INTERNET OF THINGS

Team Ludvig

Alexander Lideràs
Amanda Borg
Anton Björkman
Hedda Johansson
Jessica Hagstedt
Ludvig Wiktorsson
Sara Ranstorp
Shaya Mokarremi
Simon Rosengren

Table of Contents

1. Project Overview

1.1. Background

A group of nine students taking the course Inledande webbprogrammering med Javascript at Malmö University, were given an assignment to visualize data from the IoT-lab on a web application. The IoT-lab has many different devices that are gathering data, that then transmits to the Amazon Web Services, where data is either stored in a database or as an live feed. We would use our knowledge of JavaScript, HTML and CSS and then utilize the Scrum method to work forward with project and the web application.

1.2. Use Case

A user should be able to visit this interactive website that visualizes different types of data from the IoT lab at Malmö university.

2. Technologies

The project is a client side web application written in Javascript. jQwery has been used to easily access html elements[1].

The AWS library for dynamoDB is used in order to access the database in which historical data is stored from the IoT lab[2].

The paho MQTT library is used in order to listen to messages sent from the IoT lab[3].

All of the charts used comes from the Google chart library[4].

2.1. Implementation Details

When the page is loading are we fetching the historical data from the Aws DynamoDB, and the live data and append the data to the different graphs. When the historical data is loaded we switch from the loadingscreen to our page. databaseHandler.js takes care of the requests to the DynamoDb. chart.js appends the historical and live data to the charts.

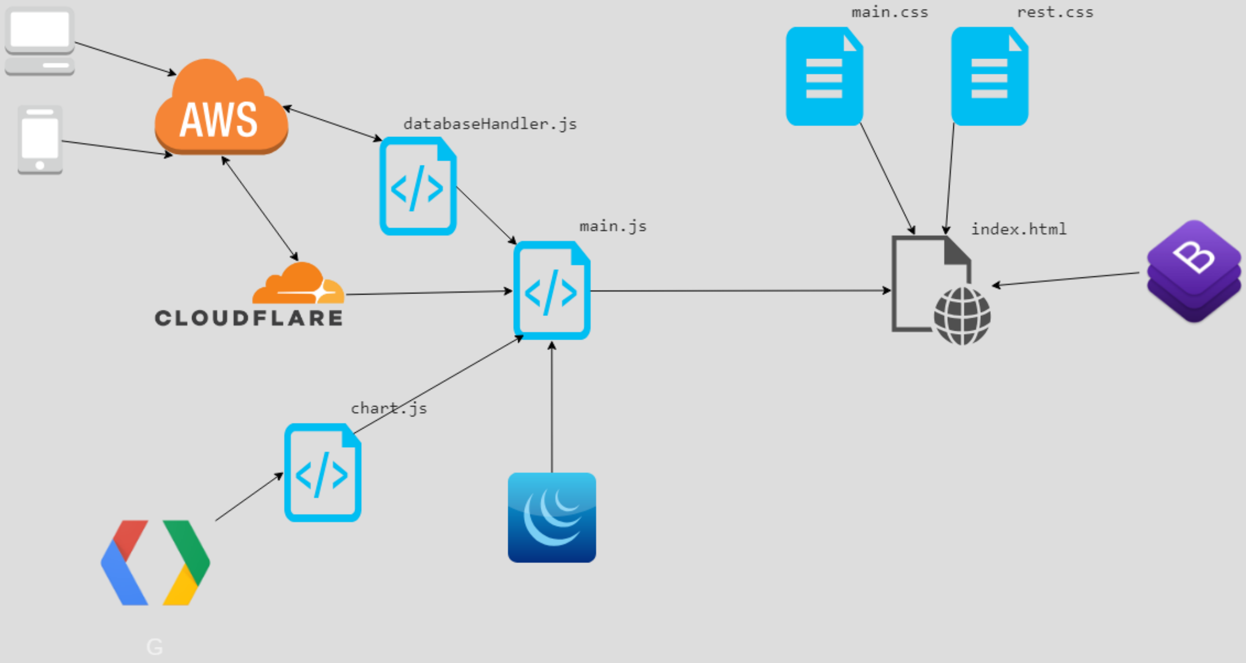
[1] https://jquery.com/

[2] https://aws.amazon.com/

[3] https://www.eclipse.org/paho/

[4] https://developers.google.com/chart/

2.2. Technical solution diagram



2.3. What worked

Everything technical worked out pretty well, the google charts were pretty straight forward. And to get the live data from the lab was pretty simple aswell.

2.4. What did not work and improvements

We had one idea of a 3D-chart, where the user was going to mix and mash up data and then view it in 3D-chart. But the result was the implementation that we did on a 3D-chart, didn't visualize the data in a better way than a traditionally 2D-chart. We also experimented on different ways to visualize the sound level. We thought of an “equalizer” that shows the noise levels, but because of the lack of updates coming from IoT-lab, this implementation was also discarded.

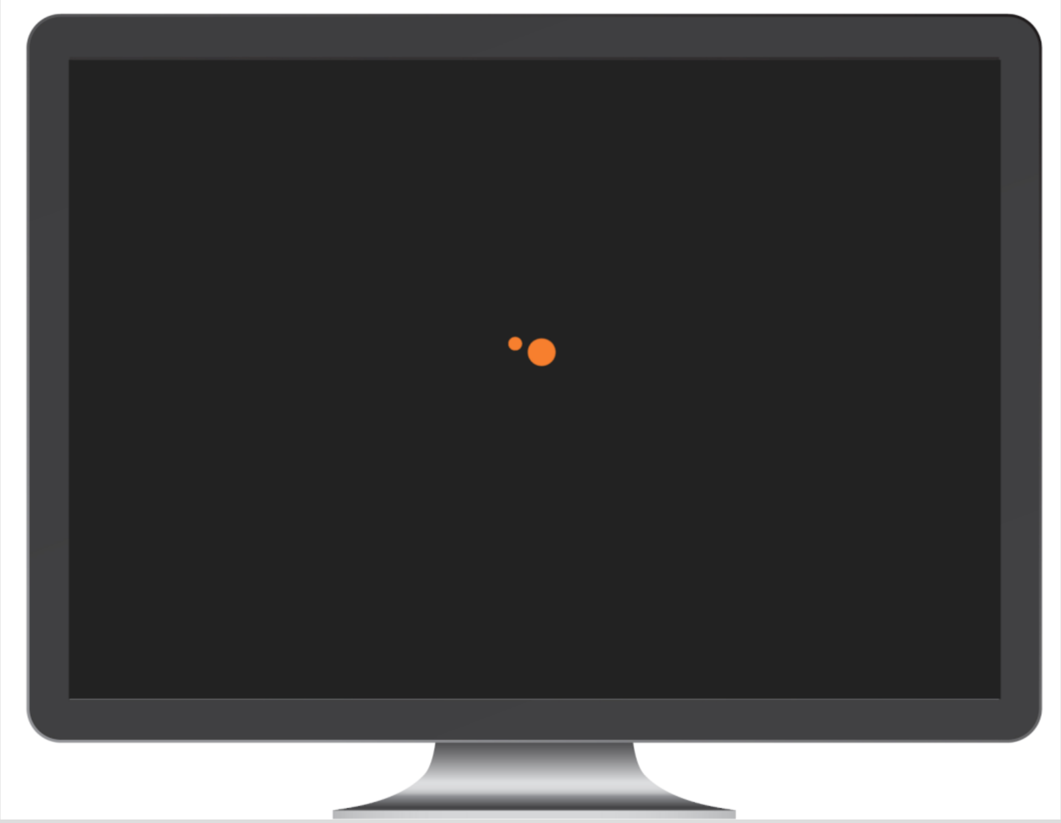
3. The Prototype

3.1. Usage scenario

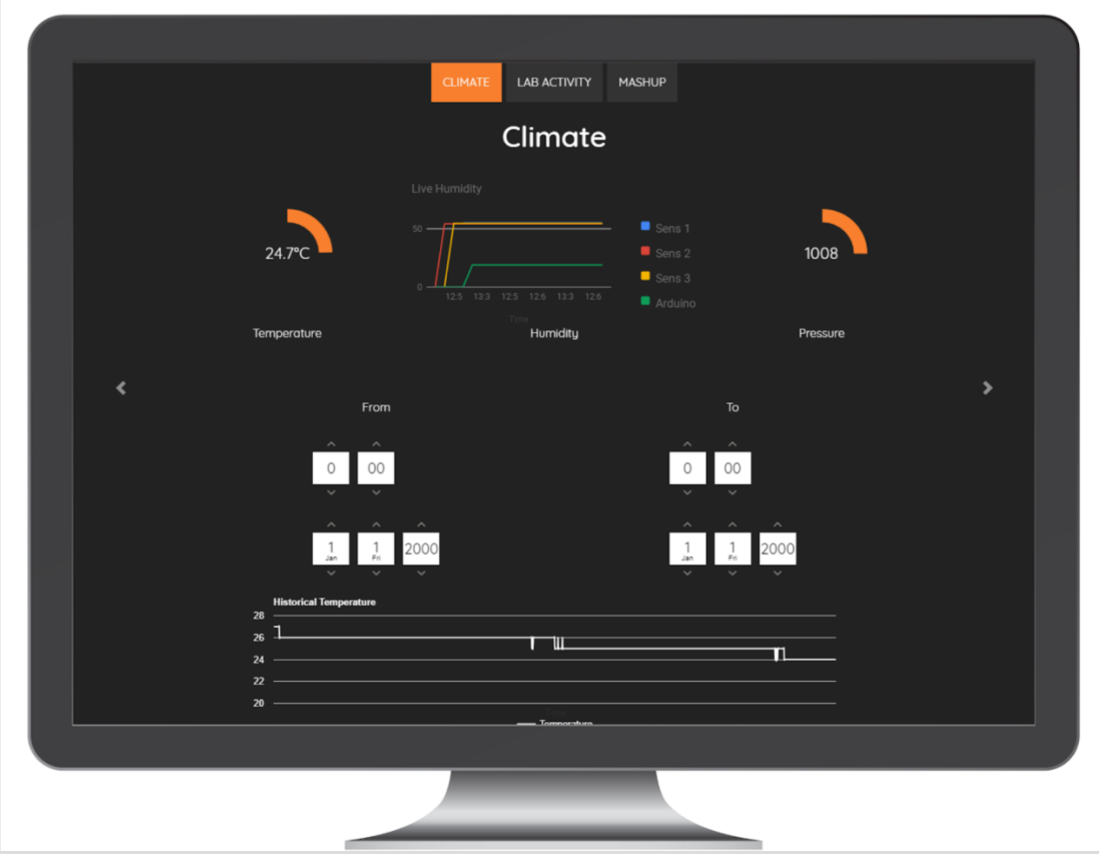
A user want to see the data from the lab. Both historical and live data. The user can switch between different tabs (Climate, Lab activity, Mashup) to see different graphs of the data the lab is producing.

3.2. Demo

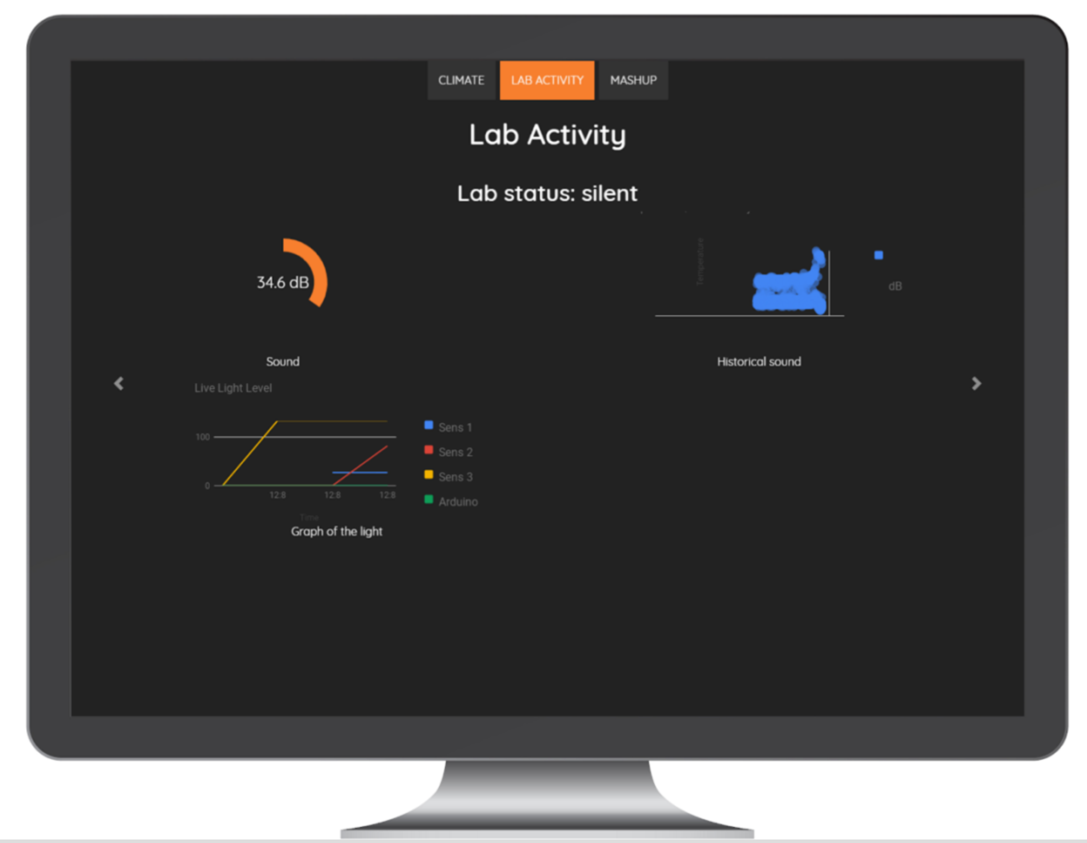
We have live demo of the web application on Github at: https://simonrosengren.github.io/js-project-iot/index.html.



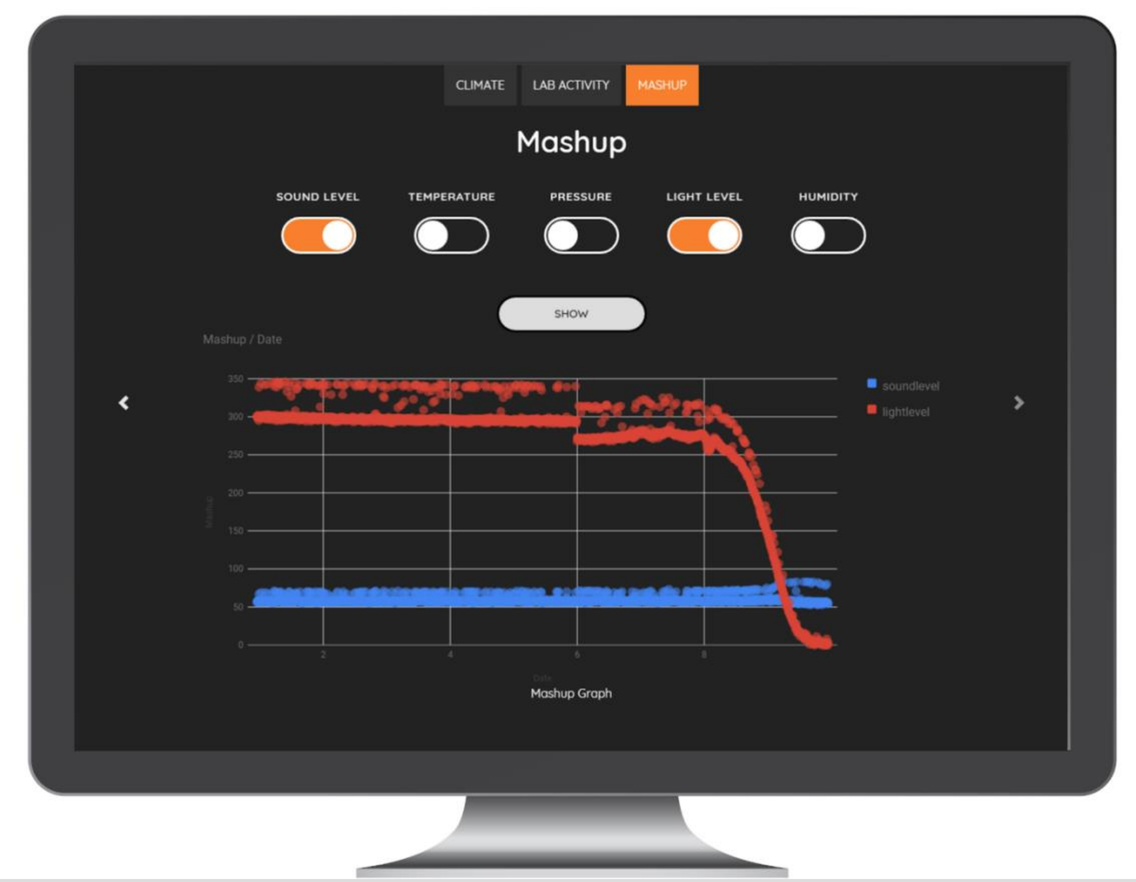
Loading screen



The first slide of the page



Second slide of the page



Third slide of the page

4. Project Challenges

One of the biggest challenges we had were that the lab restructured the database in the middle of the project without letting us know. It took us some time to get the full understanding of the new structure and made the progress complicated during that time. Another problem we had were that a couple of days before the restructure the lab was down so we couldn't keep working with the data.

4.1. Coding Challenges

Getting specific data from dynamoDB was rather difficult and we ended up doing a scan which gets all the data from the given table, saving this as an object and using this for future usage.

The charts from Google are dynamic in the sense that they automatically will resize to fit the div that they are nested into. This has resulted in some problems with having them fit into the design that we want.

Getting the Google chart to do specific things like when the user click on line, then only show the line the user clicked on. Also controlling the line individually are somethings that are hard to do with the Google Charts, cuse of this we had to change our functionalities to the restriction of the Google Charts.

4.2. Design Challenges

There were some challenges with the design process. We had difficulty with the slides on the page. We used different types of code to get the slides to work the way we wanted. The calendars were first used with regular HTML and CSS, which led to that we couldn´t style the input fields so it matched the rest of the website. We change the calendars to HTML, CSS and JavaScript so we could alter the calendars the way we wanted. Google charts had different presettings which made it difficult to make responsive and change style.

5. Teamwork

5.1. What worked

There has been a broad area of expertise in the project group. Some of the members had a good understanding of HTML and CSS while others were more comfortable with the Javascript part.

Half way through we started using a slack bot to push messages about commits to the git repo, which proved to be a good way to both let people know what was doing but also to inspire people to do more work themselves.

5.2. What did not work

Due to a low number of attendees during the weekly scrum meetings, especially early on, communication about what was needed to be done was hard. Even though we could communicate through slack and all the members of the project had access to the scrum board on Trello there was confusion regarding what the project was really about and what the end goal looked like.

5.3. Individual contribution to the project

Ludwig Wiktorsson

I have contributed to the project in a couple of ways. First of all i have been the scrum master and have tried to push the project forward. I have also together with Anton and Simon helped the project members who didn’t have that much experience with git. Code wise I have together with Simon and Anton restructured the code we got from the Lab a little bit and got the data we got into the first versions of the graphs. I have also added the Lab status to the lab tab.

Simon Rosengren

First of all I had the responsibility to set up the project on github and have together with Anton and Ludwig helped the people who were less experience with git on any problems this might have caused. The major parts code wise that I have made is the first version of the databasehandler[5] and the part that takes whatever comes from the MQTT and turns it into the correct object. I have also created the historical sound chart among others.

Anton Björkman

I have tribute to project in many ways, I've come with with many ideas and solutions to problems. Mostly I have been working with the functionally of the web application. I have code the all of the functionality of the Mashup slide, the loading function, the Humidity and Light Live Chart and I have also reworked the databasehandler so it is more effective.

Amanda Borg, Jessica Hagstedt, Alexander Liderås, Hedda Johansson, Shaya Mokarremi and Sara Ranstorp

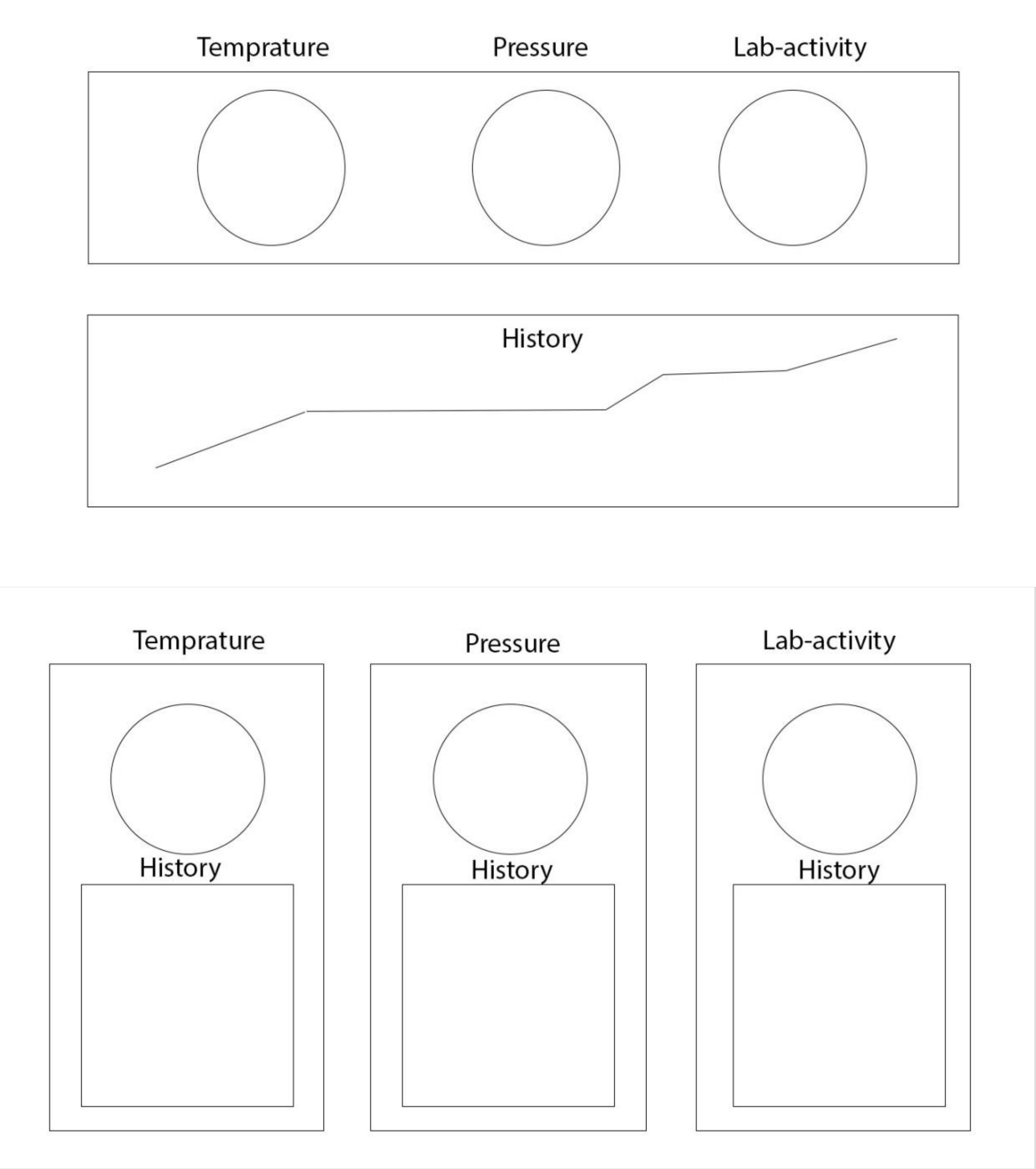
Group 2 took care of the design, the group consist of the following members: Alexander Liderås, Sara Ranstorp, Jessica Hagstedt, Shaya Mokarremi, Amanda Borg and Hedda Johansson.

We started to do some research of diagrams that you can find in Google Charts. You can find a lot of varieties of diagrams in Google Charts so we picked out few of them that were appropriate for our project. We made a Graphic profile that contained a color map and some fonts that could fit the project.

During the project we also did sketches of the design to make sure the other group agreed with our ideas.

Our first coding assignment was to divide the html code in to div-tags so it would be easier to design with css. We were also mainly in charge of the css code, as we were more familiar with this type of coding than the other group. We have used HTML, CSS and JavaScript to create the design of the page. For the layout we used flexbox and MediaQueries. The slides, buttons, calendars were also made by us. We have worked closely together in the group and made all of the decisions together to create the design and layout of the page.

[5] https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_Scan.html



Sketch-examples of the design.

We have used code examples from: https://codepen.io/artyom-ivanov/pen/ggQaLV to make the buttons on the mashup page. We have used the website: http://freefrontend.com as design inspiration.

6. SCRUM method

The project has been active for five weeks and during these weeks the scrum method was used to present our performance once a week in form of sprints. Each sprint the scrum master presented our work and went through the backlog to see which tasks were completed and which ones were not. Feedback was given and after that the group decided which tasks was necessary to the next sprint and what user story to prioritize. The tasks was distributed to the individuals in the group. At the end of the sprint the group had a discussion about what worked well during the week and what had not worked well. This was repeated each week.

6.1. What worked

When all the team members became more committed to the work the method worked well and was to good use. Instead of having all the work at once. Even though we had some problems we actually have a product that looks good and does what it’s supposed to do.

6.2. What did not work and improvements

The group had a few problems in the beginning with the method because of the absence of team members during the sprints. To much work was given to the ones that where present. The communication was also bad and the commitment was not there. This was raised during the meeting and expressed to the team members. The work improved after that and the commitment increased and almost all the team members was present during the remaining sprints. The communication also became better after that and more work was put in by all the team members.

7. Conclusions and Project Perspectives

7.1. Lessons learned

One big lesson we learned was how hard it is to work in bigger groups, especially when the group members comes from different programs with different schedules. Another thing is the value of communication. When we started to feel more comfortable with each other and started to communicate better the project moved forward much quicker and we were more effective as developers.

7.2. Next steps

If the project continued the first step would've been to update standard google graphs that we are using for light level and humidity to something more convenient. Maybe even update the graphs from google graphs to d3[6] to have more design options. Another thing to continue with is the optimization of the database. Instead of fetching the historical data as one chunk and sort it out after we could decide between which timestamps we want to fetch it. We could also add some more interactive graphs of some of the data we got. For example the number of people that are in the lab.

[6] https://d3js.org/