Stress Pendant Project

Data Visualizations

Figure 1. (McCormack 2016)

Apoorv Kansal

Supervisor:
Professor Jon McCormack, Director of sensiLab

INTRODUCTION

The Stress Pendant project was a real-time social experiment that utilised wearable technology and mobile phones to capture various measurable data that helped identify when people were stressed at any place or any time.

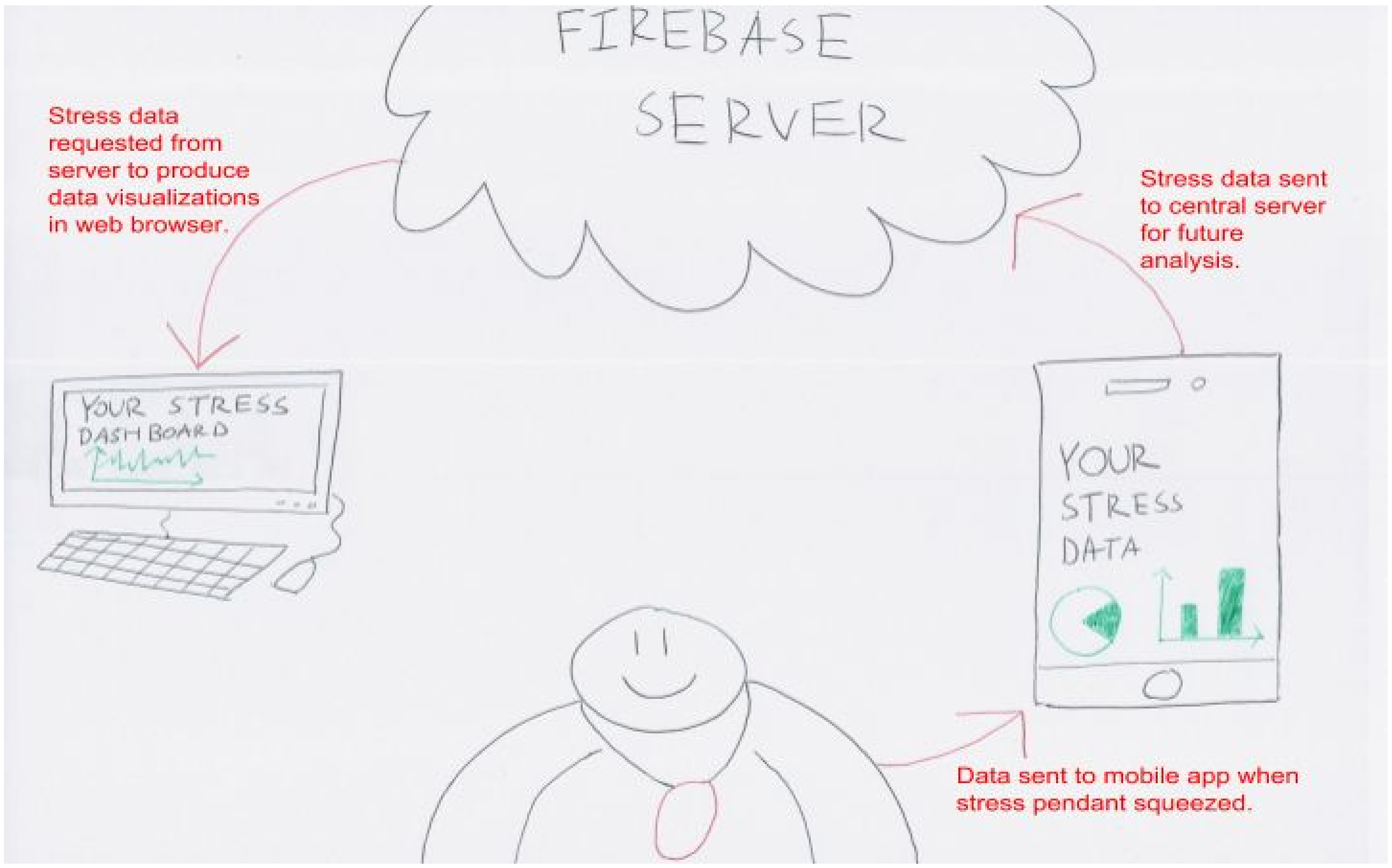

FIREBASE SERVER
Stress data requested from server to produce data visualizations in web browser.
Stress data sent to central server for future analysis.
YOUR STRESS DASHBOARD
YOUR STRESS DATA
Data sent to mobile app when stress pendant squeezed.

Figure 2. Diagram explaining the technical architecture of the Stress Pendant project.

Participants of the experiment simply squeezed the stress pendant whenever they felt stressed. The pendant then captured time, temperature, accelerometer data and more variables and then sent this data to a paired mobile phone via Bluetooth. The mobile phone showed personal data about the participant's stress history and also relayed the captured data to Google's Firebase server for data collection and analysis. Researchers at sensiLab have also produced a web-based app that shows various data visualizations of stress data collected. The currently live web-based app presents a geolocation heatmap highlighting where people felt stressed. It also included a time slider allowing users to see when people were stressed. These different uses of variables have allowed users to interact with the data visualizations.

There were various technological aspects to this project. The task required researchers to focus on producing the relevant hardware for the stress pendant and assembling the sensors. This also required a fair amount of programming in C/C++ in order to communicate with the mobile app. Another task required researchers to create the mobile app on the IOS and Android platforms meaning that one needed experience in Swift/Objective C and Java respectively. Furthermore, for data visualization, researchers also needed to work on the web-based app, which required knowledge of Javascript (and various frameworks such as D3), Firebase API and also basic HTML/CSS. Researchers also needed to focus on the shape and material used to produce an appealing and simple stress pendant for usability.

Monash researchers were intrigued about the potential of wearable technology and how it allowed collection of data about an individual's body and mind. The ability to continuously quantify an individual's life can allow people to make better life decisions and identify health risks.

As a research student, I was particularly inclined to develop the data visualization aspect of the project as I saw Data Science as a booming and emerging field and analyzing real stress data was the best opportunity to dive into this area.

AIMS AND MOTIVATION

The fundamental principles and aims that underpin this project were: presenting and exploring known and obvious data freely with clarity (eg. displaying a heat map of stress locations) to produce informed decisions and displaying data in new ways to discover unknown factors and potential causes. Essentially, sensiLab researchers were interested in having a better understanding of our stress and improve the stress levels of many individuals.

LEARNING PROCESS

The learning process was essentially divided in two main areas: programming and understanding the importance of data visualizations in today’s society. At this stage, the focus was on acquiring the necessary skills in order to actually create a final product that met the aims.

On the technical side, I first focused on exploring various Javascript visualization frameworks (eg. plot.ly, Canvas.js and Chart.js) by experimenting with examples, API documentation and watching Youtube videos. I also focused on producing simple examples such as found in Figure 3.

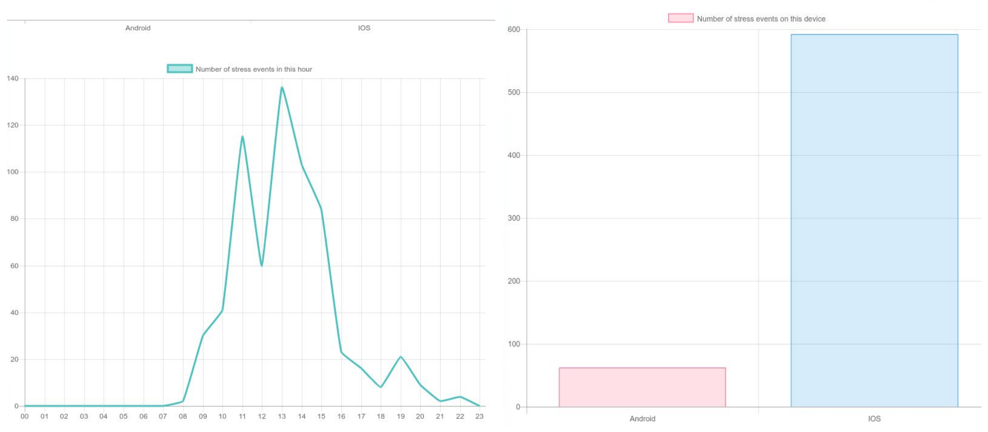
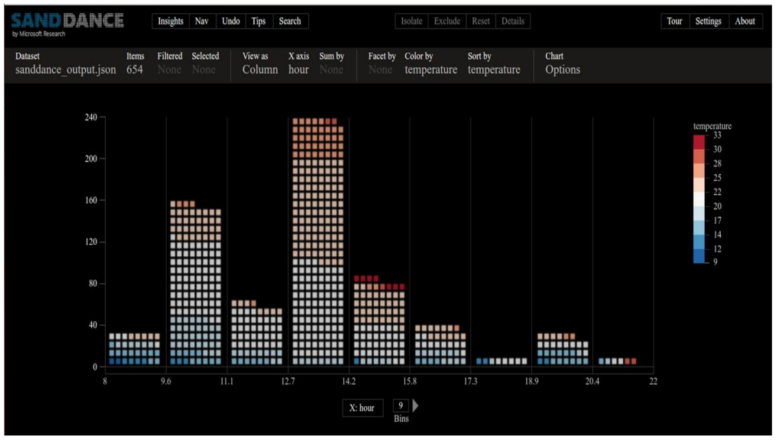


Figure 3. Simple examples I made for learning purposes

Here, I have simply produced a line chart with number of stress events against hourly intervals and a histogram with number of stress events happening on each platform available.

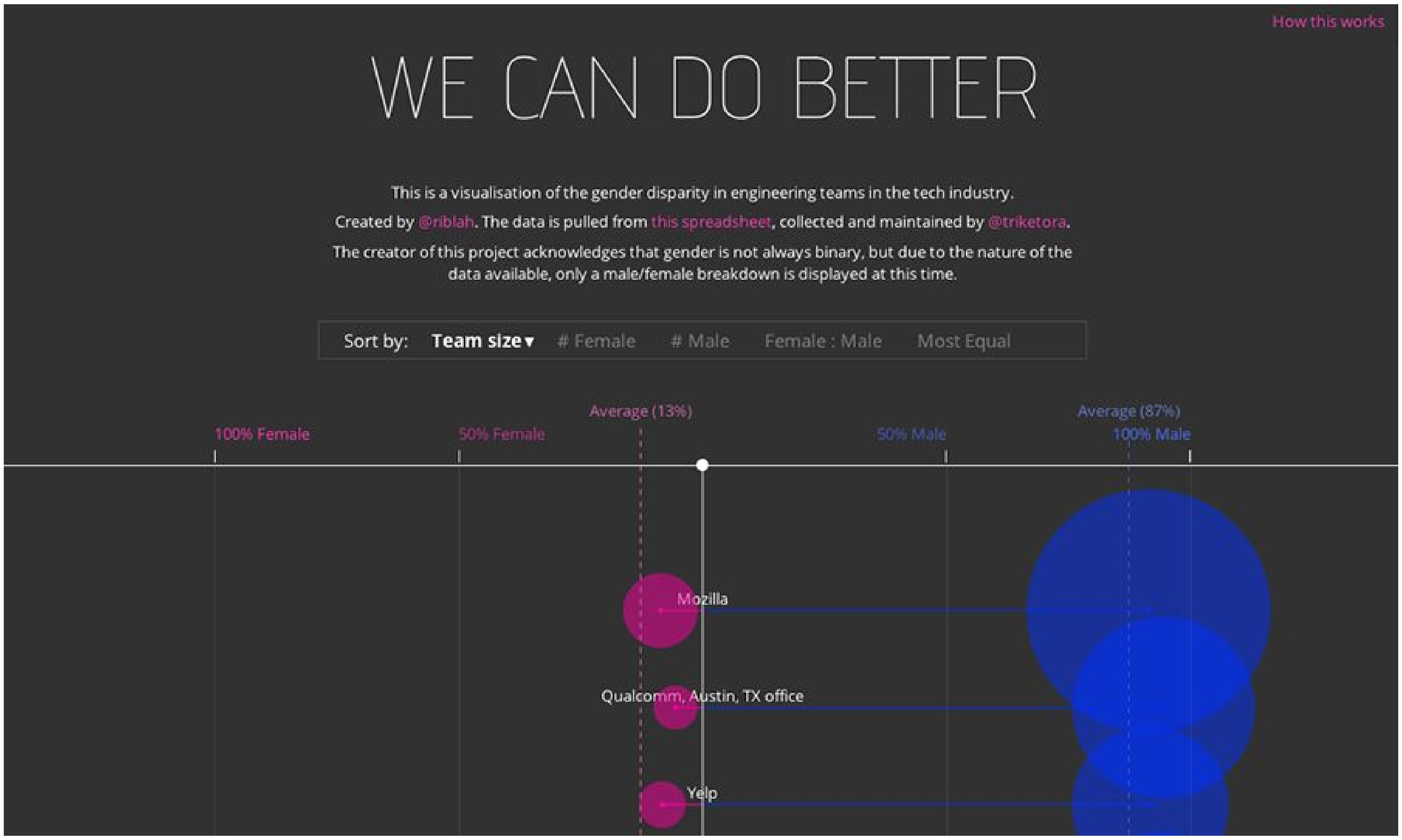


Dataset: sanddance_output.json · Items: 654 · Filtered: None · Selected: None · View as: Column · X axis: hour · Sum by: None · Facet by: None · Color by: temperature · Sort by: temperature · Chart: Options

Figure 4. Work done on Microsoft’s SandDance.

I also utilised Microsoft's SandDance which was a customizable visualization tool that essentially allowed users to input any dataset they desired and explore it through different forms of visualizations. I inputted stress data that was collected from previous trials of the project into this tool to produce the visualization found in Figure 4. Here, there was a histogram where each bar was made up of small boxes. Each small box represented a stress event. On top of this, I coloured each stress event by its temperature. By exploring SandDance, I was able to truly understand the aim of the project. It was not about finding inferences and analysing the stress data but to actually allow users of the customizable dashboard to do this form of analysis with freedom and clarity of a customizable dashboard.

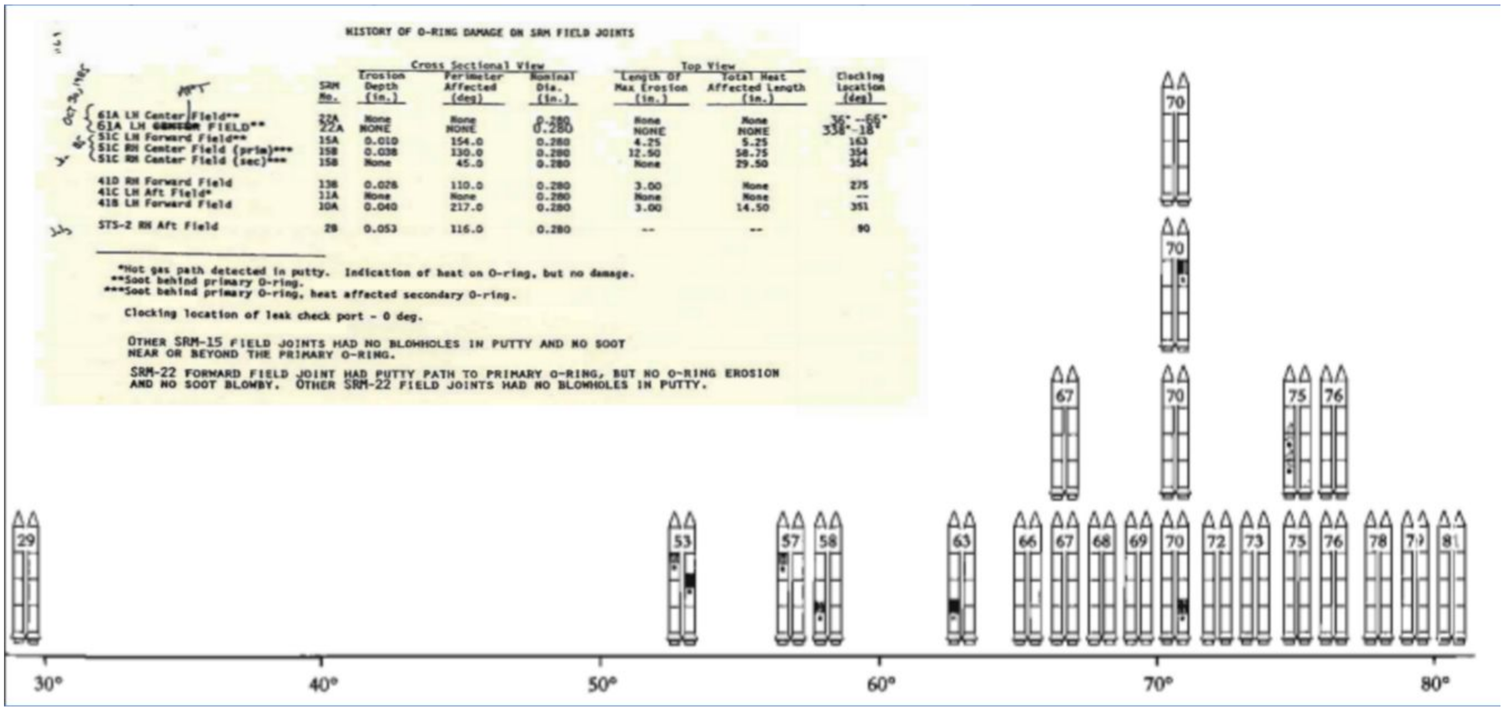
Along with technical learning, I also focused on the needs of data visualizations. In a Ri Liu's presentation to sensiLab about data visualizations, she emphasised the need for unique and different forms of visualizations to produce various and strong impacts (Sensilab monash 2016). She mentioned how data visualization was bridge between Science and Art and that one needs to focus on utilising the right form of visualization to send the right message across (Sensilab monash 2016). For instance, she produced an interactive visualization below that shows the gender population inequality amongst Engineering teams in Silicon Valley. The large blue circles on the right represented males in a team while the small pink circles on the left represented the females. The Y-axis showed different companies. Her goal was to essentially exploit the gender inequality still found in IT and Engineering and basically build a summary of proof through this visualization (Sensilab monash 2016). As such, it is clear to see that different visualizations are tailored for different goals and messages to the audience. Liu clearly showed how visualizations were really artworks that should convey ideas and themes in a flawless manner.


WE CAN DO BETTER

This is a visualisation of the gender disparity in engineering teams in the tech industry.
Created by @riblah. The data is pulled from this spreadsheet, collected and maintained by @triketora.
The creator of this project acknowledges that gender is not always binary, but due to the nature of the data available, only a male/female breakdown is displayed at this time.

Figure 5. Ri Liu's work on exploiting gender inequality in Engineering teams (Liu 2014)

Moreover, visualizations also serve as alarming indicators in difficult scenarios. For instance, the cause of the Space Shuttle Challenger disaster was due to two rubber O-rings leaking. These rings failed their purpose as the shuttle was launched on a very cold day. However, NASA officials analysed the findings and believed the problem was minute and the rocket could be launched. Unfortunately, they did not make informed decisions on the data supplied and 7 crewmembers died during the launch as a result. Edward R Tufte, data visualizer, produced a convincing argument that scientists failed to identify the serious O-ring problem due to weak and poorly made data presentations (Tufte 1997 pp. 48-50). He showed the data NASA officials were reviewing in Figure 6 (top-left diagram).



HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

		Cross Sectional View			Top View		
	SRM No.	Erosion Depth (in.)	Perimeter Affected (deg)	Nominal Dia. (in.)	Length Of Max Erosion (in.)	Total Heat Affected Length (in.)	Clocking Location (deg)
61A LH Center Field**	22A	None	None	0.280	None	None	36°-66°
61A LH ~~CENTER~~ FIELD**	22A	NONE	NONE	0.280	NONE	NONE	338°-18°
51C LH Forward Field**	15A	0.010	154.0	0.280	4.25	5.25	163
51C RH Center Field (prim)***	15B	0.038	130.0	0.280	12.50	58.75	354
51C RH Center Field (sec)***	15B	None	45.0	0.280	None	29.50	354
41D RH Forward Field	13B	0.028	110.0	0.280	3.00	None	275
41C LH Aft Field*	11A	None	None	0.280	None	None	--
41B LH Forward Field	10A	0.040	217.0	0.280	3.00	14.50	351
STS-2 RH Aft Field	2B	0.053	116.0	0.280	--	--	90

*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.
**Soot behind primary O-ring.
***Soot behind primary O-ring, heat affected secondary O-ring.

Clocking location of leak check port - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.

SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY.

Figure 6. Comparisons between NASA's and Tufte's visualizations (Tufte 1997 pp. 41-50).

This form of tabular data and rough annotations made it hard to decipher any potential problems or identify important factors. As such, Tufte showed his own data visualization in Figure 6 (bottom-right diagram), which would have easily alarmed scientists to abort the mission. It showed the Challenger space shuttle being launched at a temperature of 29 degrees. The damaged rockets clearly occurred between 50 and 70 degrees and the more successful rockets were launched a temperature above 70 degrees. As such, this graph strictly emphasises that the Challenger should have been launched at a much higher temperature. By simply having different visualization structures, NASA could have prevented one of the biggest disasters in history. Tufte clearly highlighted the importance of having the right visualization for the intended effect.

Along with with seeing data visualizations in different forms to produce different impacts, visualizations are also important to process and analyse large datasets that are being produced in today's society. Another data visualizer expert, Matthew Ward, emphasised how visualizations were significantly more than just a subfield/extension of computer graphics (Ward, Grinstein & Keim 2015, pp. 3-21). He saw that the aim of visualization is about communication and that it encompasses multiple different fields together such as statistics, human-computer interaction, perceptual Psychology, databases and many more (Ward, Grinstein & Keim 2015, pp. 3-21). In contrast, computer graphics were used to produce synthetic images and animations for visual realism (eg. applied to cartoons, entertainment etc.). He claimed that many different fields such as marketing, business and medical fields now have huge datasets from research (Ward, Grinstein & Keim 2015, pp. 3-21). As such, most of these fields require tools, particularly data visualizations, to help people make better informed decisions and freely explore their data. Ward clearly distinguishes the purpose of visualizations which was to allow people to make better informed decisions when analysing large datasets in various different industries and not just about producing appealing and eye-catching graphs.

Visualizations Development

Design

After utilising Microsoft's Research product, Sanddance, it was fairly evident that they had used a design base below their visualizations. It was easy to see visualizations being hidden, added, removed but still maintaining the design base. As such, it made logical sense to first focus on building a design base for the Stress Pendant dashboard. I decided to utilise a jQuery (Javascript framework) and Bootstrap (Javascript and CSS frameworks) along with a Bootstrap theme, AdminLTE, to produce the simple dashboard base in Figure 7.

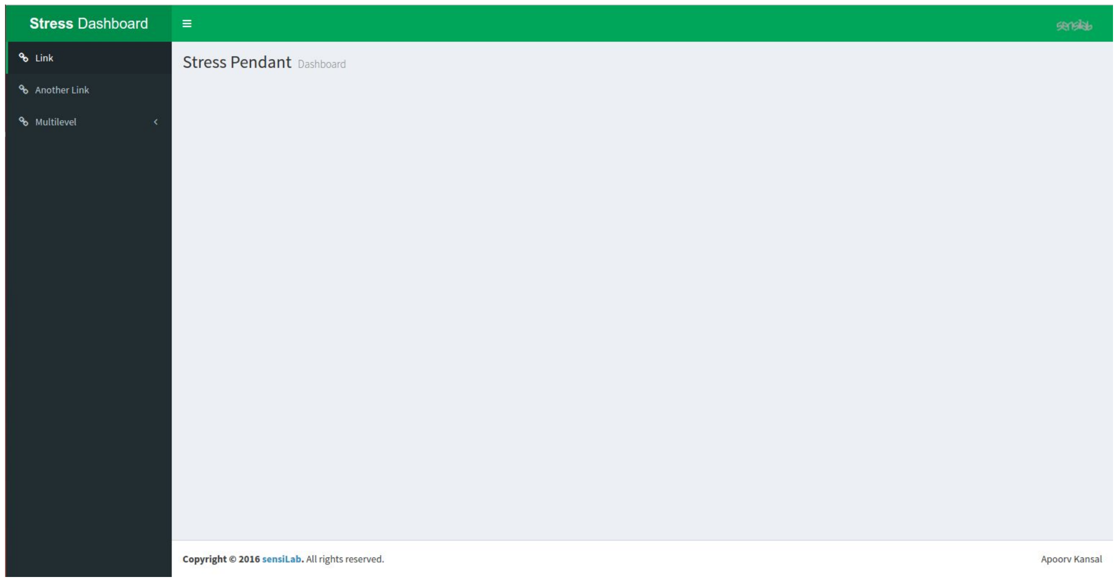
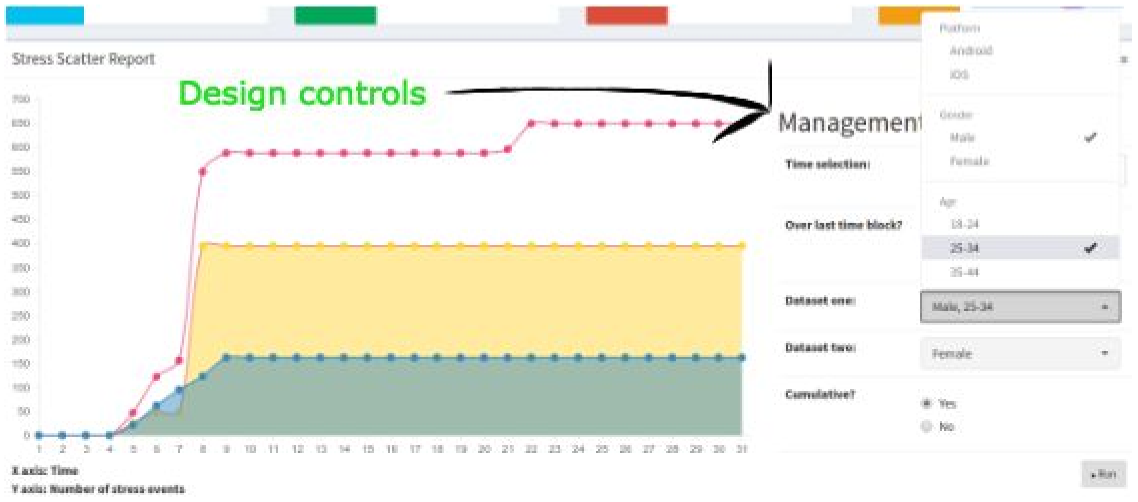


Figure 7. Design base of the final product.

Now that a base was completed, it was very simple to quickly add/ remove/ modify visualizations (also known as widgets) to the dashboard without being concerned about layout issues in different and varying computer screens. This ensured that users could work on the dashboard from multiple different devices.

Simultaneously, it was important to also explore and focus on different visualization frameworks and pick out interesting and useful examples for the project. Having these examples ensured that I was applying the learning and exploration I did previously to the final product. It also allowed me to brainstorm more visualizations. As such, I explored plot.ly js, Canvas.js and Chart.js.

From here, it was beneficial to see how these picked out examples can be applied to the dashboard from a design perspective. Once they were added (as shown in Figure 8.), the next step was to decide and add design controls that the user would use in order to customize and explore the visualization. At this stage, the aim was about design and comprehending the final look. It was essential to focus on design before functionality as it ensured that I understood the scope of the final product and knew exactly what needed manipulation (eg. design controls defined what and how the visualization would be manipulated).



Time selection:

Over last time block?

Dataset one: Male, 25-34

Dataset two: Female

Cumulative? [✓ Yes] No

Figure 8. An example of a visualization in the final product.

Once the design step was complete, it was necessary to analyse the example code and convert them into usable Javascript functions that managed rendering of the visualizations. Essentially, these drawing subroutines were viewed as "rendering machines" where I only needed to input parsed data to output a customized visualization.

Rendering Visualizations

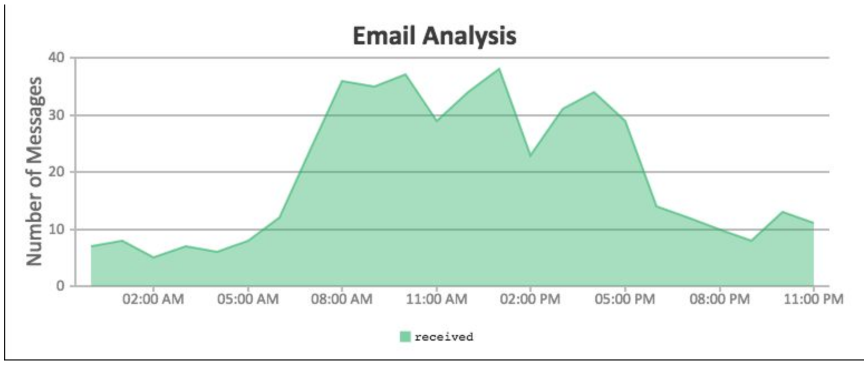


Figure 9. An example graph.

In the above figure, I had to focus on analysing peculiar elements of the example visualization such as the x-axis, y-axis, legend, co-ordinates etc. and then see where they were implemented in the example code. From here, I had to convert the code into modularized Javascript functions where these peculiar elements would be inputted upon a function call. The main reason that I built these drawing code snippets was to ensure that my code was modularized and reusable along with abstracting the technicalities and design components of the visualizations from future steps such as retrieving design control selections and parsing stress data.

Data Manipulation

Now that the example visualizations were implemented along with the design controls and the "rendering machines", the next step was to parse the stress data based on selections made by the user and simply inputting this data into the drawing subroutines. This final step satisfied the main aim of allowing users to explore the dataset freely and with clarity to a large extent as users were now able to manipulate the visualizations.

Event Handling

To ensure that the visualizations were highly responsive, I had to build event listeners that would refresh the graph every time a design control was changed by the used (eg. graph refreshes when user changes dataset to males).

Production of Visualizations during Runtime (Algorithmic Overview)

There were two different events that would trigger the production/refreshing of a visualization. These were when the user logs in or when the user changes a design control. At both of these times, the browser must have access to the Firebase stress data and also able to retrieve current user selections from the design controls. Provided that these two conditions were met, the visualizations underwent a refreshing process. Essentially, this process encompasses modularized subroutines that would reinitialize the graphs by retrieving current selections made by the user from the design controls and then parse the Firebase stress data accordingly. This parsed data was then sent to the rendering visualizations' subroutines from which new visualizations were produced. By having separate and structured code snippets in organised stages, this refreshing process could be used at two different times (when the user logs in or changes a design control).

Dashboard State Development

Another main component of the project was to implement a persistent dashboard state. This concept meant that each user would have their own personalised dashboard. Upon logout, their current dashboard state would be saved into Firebase and upon login, it would be restored. In order to produce such a system, I needed to first build the authentication system. By exploring the current data visualization heatmap project, I was able to design a consistent authentication scheme which aligned directly with Firebase's authentication mechanisms.

In order to save the state into Firebase, I had to remodel the Firebase's database where I essentially added a "state" node that stored all the design control selections for a particular user. On top of this remodelling process, it was important to modify the authentication rules to ensure that participants were only accessing their own state. I also implemented a default state for new users. By having the default state in Firebase, the software was able to reuse the restoring state algorithm and sensiLab researchers or managers of the program were able to customize the default state.

Process of after Successful Login

The dashboard was first completely resetted where all options and attributes were removed. From here, the Firebase stress data was analysed to import new and available options/attributes (eg. importing age groups into the dashboard). The code then imported the authenticated user's dashboard state from Firebase. This state data was parsed and then applied onto the dashboard by automatically making selections on the imported attributes/options. Once the the Firebase stress data was requested and loaded, the visualizations were produced (see Production of Visualizations during Runtime (Algorithmic Overview)). The user was finally redirected from the login page to the dashboard view.

Process of Logout

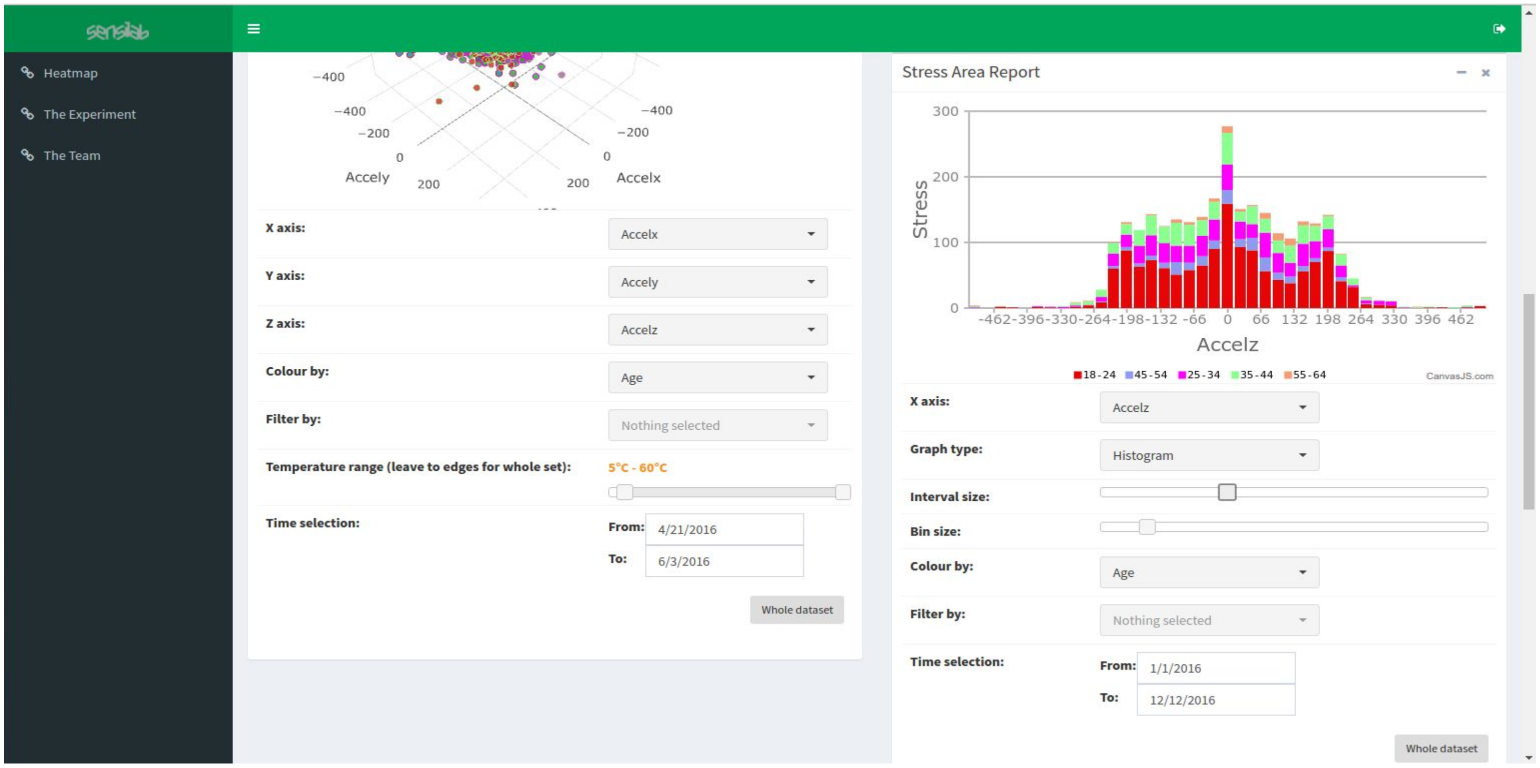
The logout system first collected the current selections made on the design controls in the dashboard and then finally saved it into Firebase. From here, the user was logged out and returned back to the login page.

Integrating Current Data Visualization Project

To ensure that the existing project's code did not break while integrating into the the stress dashboard, I utilised an iFrame where their code would not need any modification. There were two main problems with using an iFrame. The first was potentially needing to authenticate twice (once for the dashboard and then another time for the iFrame). Fortunately, because Firebase's realtime authentication system was utilised, Firebase resolved this issue immediately as it automatically logs the user into the current existing project (heatmap iFrame) once they log in through the dashboard. It seemed that Firebase was able to detect and maintain persistence of authenticated users even in different HTML frames but on the same browser. The second problem was that there was a same-origin security policy issue when turning on/off full screen mode for the heatmap. Essentially, I was unable to listen for the "esc" key in the iFrame when it was on full screen mode through Javascript as modern browsers see this as a major security risk. As such, every browser blocks any script trying to access a frame with a different origin. An origin is considered to be the same if the protocol, hostname and ports are all the same across both frames. So to resolve this issue, I simply ran a HTTP server which ensured that browsers knew that both frames (iFrame and main frame) were coming from the same origin and as such, there would be no security issues when the main frame listened for the “esc” key in the iFrame in order to escape out of fullscreen mode.

USER’S PERSPECTIVE

From a user’s standpoint, they were able to view and analyse stress data by logging into the dashboard which would show an array of different widgets as shown in Figure 10. They would be allowed to manipulate the dashboard and explore the collected stress data in different customizable visualizations.



X axis: Accelx
Y axis: Accely
Z axis: Accelz
Colour by: Age
Filter by: Nothing selected
Temperature range (leave to edges for whole set): 5°C - 60°C
Time selection: From: 4/21/2016 To: 6/3/2016

X axis: Accelz
Graph type: Histogram
Interval size:
Bin size:
Colour by: Age
Filter by: Nothing selected
Time selection: From: 1/1/2016 To: 12/12/2016

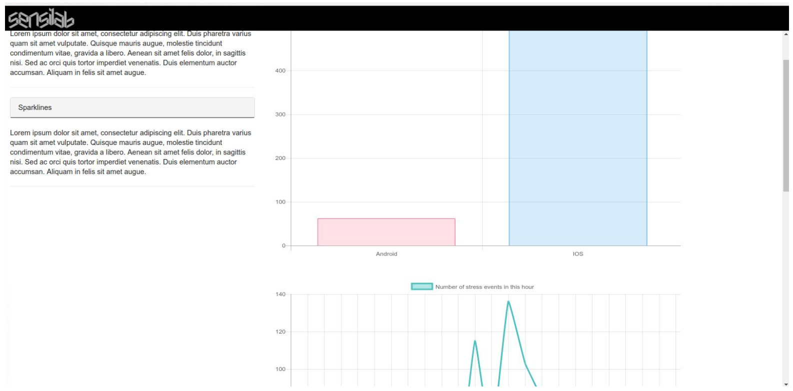
Figure 10. Final product.

To manipulate the visualizations, users would change the controls that are located next to each visualization. Common controls include: filtering dataset by variables (eg. time, temperature, gender, age and platform), colouring by subsets of the population and also customizing axes with different variables (eg. accelerometer values, temperature, specific time selections and latitude/longitude). When any control was changed by the user, the visualization would refresh to reflect the changes. Upon logout, their dashboard state (the selections made through the controls) would also be saved to ensure that it would be restored when logging back in. The dashboard would automatically update as more stress events were recorded on the server.

DIFFICULTIES

There were a number of difficulties encountered throughout the project and most were solved easily while some were more challenging than expected. At the start of the project, there was some difficulty in inputting the collected stress data into a customizable dashboard, Microsoft’s SandDance, for exploration. Not knowing how exactly Sanddance expected its input, Professor McCormack provided assistance on the dataset should be structured so that it can be used in Sanddance. As such, I had to write Javascript code to parse and format the stress data so that SandDance was able to use the data.

Moreover, while learning by creating small examples, I encountered layout issues just like in Figure 11.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt condimentum vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi. Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan. Aliquam in felis sit amet augue.

Sparklines

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt condimentum vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi. Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan. Aliquam in felis sit amet augue.

Figure 11. Development of final product.

Here, I found difficulty in aligning the graphs to fit the whole page. At this point and after using SandDance, I realised the need for a responsive and mobile-friendly framework, such as Bootstrap, might become useful.

Another difficulty of the project was that I had to work under the constraints of a limited dataset. Due to the small sized dataset, it was sometimes difficult to perceive the potential use of some of the created visualizations as there weren't enough data points to explore.

CONCLUSION

The Stress Pendant project has allowed me to explore research at sensiLab and comprehend how researchers and students collaborate to work on innovative projects. I have learnt various Javascript visualization frameworks along with working with Firebase and understanding the whole software architecture behind the project. I have successfully created a usable and customizable stress dashboard that allowed people to explore the collected stress datasets.

There was potential future work that can done on the project. For instance, the dashboard could be tailored for different research projects with different datasets. Now that the dashboard has a responsive and mobile friendly design base, it was easily possible to add/remove and modify visualizations. Finally, analytics data could be set up (eg. Google analytics or even the dashboard itself) to better understand how users utilise the dashboard and how long they stay on it. This form of insight would allow researchers to make better informed decisions on what kind of visualizations should be utilised.

In terms of improvement, there could be even more flexibility involved in the dashboard and abstract the system completely from the stress data. That way, the dashboard could become completely independent of data and allow for analysis on different projects. However, the aims of presenting and exploring data freely and with clarity along with allowing informed decisions to be made about the stress data has been met to a large extent. The platform would allow sensiLab researchers to gain a better understanding of the public's stress and potentially improve stress levels of many individuals as they were able to analyse the stress data in different forms and manipulate it significantly.

REFERENCES

1. McCormack, J 2016, Stress Pendant Project, online image, viewed 5th October 2016, http://sensilab.monash.edu/project/stress-pendant/
2. Sensilab monash 2016, sensiLab Forum - Ri Liu, online video, viewed 3rd September 2016, https://www.youtube.com/watch?v=M0glWgk3EPk
3. Liu, R 2014, We Can Do Better, online image, viewed 17th October 2016, http://ri.id.au/work
4. Tufte, E 1997, Visual Explanations: Images and Quantities, Evidence and Narrative, Graphics Press, Connecticut USA
5. Ward, M, Grinstein, G & Keim, D 2015, Interactive Data Visualization: Foundations, Techniques, and

Applications, CRC Press, Florida USA