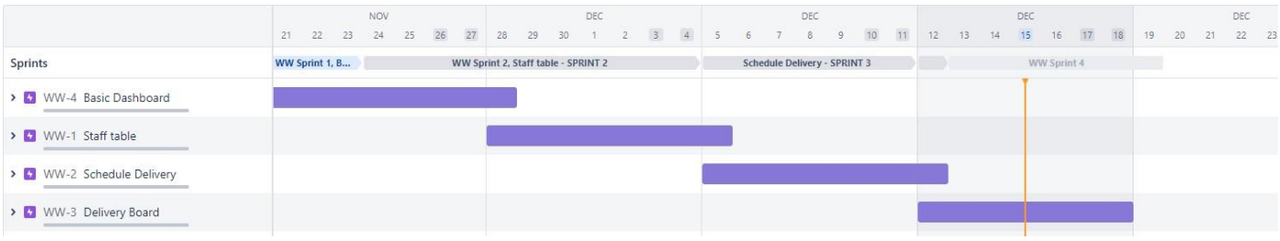
Reflection Report for SP1 – Lasse Stavland

Roadmap

I created 4 sprints, to get the workload down to bite-sized loads.

1 week per sprint, as I am fulltime and get 4 weeks to complete this project.

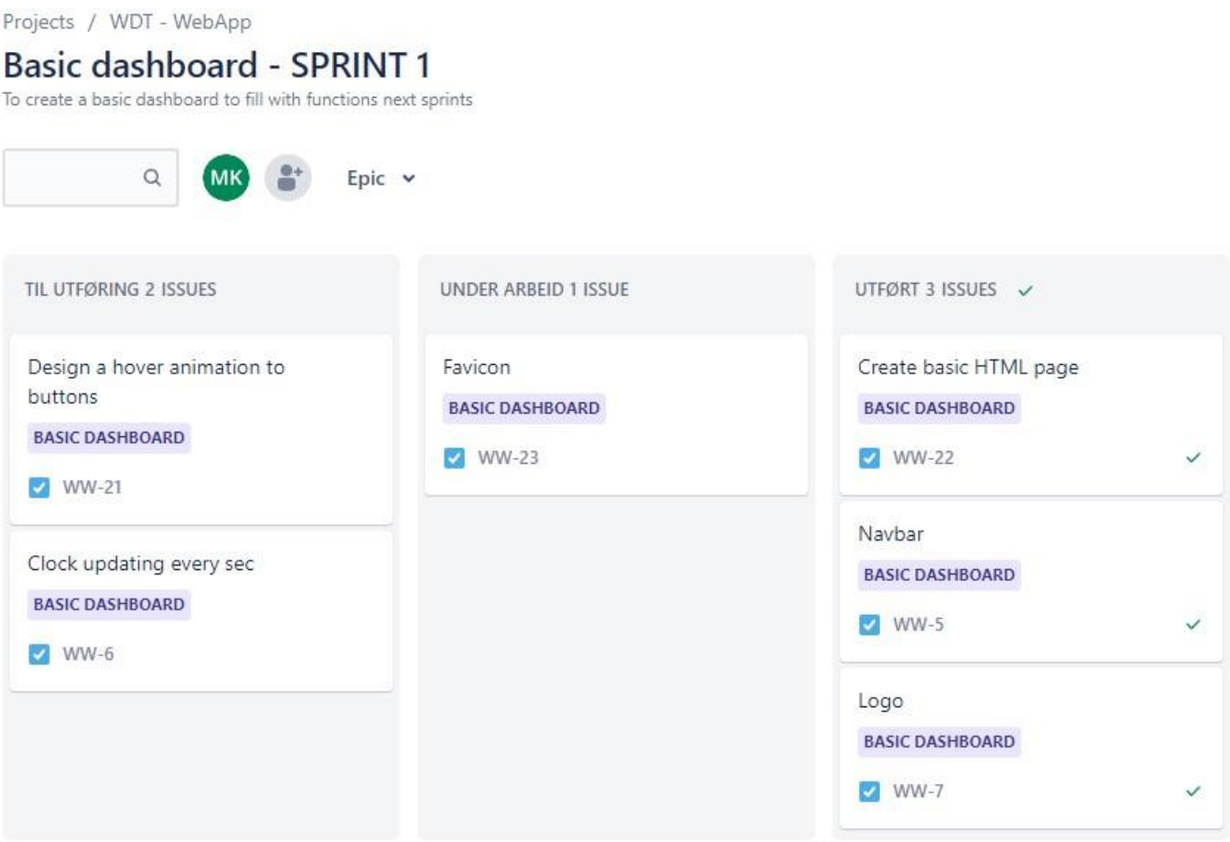


Sprint and Issues

Sprint 1:

This sprint was set up for me to get something visual to relate to when creating functions later. I created a Favicon picture and a digital clock. The navbar is not in the Mockup photo so I placed it next to the company logo. Clicking the logo or Dashboard button will update the homepage/dashboard and update the webapp with 5 new random users.

This sprint went as planned with no problems worth mentioning.



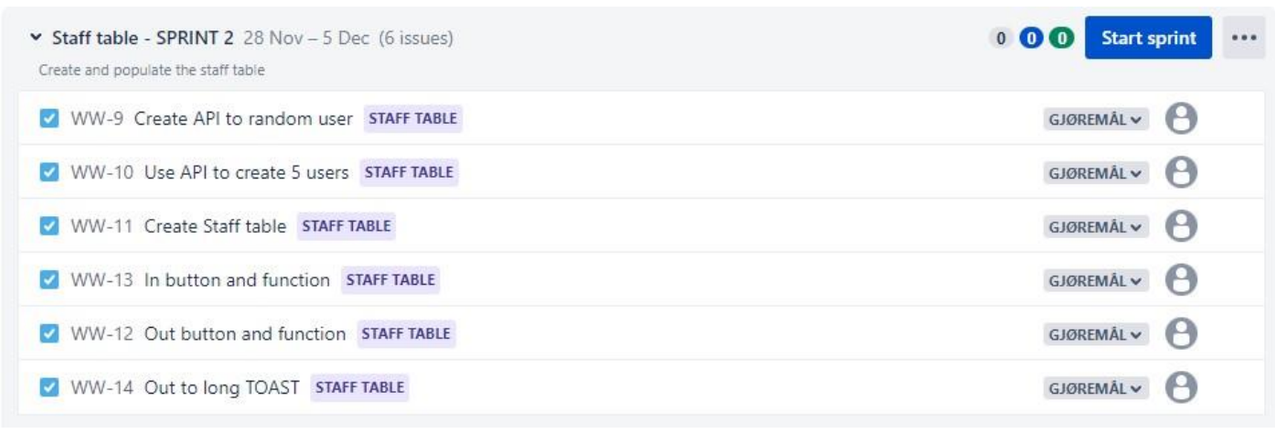
Sprint 2:

For this sprint I wanted to create and populate the “Staff” table. I used an Ajax API call to get 5 random users. I then used the data from the API call to create 5 new “StaffMember” class instances.

At the start I made the table hard coded to 5 users but felt that the table should be populated in a dynamic and scalable way, so I re-did the function that populates the table in a way that you can give it unlimited number of users from the API call, and it will work.

To get the duration for the “Staff” toast notification I prompted the user for an expected duration of absence from office and converted it to milliseconds. Then I created a function called “TimeConvert()” to take the minutes from the user prompt and convert everything that’s over 60 minutes to hour and minutes for the duration cell in the table HTML.

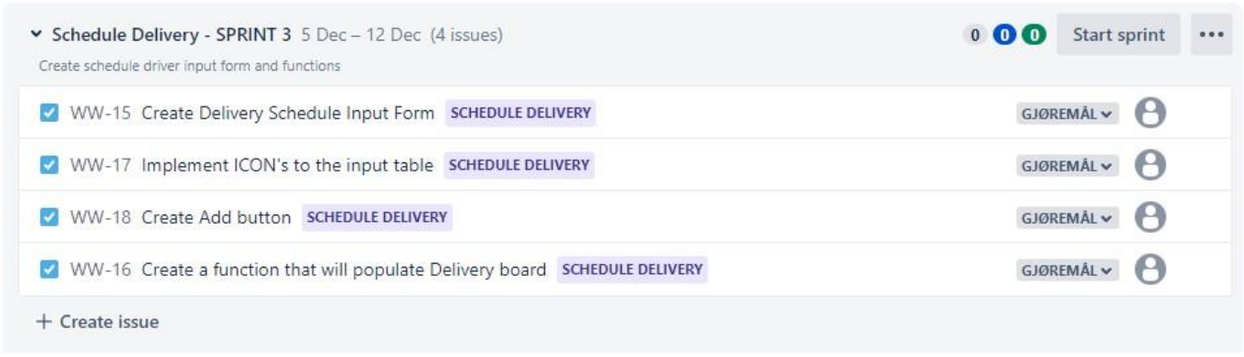
I created a function that will highlight the clicked row and bind the selected row index to a variable used for the in/Out buttons. If you click the header of the table, it will unselect/unhighlight itself.



Sprint 3:

The 3rd sprint went as planned; I created a function called “addDelivery()” that is connected to a button with onclick event. This will trigger a function that will create a new class “DeliveryDriver” with all properties provided and populate the Delivery Drivers table.

All fields must be filled, and the correct format must be provided for the class instance to even get created.



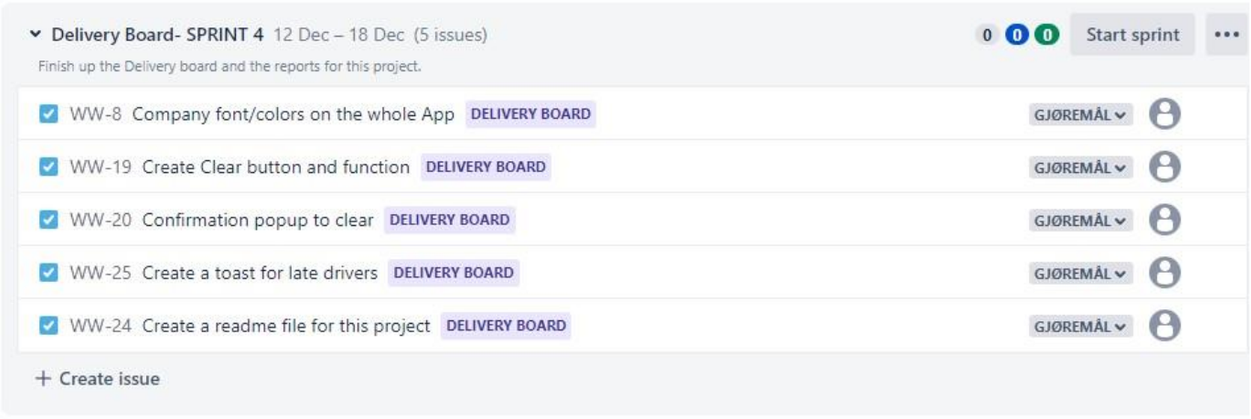
Sprint 4:

This sprint I had some struggle with clearing the toast notification for Delivery Drivers.

The toast was hard to solve because it uses the “setTimeout()” function and I needed a way to use “clearTimeout()” when pressing the “Clear” button for a specific employee. I needed to find a way to give each setTimeout an index for me to use if I want to cancel the toast.

Clearing the toast worked fine, until I deleted some of the first rows, which would change the index for the rest of the timeouts/rows.

I ended up pushing every instance of a setTimeout into an array, so I could use the index from that array.



What I learned:

I have learned that there are many ways to solve the same task, some more complicated than others. And some can be incompatible with functions that need to be implemented later. This is something me as a developer need to account for when planning.

I believe that my train of thought for this project was to focus on a function-to-function basis and I should in the future think further ahead to make sure the functions I am creating “now” will be compatible with the functions I am yet to develop so that I can avoid the mistake of having to re-do code later down the road.