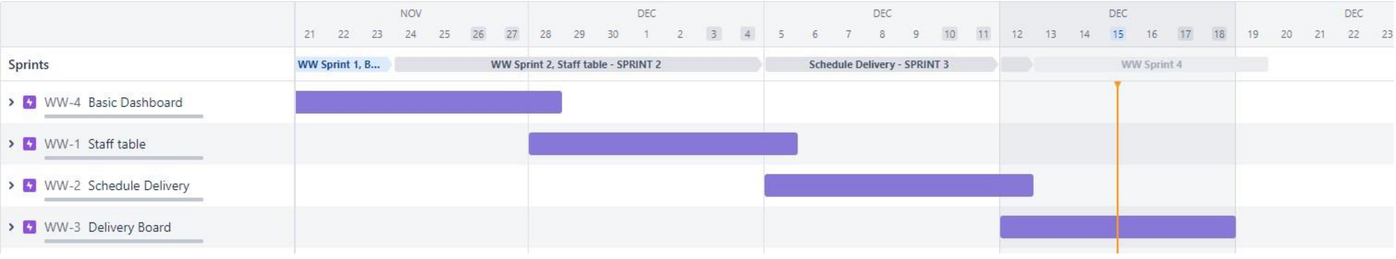
Reflection Report for SP1 – Lasse Stavland

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.



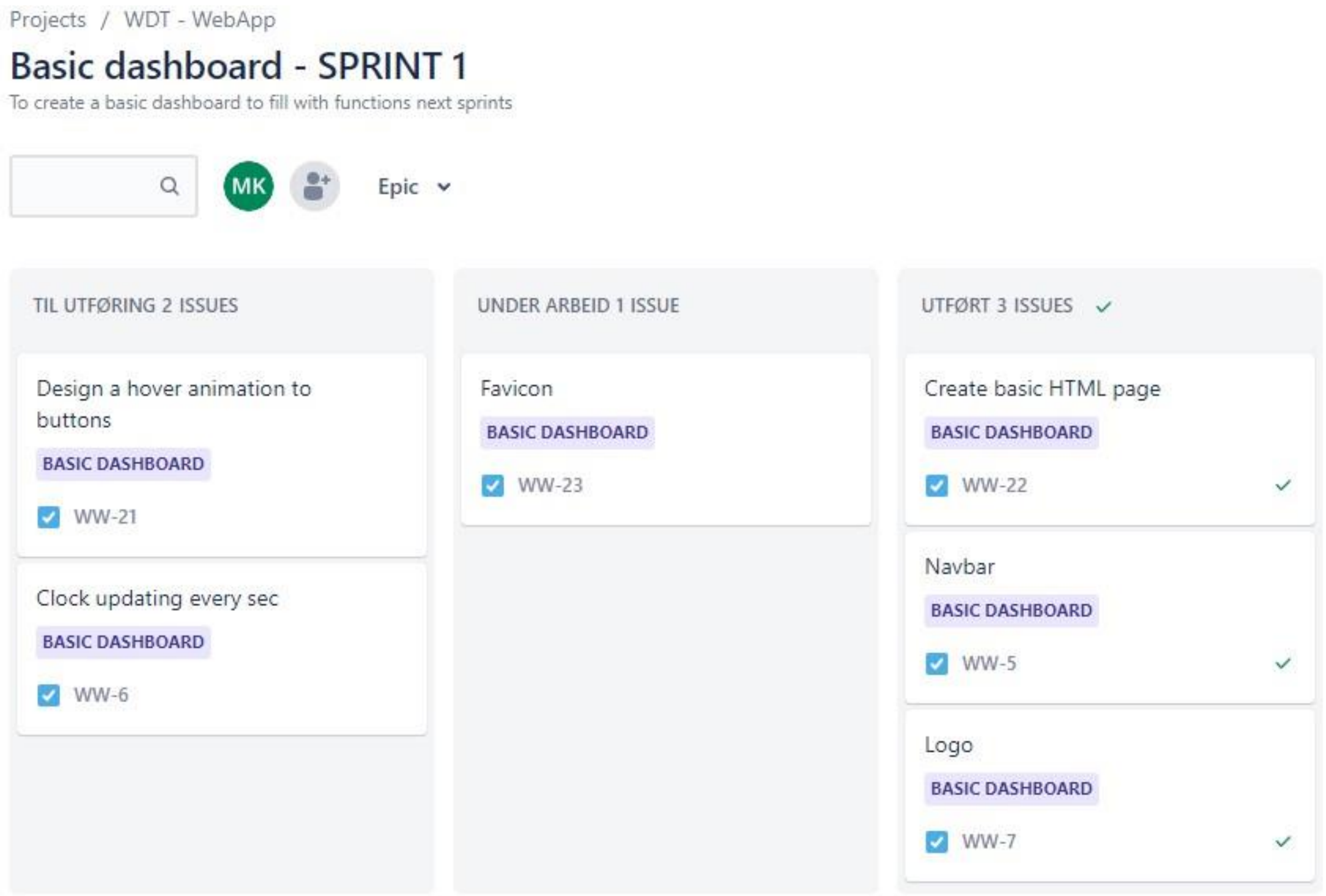
Sprint and Issues

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.



Projects / WDT - WebApp

Basic dashboard - SPRINT 1

To create a basic dashboard to fill with functions next sprints

TIL UTFØRING 2 ISSUES

Design a hover animation to buttons
BASIC DASHBOARD
☑ WW-21

Clock updating every sec
BASIC DASHBOARD
☑ WW-6

UNDER ARBEID 1 ISSUE

Favicon
BASIC DASHBOARD
☑ WW-23

UTFØRT 3 ISSUES ✓

Create basic HTML page
BASIC DASHBOARD
☑ WW-22 ✓

Navbar
BASIC DASHBOARD
☑ WW-5 ✓

Logo
BASIC DASHBOARD
☑ WW-7 ✓

Sprint 2:

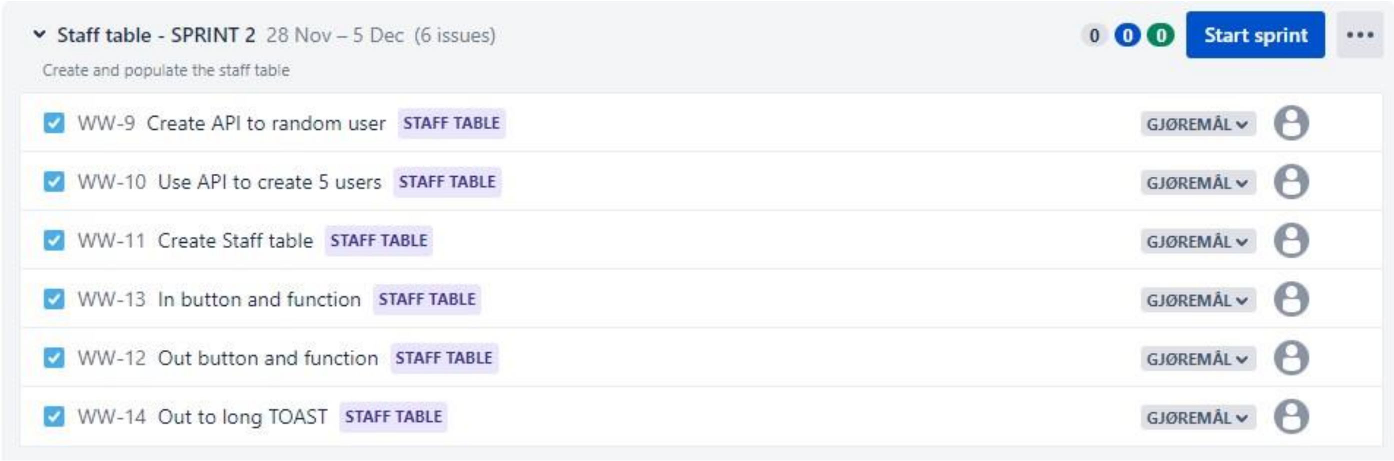
For this sprint I wanted to create and populate the “Staff” table.

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.

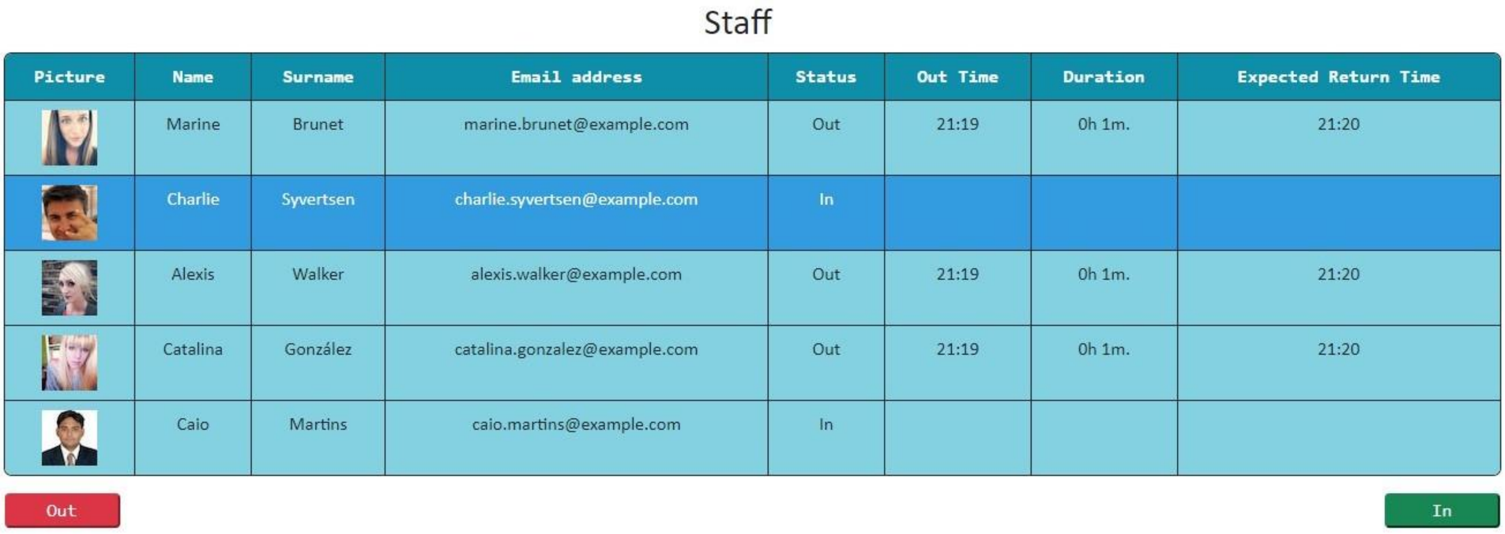


Staff table - SPRINT 2 28 Nov – 5 Dec (6 issues)

Create and populate the staff table

- ☑ WW-9 Create API to random user STAFF TABLE GJØREMÅL
- ☑ WW-10 Use API to create 5 users STAFF TABLE GJØREMÅL
- ☑ WW-11 Create Staff table STAFF TABLE GJØREMÅL
- ☑ WW-13 In button and function STAFF TABLE GJØREMÅL
- ☑ WW-12 Out button and function STAFF TABLE GJØREMÅL
- ☑ WW-14 Out to long TOAST STAFF TABLE GJØREMÅL

Finished Staff table:



Staff

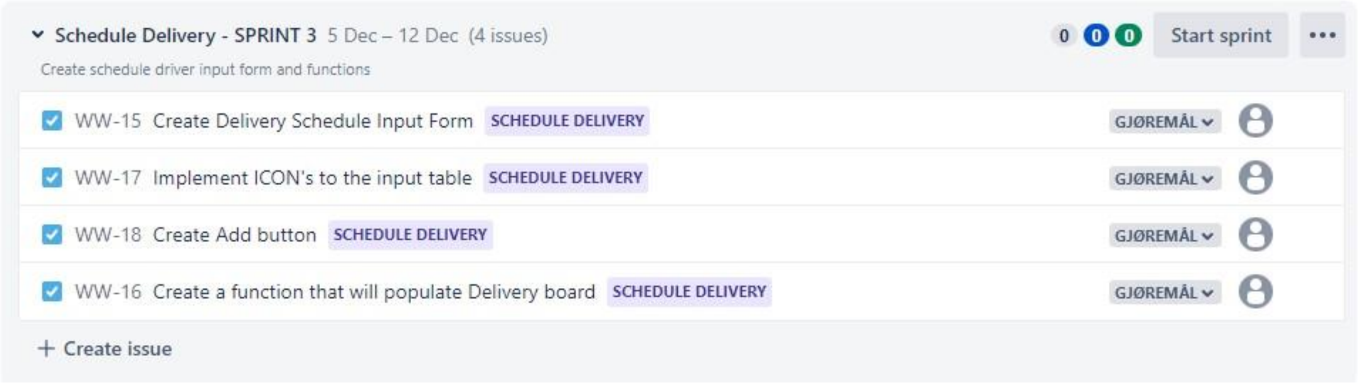
Picture	Name	Surname	Email address	Status	Out Time	Duration	Expected Return Time
	Marine	Brunet	marine.brunet@example.com	Out	21:19	0h 1m.	21:20
	Charlie	Syvertsen	charlie.syvertsen@example.com	In			
	Alexis	Walker	alexis.walker@example.com	Out	21:19	0h 1m.	21:20
	Catalina	González	catalina.gonzalez@example.com	Out	21:19	0h 1m.	21:20
	Caio	Martins	caio.martins@example.com	In			

Sprint 3:

For the 3rd sprint I created input fields in a table for the client to be able to add drivers manually.

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 in the input table 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.



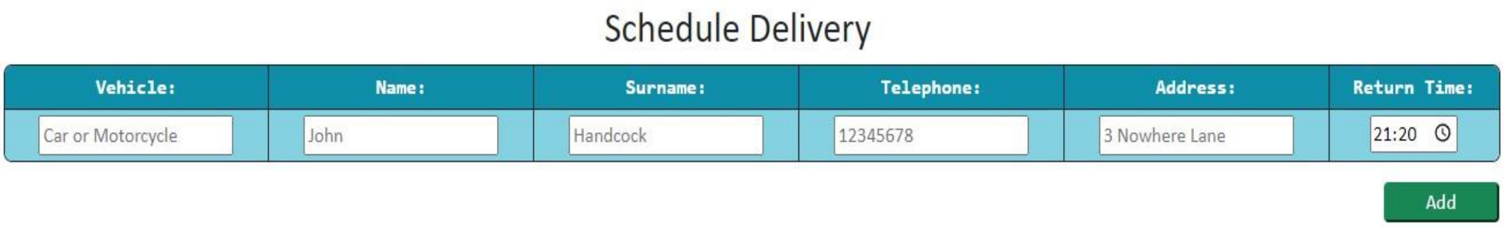
Schedule Delivery - SPRINT 3 5 Dec – 12 Dec (4 issues) 0 0 0 Start sprint •••

Create schedule driver input form and functions

☑	WW-15	Create Delivery Schedule Input Form	SCHEDULE DELIVERY	GJØREMÅL
☑	WW-17	Implement ICON's to the input table	SCHEDULE DELIVERY	GJØREMÅL
☑	WW-18	Create Add button	SCHEDULE DELIVERY	GJØREMÅL
☑	WW-16	Create a function that will populate Delivery board	SCHEDULE DELIVERY	GJØREMÅL

+ Create issue

Finished Input field – table:



Schedule Delivery

Vehicle:	Name:	Surname:	Telephone:	Address:	Return Time:
Car or Motorcycle	John	Handcock	12345678	3 Nowhere Lane	21:20

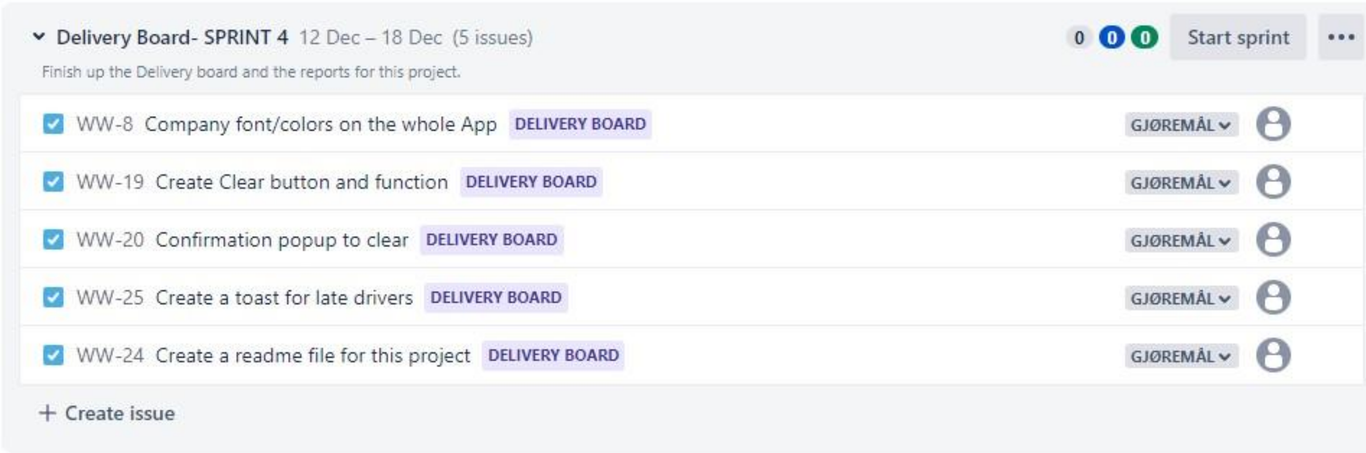
Sprint 4:

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.



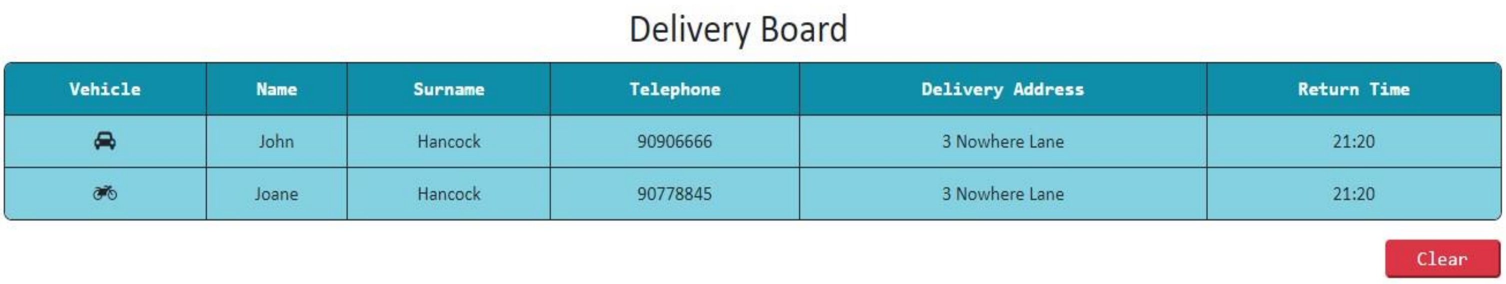
Delivery Board- SPRINT 4 12 Dec – 18 Dec (5 issues)

Finish up the Delivery board and the reports for this project.

- ☑ WW-8 Company font/colors on the whole App DELIVERY BOARD GJØREMÅL
- ☑ WW-19 Create Clear button and function DELIVERY BOARD GJØREMÅL
- ☑ WW-20 Confirmation popup to clear DELIVERY BOARD GJØREMÅL
- ☑ WW-25 Create a toast for late drivers DELIVERY BOARD GJØREMÅL
- ☑ WW-24 Create a readme file for this project DELIVERY BOARD GJØREMÅL

At the end of sprint 4 I was going to make some final changes on the Webapp, but every time I read the client documentation again, I noticed new things to implement. Like the fact that the client wanted all table borders to be rounded. That was not in the mockup, but it was in the documentation/order. I noticed that I had named the functions with different names than my client wanted. Etc. etc.

<u>Finished Delivery Board – with example data:</u>



Delivery Board

Vehicle	Name	Surname	Telephone	Delivery Address	Return Time
🚗	John	Hancock	90906666	3 Nowhere Lane	21:20
🏍	Joane	Hancock	90778845	3 Nowhere Lane	21:20

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 too much focused on a function-to-function basis and that I in the future should 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.