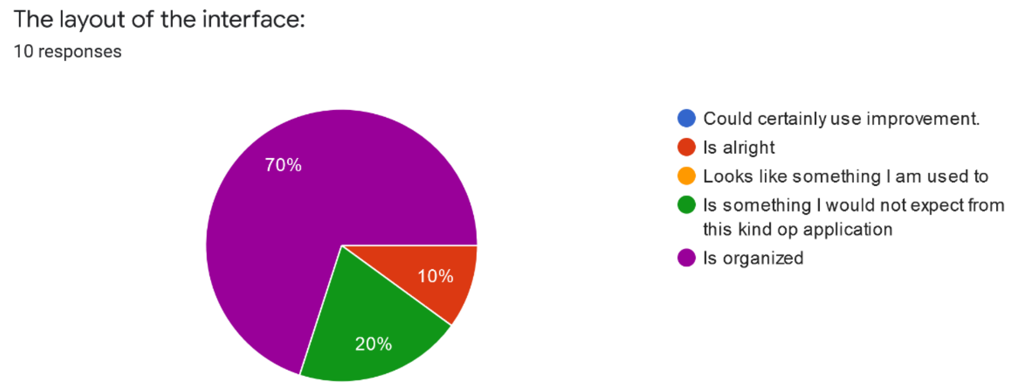
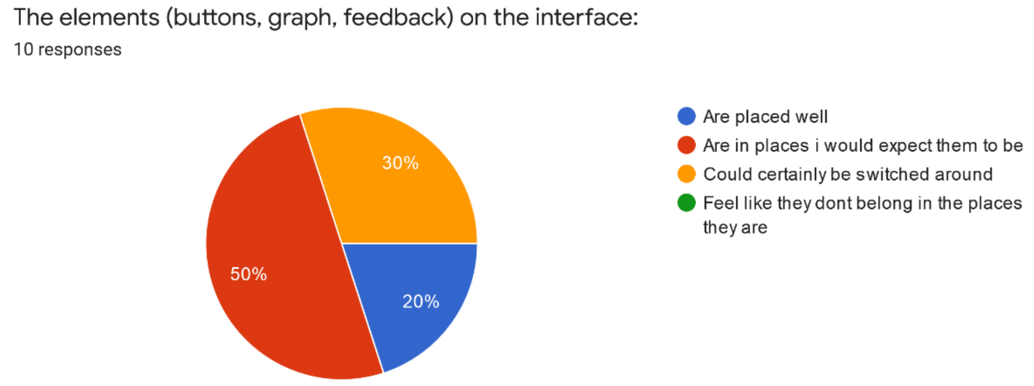


Figure 67: The zoom function in the uroflow dashboard

Appendix K: Layout and design questionnaire results.

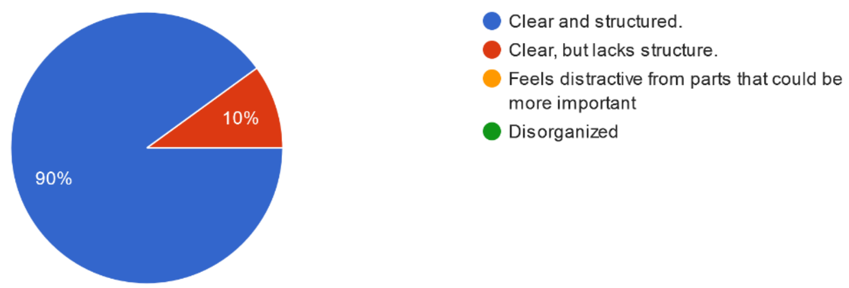






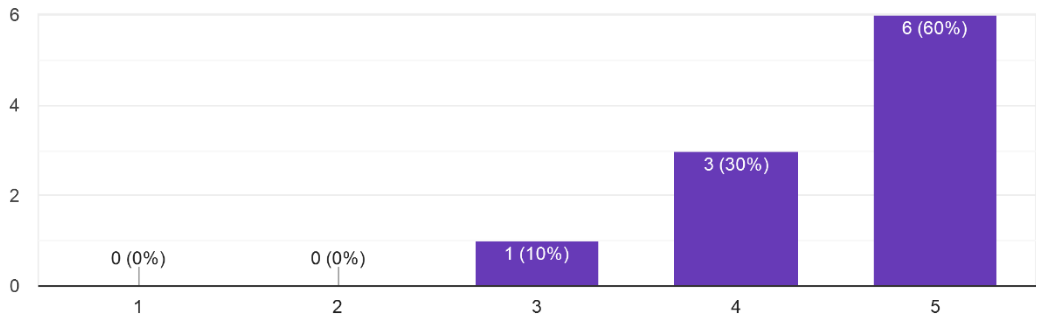
The information displayed on the interface is:

10 responses



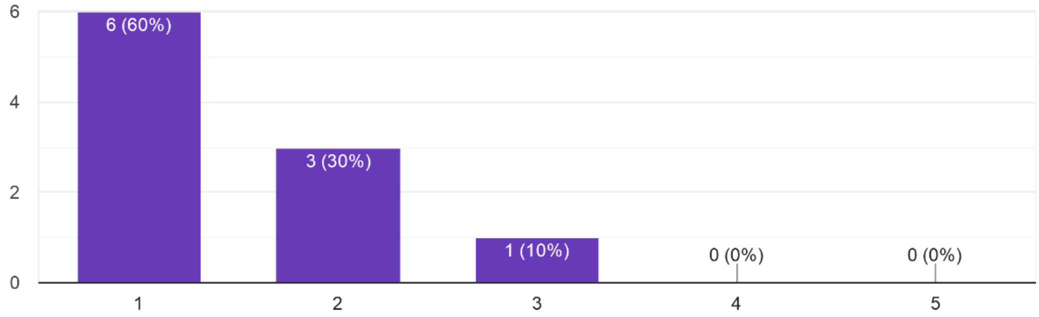
The interface presents information i would suspect (if I was an urologist)

10 responses



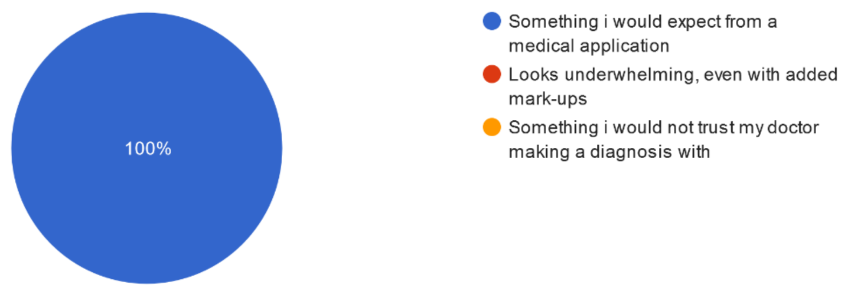
The interface becomes hard to look at after a while

10 responses



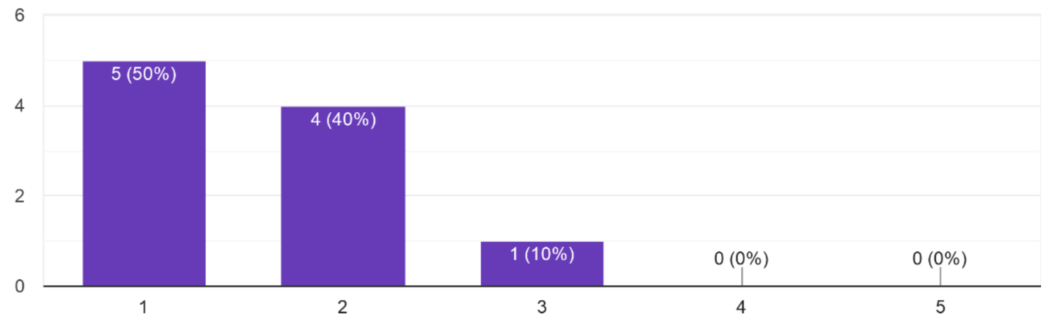
Consider the image below for the following questions. The graph is:

10 responses



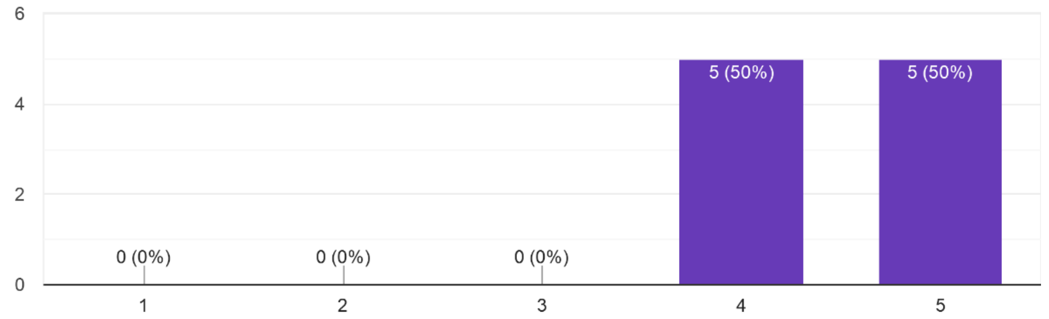
The sheer amount of information confuses me

10 responses

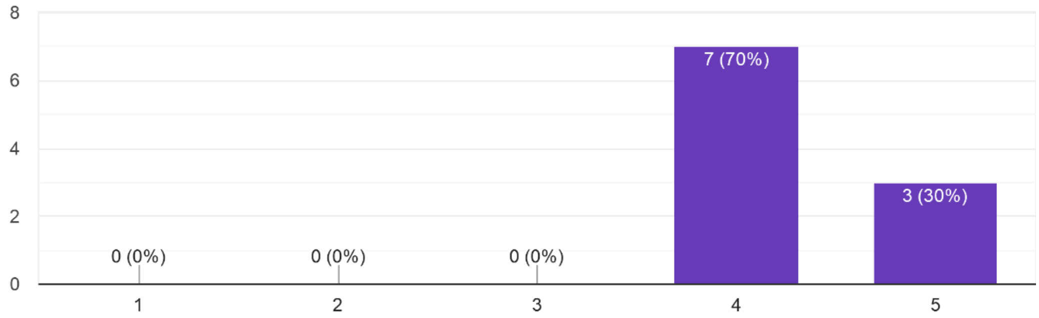


The system looks like it has added value for diagnosing

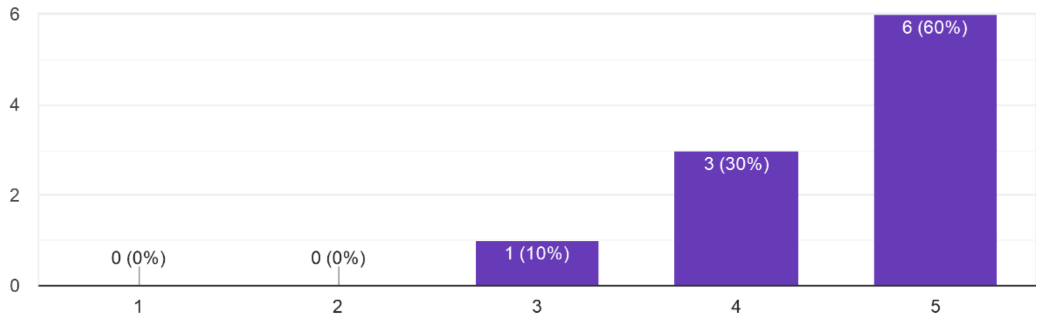
10 responses



The system looks like something I would trust, if my diagnosis was based on it.
10 responses



The systems seems practical
10 responses



I would use a similar system if one existed for my use case.
10 responses