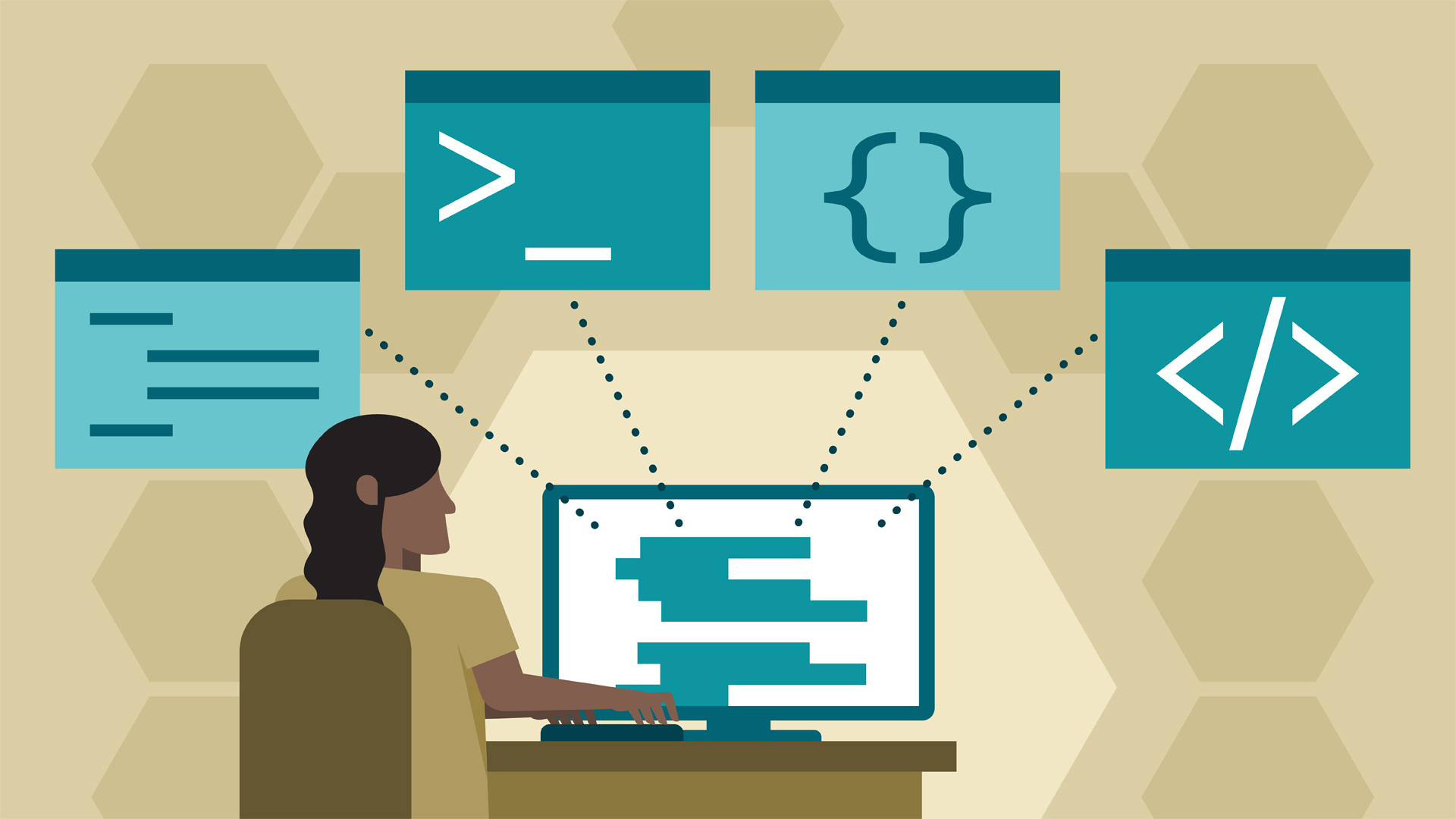
**Development process web-application**

****

**Development process Plant Patrol**

**Luuk Vlug**

**500807974**

**luuk.vlug@hva.nl**

**Plan of development**

In the first week of the project, I was still coming up with my idea and getting it accepted.

The second week I dedicated to getting familiar with programs used and getting the hang of basic html.

After that I had no real plans, as I was not expecting html to go so smoothly. During one of the classes I asked for a sensor and started coding that in Arduino. After a bit of playing around with the code and the sensor and a bit of doing research, I got the sensor to work.

Next up was displaying the sensor results on the website. This is where the problems started to occur. This step took me at least 2 to 3 weeks to get to work. I got what felt like endless amounts of errors. Once I managed to fix one error, another 10 replaced it. But luckily after a bit more research, messing around and asking around on forums, I got to work after about 2 to 3 weeks. It’s a good thing I managed to get it to work, because my motivation was pretty much nonexistent. About to give up, the solution and final result returned the motivation lost during the weeks of research and errors. After finally being able to display the moisture value on my website, I had to make it look pretty. A new chapter started.

The value was displayed on my website with just a value, no other context or information, making the random number meaningless. Next to that, I used document.write, so every 0.3 seconds a new value was pushed from the arduino code, it added a new value to the website. This created a page full of numbers, constantly adding new ones. After some more research, I fixed that problem using innerhtml, sadly there still was no context. Never before had I worked with objects or json, so this took some research as well. 2-3 days to be exact. The answer was using stringify, .replacing the “{}” with nothing and adding “Moisture: “ to it. Being able to figure all this out give me even more motivation. I decided to start working on the API.

After checking out multiple forms of weather API’s, I got to buienradar. Buienradar was able to show forecast results for every city in the Netherlands. Sadly this was all one massive json with hundreds of lines. After doing some more research I decided to try openweathermap. I followed some tutorials on Youtube and read some guides on the internet. Luckily this was not as complicated as I thought it was. I already had encountered json before when trying to display the moisture value in a proper way. This lead to me being able to at least come up with decent Google search terms. After finishing the openweathermap API and being happy with the result, I remembered I wanted a forecast API, which I forgot when implementing it. So I had a regular weather API. I tried thinking of some reason this could be relevant, but decided to change it to a forecast API. All this took was changing the .fetch API link, looking in the JS console in Chrome to find the location of the correct information and changing that in the code. This displayed (somewhat chaotically) the information I requested. Because we got extra time for the project, I decided to also focus on trying to display the forecast in a table. Using some online guides and an online tool, I was able to quit easily add this.

At this point I wanted to also do something with php, after hearing positive stories about it. I download XAMPP and Sublime and did some googling. With the help of a classmate, I was able to finish my website with php. Adding some fun features like the calibration, with some code in Arduino and some php and js, and adding a graph, I was excited to see the improvements. With 3 weeks till the deadline I decided to leave the website as is and focus on other aspects of the project. Such as the poster, physicalization and business and trends.

**Resources used**

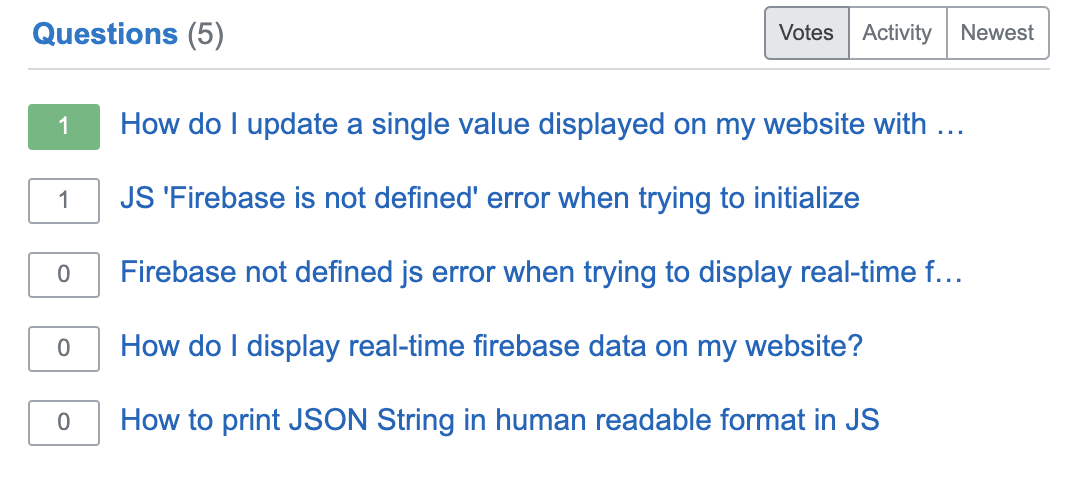
To solve the issues I encountered when displaying Firebase data on my website I used Stackoverflow. *Stackoverflow Questions*. (2020, 16 maart). Geraadpleegd op 16 maart 2020, van <https://stackoverflow.com/users/11101936/luukv19>

For an introduction to API’s I used a Youtube video. Get Set Python. (2018, 31 maart). *OpenWeatherMap API - Overview*. Geraadpleegd op 18 maart 2020, van <https://www.youtube.com/watch?v=SXsaB9TUfkk>

Also w3schools was a massive help with html. Especially adding images, as I was lost on how to do this. After using the ‘Try it yourself’ on w3schools, it became a lot clearer. w3schools. (z.d.). *HTML Images*. Geraadpleegd op 3 maart 2020, van <https://www.w3schools.com/html/html_images.asp>

In the first week of struggling with the Firebase connection and data, I used an online guide for some information. A portion of the information was outdated, but it gave me an idea on how to get to work. ElectroPeak. (2019, 18 april). *Connecting Arduino to Firebase to Send & Receive Data*. Geraadpleegd op 12 maart 2020, van https://create.arduino.cc/projecthub/electropeak/connecting-arduino-to-firebase-to-send-receive-data-cd8805

Screenshot of the question I asked on Stackoverflow:



3 of the question are regarding the Firebase issue I had with displaying the values on my website. The biggest part about asking questions here was getting some example code back with methods I had not heard of before. Such as Stringify. This gave me the possibility to do research on this.

**New knowledge and acquired skills**

This is an easy topic in my case. The only coding experience I had prior to the project was with Python. Therefore, all the information I looked up, all the code I wrote and every tool/program used was new to me. But looking back at the project, the biggest improvement I made is with html. Because it’s a little easier to pick up, and easier to distinguish, I was able to not get it mixed up with other languages. Also because for example with js, there was also information on frameworks, such as react or angular, that was confusing as well. With html I encountered no such things during my research, making every bit of information I found useful.