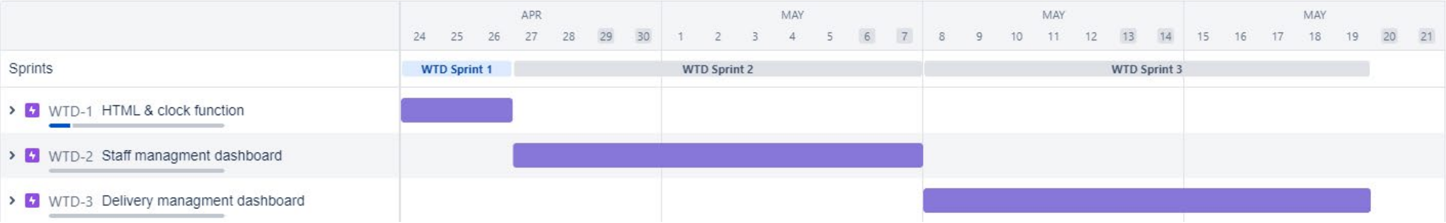
JIRA

Three sprints were organized for this project. The project kicked off on April 24th with the first sprint, which lasted only three days. This brief duration was due to the small number of issues presented in this particular epic. The epic is about the 'HTML & Clock', and it includes issues relevant to these two elements. The reason I started with this epic is because of its foundational relevance to the broader project architecture.

Sprints and roadmap:



WTD-1 HTML & clock function

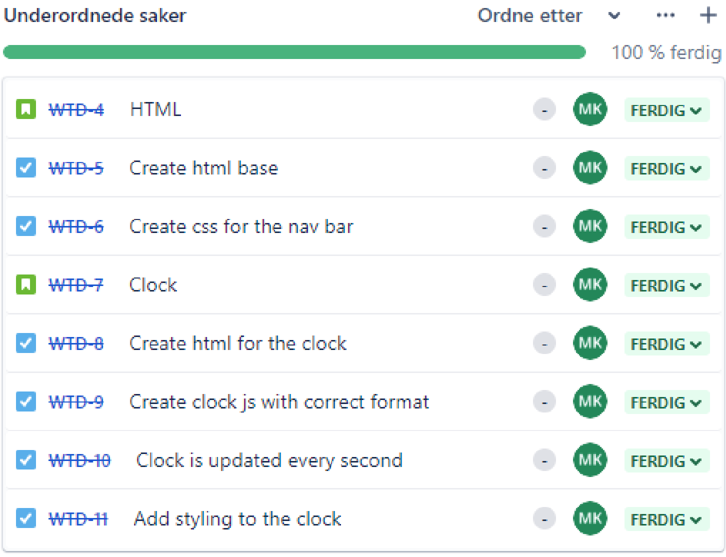
WTD-2 Staff managment dashboard

WTD-3 Delivery managment dashboard

The focus of the second sprint, which occurred from April 27th to May 7th, shifted to the 'Staff Management Board'. Hence, issues regarding this table were created. As work advanced, extra issues appeared that demanded attention, and I incorporated these into the project. I chose these issues according to the preferences and needs of WeDeliverTECH™. (see issues on the next page)

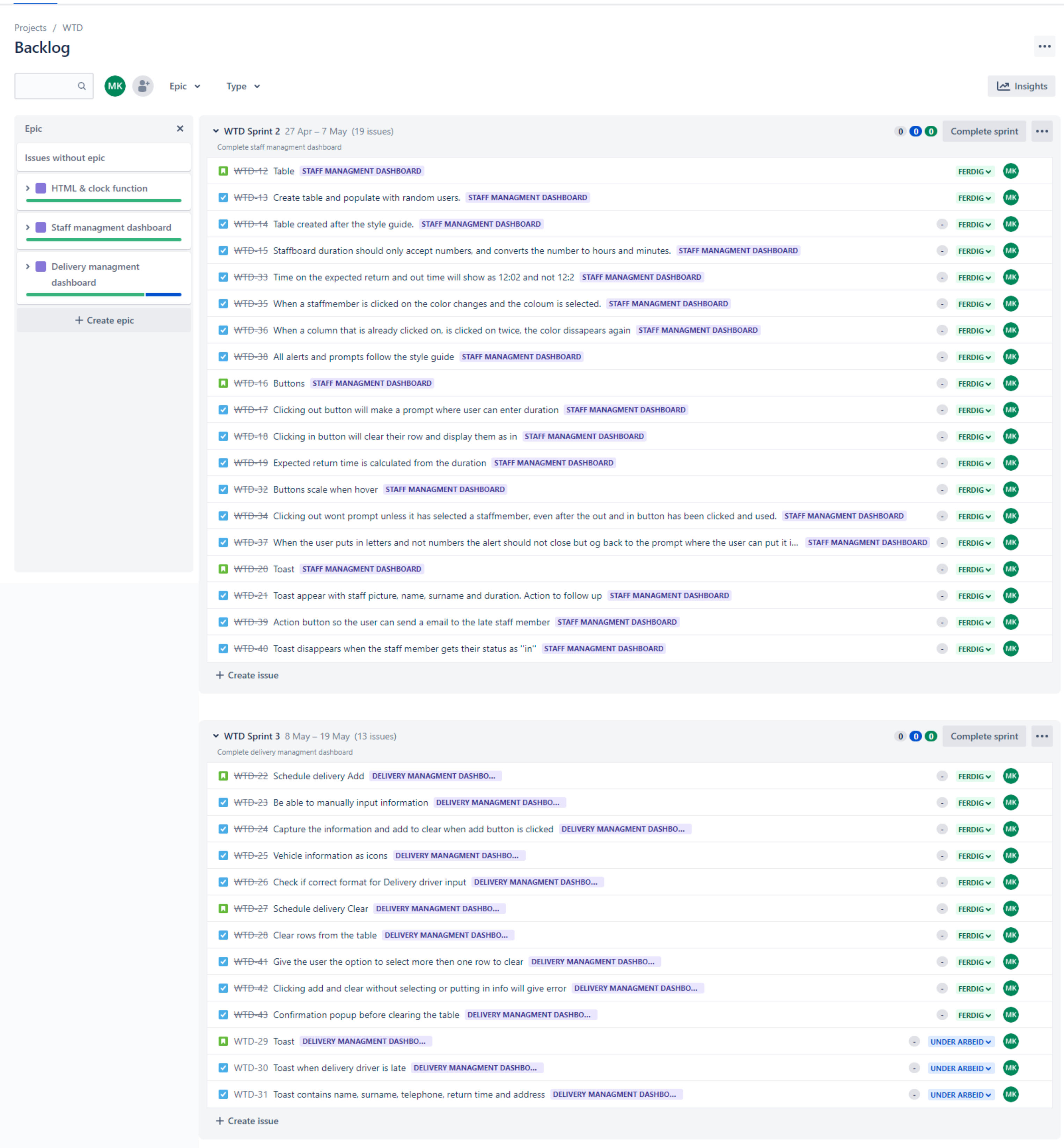
The key task during the final sprint was to establish the 'Delivery Management Board'. This phase primarily concentrated on resolving issues related to the requirements of WeDeliverTECH™, with additional ones being incorporated as they arose during the process.

Before I screenshotted the first sprint in the backlog I completed the sprint and i couldnt get it back, so the screenshot for the first sprint is from the details of the epic. I then made sure to not click on the ''complete sprint'' on the two other sprints before the end of the project (for the screenshots).



~~WTD-4~~	HTML	-	MK	FERDIG
~~WTD-5~~	Create html base	-	MK	FERDIG
~~WTD-6~~	Create css for the nav bar	-	MK	FERDIG
~~WTD-7~~	Clock	-	MK	FERDIG
~~WTD-8~~	Create html for the clock	-	MK	FERDIG
~~WTD-9~~	Create clock js with correct format	-	MK	FERDIG
~~WTD-10~~	Clock is updated every second	-	MK	FERDIG
~~WTD-11~~	Add styling to the clock	-	MK	FERDIG

Screenshot of *backlog with sprints, epics and issues:*



Projects / WTD

Backlog

MK · Epic · Type · Insights

Epic

Issues without epic

- HTML & clock function
- Staff managment dashboard
- Delivery managment dashboard

+ Create epic

WTD Sprint 2 27 Apr – 7 May (19 issues) 0 0 0 Complete sprint

Complete staff managment dashboard

~~WTD-12~~	Table	STAFF MANAGMENT DASHBOARD		FERDIG	MK
~~WTD-13~~	Create table and populate with random users.	STAFF MANAGMENT DASHBOARD		FERDIG	MK
~~WTD-14~~	Table created after the style guide.	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-15~~	Staffboard duration should only accept numbers, and converts the number to hours and minutes.	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-33~~	Time on the expected return and out time will show as 12:02 and not 12:2	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-35~~	When a staffmember is clicked on the color changes and the coloum is selected.	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-36~~	When a column that is already clicked on, is clicked on twice, the color dissapears again	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-38~~	All alerts and prompts follow the style guide	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-16~~	Buttons	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-17~~	Clicking out button will make a prompt where user can enter duration	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-18~~	Clicking in button will clear their row and display them as in	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-19~~	Expected return time is calculated from the duration	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-32~~	Buttons scale when hover	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-34~~	Clicking out wont prompt unless it has selected a staffmember, even after the out and in button has been clicked and used.	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-37~~	When the user puts in letters and not numbers the alert should not close but og back to the prompt where the user can put it i...	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-20~~	Toast	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-21~~	Toast appear with staff picture, name, surname and duration. Action to follow up	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-39~~	Action button so the user can send a email to the late staff member	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK
~~WTD-40~~	Toast disappears when the staff member gets their status as "in"	STAFF MANAGMENT DASHBOARD	-	FERDIG	MK

+ Create issue

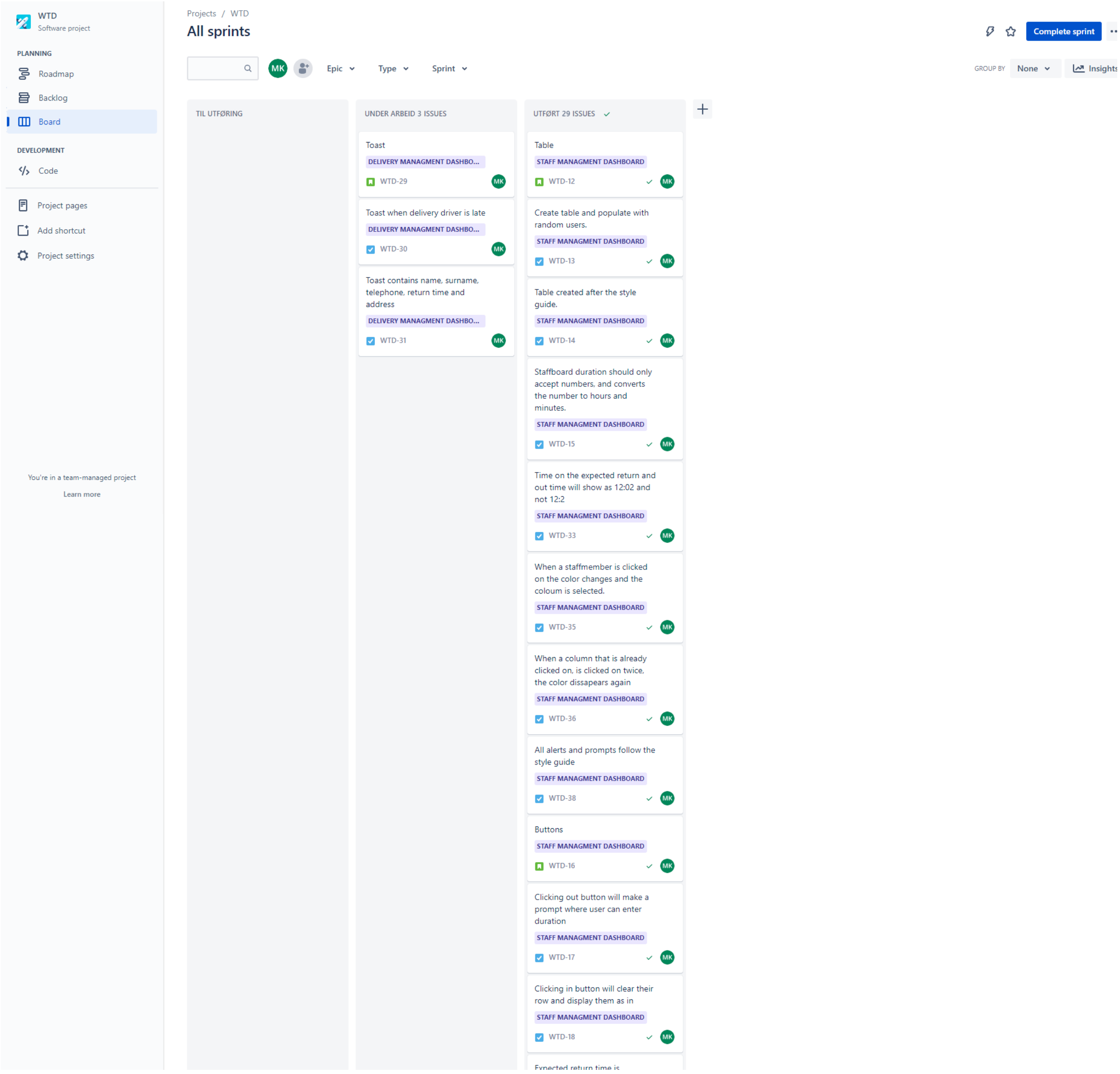
WTD Sprint 3 8 May – 19 May (13 issues) 0 0 0 Complete sprint

Complete delivery managment dashboard

~~WTD-22~~	Schedule delivery Add	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-23~~	Be able to manually input information	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-24~~	Capture the information and add to clear when add button is clicked	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-25~~	Vehicle information as icons	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-26~~	Check if correct format for Delivery driver input	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-27~~	Schedule delivery Clear	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-28~~	Clear rows from the table	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-41~~	Give the user the option to select more then one row to clear	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-42~~	Clicking add and clear without selecting or putting in info will give error	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
~~WTD-43~~	Confirmation popup before clearing the table	DELIVERY MANAGMENT DASHBO...	-	FERDIG	MK
WTD-29	Toast	DELIVERY MANAGMENT DASHBO...	-	UNDER ARBEID	MK
WTD-30	Toast when delivery driver is late	DELIVERY MANAGMENT DASHBO...	-	UNDER ARBEID	MK
WTD-31	Toast contains name, surname, telephone, return time and address	DELIVERY MANAGMENT DASHBO...	-	UNDER ARBEID	MK

+ Create issue

A cutout of the board:



TIL UTFØRING

UNDER ARBEID 3 ISSUES

Toast
DELIVERY MANAGMENT DASHBO...
WTD-29 MK

Toast when delivery driver is late
DELIVERY MANAGMENT DASHBO...
☑ WTD-30 MK

Toast contains name, surname, telephone, return time and address
DELIVERY MANAGMENT DASHBO...
☑ WTD-31 MK

UTFØRT 29 ISSUES ✓

Table
STAFF MANAGMENT DASHBOARD
WTD-12 ✓ MK

Create table and populate with random users.
STAFF MANAGMENT DASHBOARD
☑ WTD-13 ✓ MK

Table created after the style guide.
STAFF MANAGMENT DASHBOARD
☑ WTD-14 ✓ MK

Staffboard duration should only accept numbers, and converts the number to hours and minutes.
STAFF MANAGMENT DASHBOARD
☑ WTD-15 ✓ MK

Time on the expected return and out time will show as 12:02 and not 12:2
STAFF MANAGMENT DASHBOARD
☑ WTD-33 ✓ MK

When a staffmember is clicked on the color changes and the coloum is selected.
STAFF MANAGMENT DASHBOARD
☑ WTD-35 ✓ MK

When a column that is already clicked on, is clicked on twice, the color dissapears again
STAFF MANAGMENT DASHBOARD
☑ WTD-36 ✓ MK

All alerts and prompts follow the style guide
STAFF MANAGMENT DASHBOARD
☑ WTD-38 ✓ MK

Buttons
STAFF MANAGMENT DASHBOARD
WTD-16 ✓ MK

Clicking out button will make a prompt where user can enter duration
STAFF MANAGMENT DASHBOARD
☑ WTD-17 ✓ MK

Clicking in button will clear their row and display them as in
STAFF MANAGMENT DASHBOARD
☑ WTD-18 ✓ MK

Expected return time is

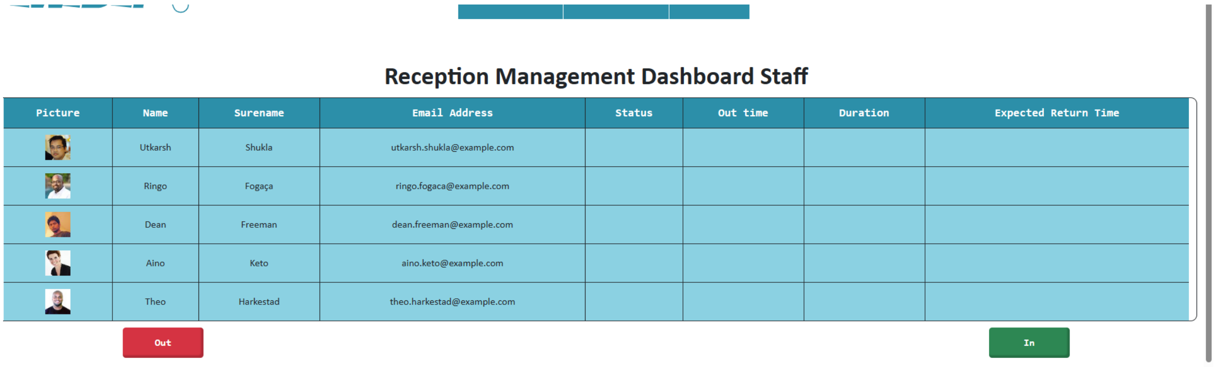
Reflection report

At the start of this project, I established four goals for myself. *Firstly*, I aimed to adhere to the project plan created in Jira and follow a structured approach throughout the work. *Secondly*, I set out to complete all tasks and issues to the best of my ability and not give up when encountering obstacles. *Thirdly*, I aimed to gain a thorough understanding of all the code snippets that i would copy and avoid copying and pasting code without comprehending how it worked. My *fourth* and last goal was to finish the project and present a high-quality web application that met WeDeliverTECH™'s requirements.

While working on the code, I faced several challenges, one of which was ensuring that all dialog boxes conformed to the style guide. Initially, i used the alert or prompt functions to create the dialog boxes. However, I realized that I could not modify their appearance. Therefore, I devised a solution to build my own dialog containers directly into JavaScript, which I could easily add CSS to. This approach resulted in longer code, but the dialog boxes matched the style guide perfectly.

I also attempted to make the dialog boxes using HTML to shorten the JavaScript code. However, this approach caused several problems in relation to CSS and the boxes' functionality, so I decided to abandon it and use the initial approach that I had tried out.

I encountered another problem during my project when I added the bootstrap link (for the toast) to the HTML file, and it took over the table, overriding my CSS. Initially, I searched for possible solutions and found some helpful suggestions. One of them was to ensure that the CSS link was placed under the bootstrap link in the HTML file. However, this did not entirely solve the problem.



Reception Management Dashboard Staff

Picture	Name	Surename	Email Address	Status	Out time	Duration	Expected Return Time
	Utkarsh	Shukla	utkarsh.shukla@example.com				
	Ringo	Fogaça	ringo.fogaca@example.com				
	Dean	Freeman	dean.freeman@example.com				
	Aino	Keto	aino.keto@example.com				
	Theo	Harkestad	theo.harkestad@example.com				

I also tried to download a customized version of bootstrap but that didn't work either. Finally, I resorted to using the "!important" attribute on the necessary CSS lines that were overwritten, as I had previously used this technique and it had resolved a similar problem. Finally, this solution resolved the issue, and my CSS worked as expected.

Another problem i faced was with the toast notification. After closing the toast notification for the first person who was late, it did not appear again when a new person was late, which would not be helpful for the user of the dashboard. To resolve this issue, I spent considerable time examining the code to determine what might be causing the problem. Eventually, I discovered that I had used ".hide" instead of ".dispose," which solved this particular issue. However, a new problem emerged: the toast notification would not disappear until the person had gotten their status from out to in again.

I spent a lot of time trying to understand the issue, and in the process, I introduced several modifications to the code. First, I added a new Boolean property 'showtoast' to the class, which is initialized as 'true'. This helps control whether a toast notification should be shown for a staff member or not. Then, I simplified the 'staffMemberIsLate()' function to directly return the result of the comparison, also checking the 'showtoast' flag. If a staff member is late and 'showtoast' is 'true', it will return 'true', otherwise 'false'.

In the 'checkStaffMembersExpectedReturnTime()' function, when a staff member is confirmed late and a toast notification is shown, the 'showtoast' is immediately set to 'false'. This adjustment ensures that the toast notification won't be shown again for the same staff member until their 'showtoast' flag is reset to 'true'. Moreover, in the event handler for the 'take action' button, the toast is programmatically hidden once clicked. This way, the toast stays visible until the user decides to interact with it. Also in the staffIn function, showtoast is set back to true. This means if the staff member becomes late again in the future, the toast notification can show again.

Finally, I set 'autohide' to 'false' when initializing the toast notifications. This guarantees that the toast notification will stay on screen until it's explicitly closed, thus addressing the core issue I had encountered. The implementation of these changes ensured that a toast notification for a particular late staff member was shown only once, and stayed visible until manually dismissed.

Overall, I followed the plan in Jira, and even though there was a bit of back and forth with a couple of issues that arose during the development of the project, the plan was overall followed. I have also not given up when encontering a problem but taken the time to understand the root cause of the problem, and then researching different solutions. For my third goal, I have also gotten a understanding of every code snippet I have copied. And for my fourth goal, to finish the project and present a high-quality web application that met WeDeliverTECH™'s requirements, I have with staying stright on my third goal, been able to deliver the web application.

In summary, while there may be additional details to look at again, I would say that I feel confident in the work I have accomplished and believe that I have achieved my goals for this project.