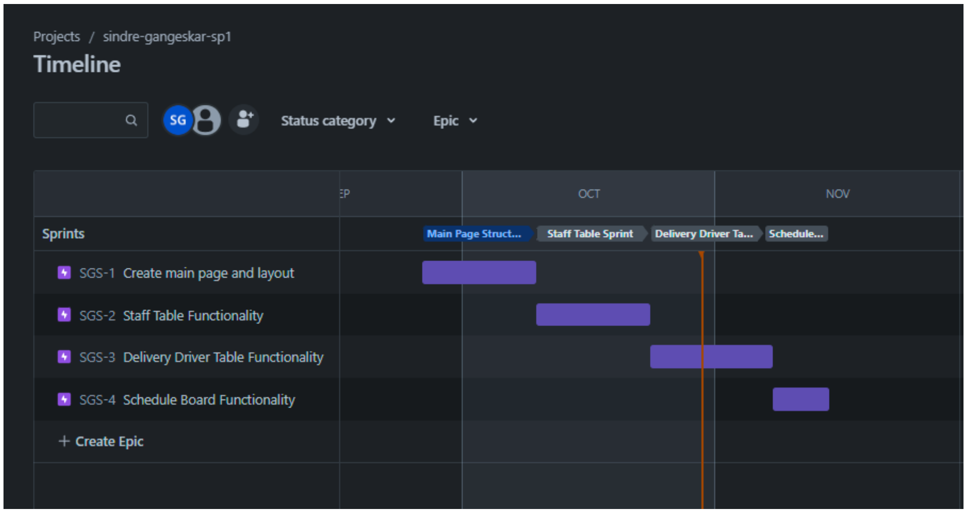
Reflection Report – Sindre Gangeskar
“Reception Management Dashboard”

Getting started

With the help of Jira, it was easy and fun to plan a bit ahead, which helped me greatly. I started setting up the backlog bit by bit with the tasks I thought would be fitting as a start for the project, as well other tasks which would have to be dealt with later in the life cycle of the project.

(Full overview of backlog at bottom page)

In the end I ended up with this timeline:



Projects / sindre-gangeskar-sp1

Timeline

SGS-1 Create main page and layout

SGS-2 Staff Table Functionality

SGS-3 Delivery Driver Table Functionality

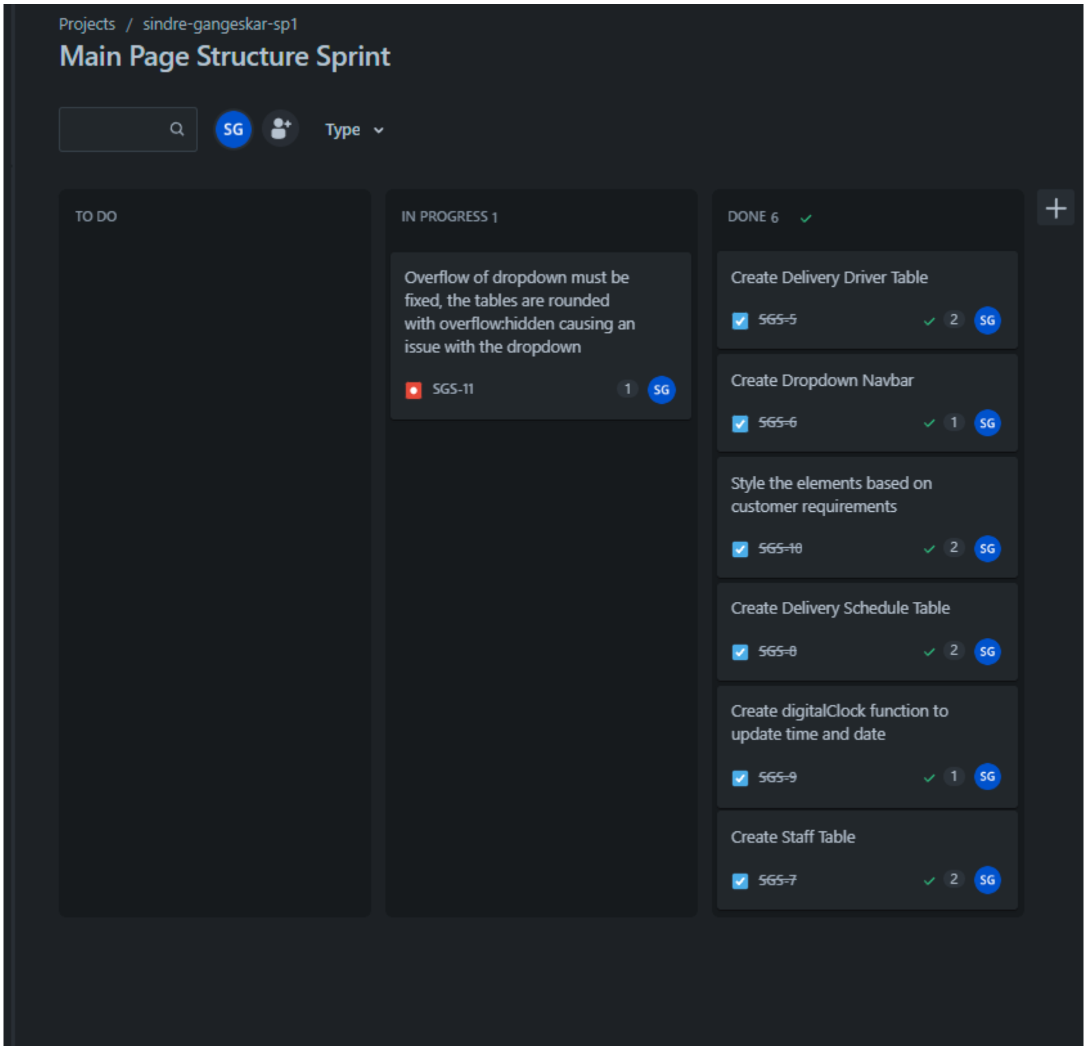
SGS-4 Schedule Board Functionality

+ Create Epic

Looking at the **Main Page Structure Sprint** which I started with gave me a good visualization as to how the html page would be structured.

I started structuring the HTML’s layout following the tasks I had set up in Jira for this sprint and applying the styling required based on the customer’s requirements, giving me the foundation and structure I needed to move on.

I had one bug at the end of the sprint.



Projects / sindre-gangeskar-sp1

Main Page Structure Sprint

SG Type

TO DO

IN PROGRESS 1

Overflow of dropdown must be fixed, the tables are rounded with overflow:hidden causing an issue with the dropdown

SGS-11 1 SG

DONE 6 ✓

Create Delivery Driver Table

☑ ~~SGS-5~~ ✓ 2 SG

Create Dropdown Navbar

☑ ~~SGS-6~~ ✓ 1 SG

Style the elements based on customer requirements

☑ ~~SGS-10~~ ✓ 2 SG

Create Delivery Schedule Table

☑ ~~SGS-8~~ ✓ 2 SG

Create digitalClock function to update time and date

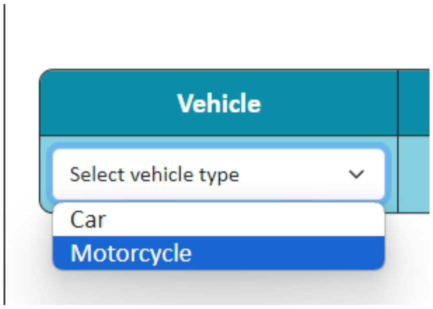
☑ ~~SGS-9~~ ✓ 1 SG

Create Staff Table

☑ ~~SGS-7~~ ✓ 2 SG

(All bugs moved to Done after screenshots)

It was a simple fix, the dropdown in the schedule delivery table was affected by its parent's overflow style property, which would make the dropdown options stay hidden within the parent table's boundaries. Tapping away in CSS I concluded that giving the dropdown element a position of absolute would ignore its parent's overflow: hidden property, leading to a fully working dropdown.



Staff Table Functionality

With a foundation established, I initiated the **Staff Table Functionality** sprint.

The process of generating the users and appending to the respective table were straightforward in the end, I wrapped each generated user data in a row and appended the rows to the staff table with the user's class data in the respective positions.

With that in place I continued my work on the clocking in and out functions. I stored the generated users in a nested array where the first array would be for the users, and the second array would be the generated toasts for the respective user that was generated.

'Clock-in' selects the row's index, retrieves data and updates the staff member's class status to "In" and updates the html counterpart with the class properties and with a brief visual confirmation of a successful clock-in.

A similar approach was made to clocking out. A modal will appear asking for a max number of 1380 (23hrs) of a clock-out value, go through the validation process with regex validation for the selected user's row and assign the user's class properties the status of "Out" and the expected return time in month, day, hour and minute with a clocking date getting the current time in month, day, hour and minute as well.

Going beyond 1380 in value or giving an input that's not a number will display an error above the input with a fading animation and red text stating that only numbers are expected, resulting in nothing else happening.

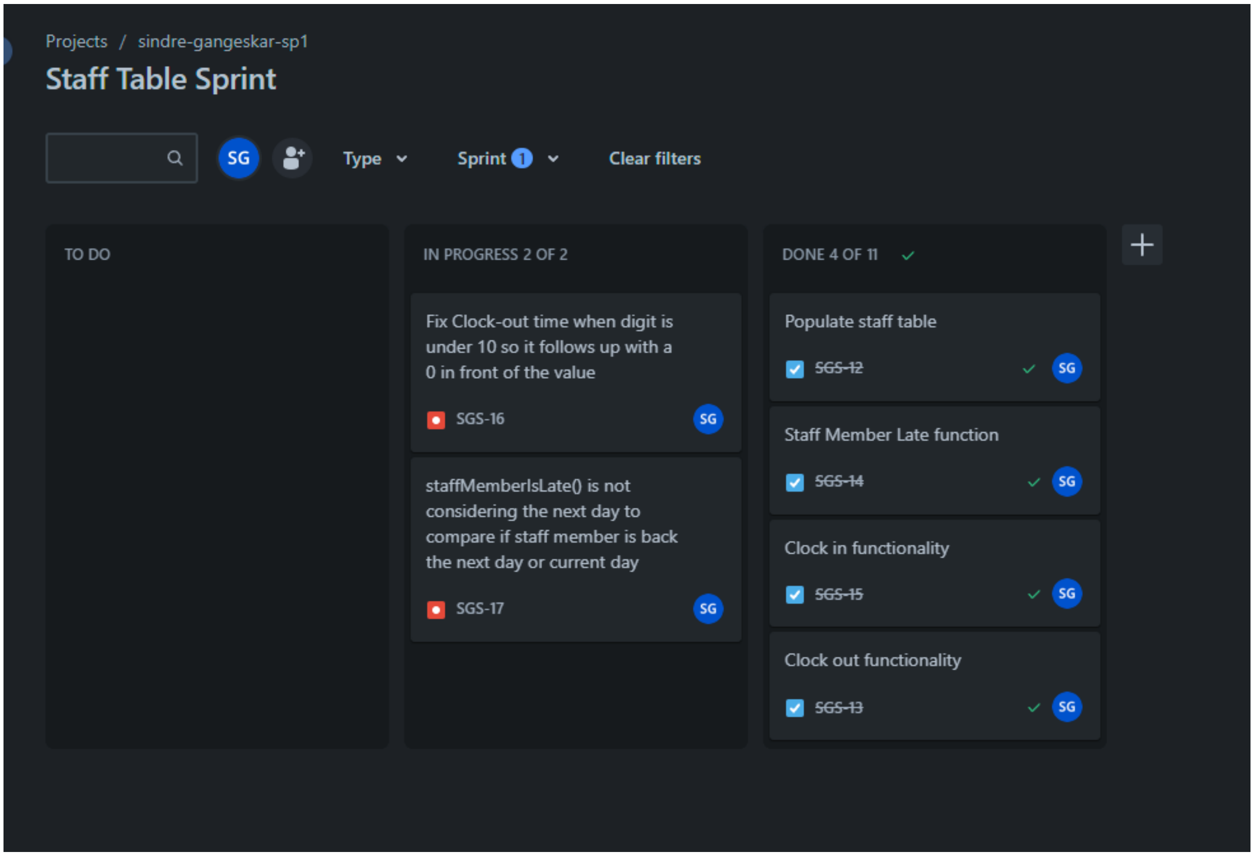
Upon successful validation I decided to give a visual confirmation to the user that was just clocked out with a background color of red as well that fades away after a short while, similar to clocking in.

The only challenge I faced during the creation of the validation was the usage of Regex, which can be a complicated form of validation but a much more secure one. Fiddling with it for a while I managed to get to a validation phase I found to be acceptable with no loopholes I could find, as things were rather basic in nature, and very fun to play around with.

I created a function which would calculate the return time for both the delivery drivers, and the staff members, promoting modularity. In this function, I made the hours and minutes roll to stay within the range of how we keep time. This was a challenge as calculating time and converting hours and minutes could've been quite complex, but researching how time is converted from minutes to hours and vice versa, I was able to find good use of the modulo operator.

For extra polish I made ternary operations for the time strings showing singular or plural of hours and minutes depending on the values given, e.g.: "1 hr." "2 mins." In duration. Fixing one of the bugs I assigned to this sprint, as seen below.

The other bug was the rollover, which was a simple if statement correction, getting the day and the return day, hour and minute, and if the clock's minute exceeded the user's return date, it'd call staffIsLate() and displaying the toast for the respective user.



Projects / sindre-gangeskar-sp1

Staff Table Sprint

TO DO

IN PROGRESS 2 OF 2

- Fix Clock-out time when digit is under 10 so it follows up with a 0 in front of the value
  🟥 SGS-16 SG
- staffMemberIsLate() is not considering the next day to compare if staff member is back the next day or current day
  🟥 SGS-17 SG

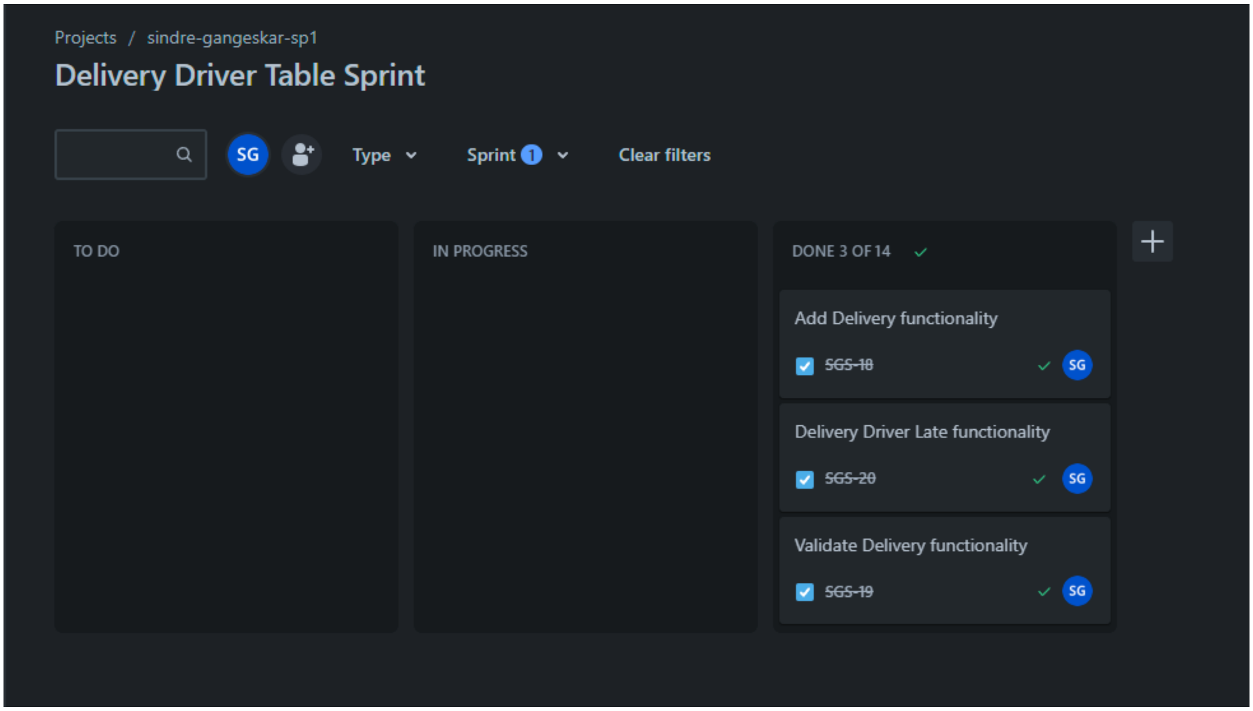
DONE 4 OF 11 ✓

- Populate staff table
  ☑ ~~SGS-12~~ ✓ SG
- Staff Member Late function
  ☑ ~~SGS-14~~ ✓ SG
- Clock in functionality
  ☑ ~~SGS-15~~ ✓ SG
- Clock out functionality
  ☑ ~~SGS-13~~ ✓ SG

Delivery Driver Table Sprint

Since most calculations and functions had already been created for the staff table sprint, it was straightforward creating rows for the delivery drivers, using a similar approach I validated the different fields using regex, and if the telephone field was less than 7 digits, it'd pop a modal stating the number needs to be at least 7 digits. If other fields were incorrect, I made the modal display all the different values that were incorrect via an error string and a loop which += to the error string per invalid value.

The challenge was to get the return time to roll over to the next day if the totalMinutes were negative by creating an if statement (totalMinutes < 0) totalMinutes += 24 * 60 which took some time to figure out.



Projects / sindre-gangeskar-sp1

Delivery Driver Table Sprint

TO DO

IN PROGRESS

DONE 3 OF 14 ✓

Add Delivery functionality

☑ ~~SGS-18~~ ✓ SG

Delivery Driver Late functionality

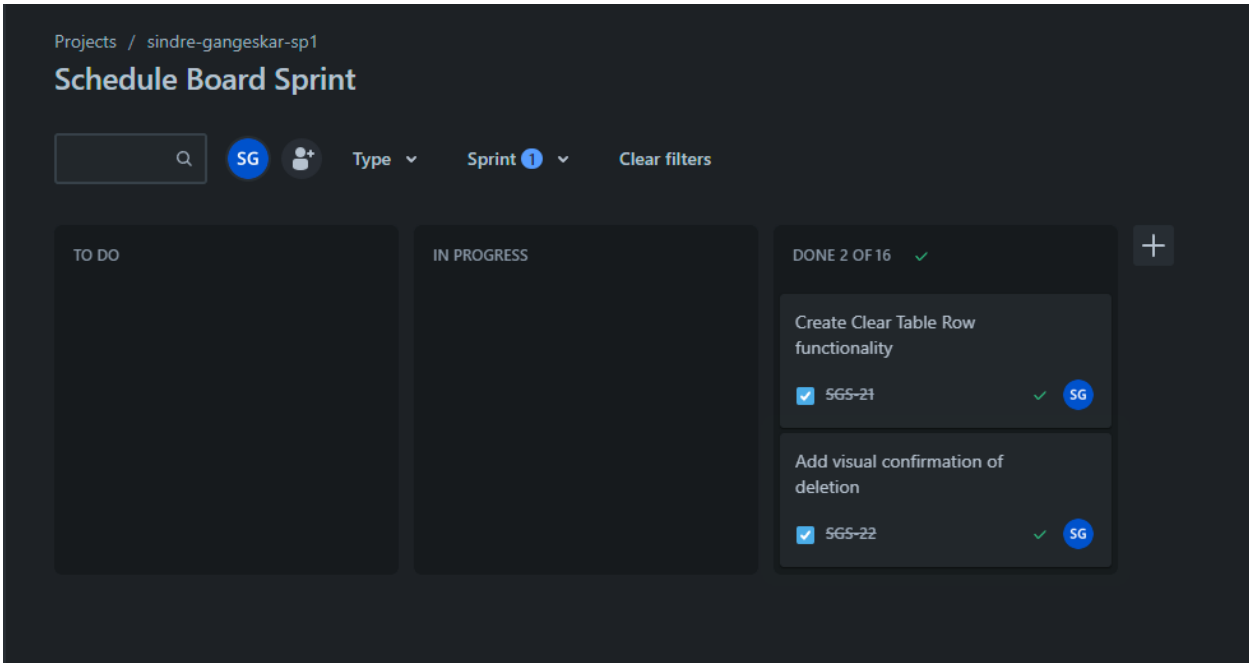
☑ ~~SGS-20~~ ✓ SG

Validate Delivery functionality

☑ ~~SGS-19~~ ✓ SG

Delivery Board Schedule Sprint

In this sprint I had to create a clear row functionality, best way I could think of was to create a modal, which upon confirmation deletes the selected row upon accept, this way I could use the index of the row to delete the selected row and remove the delivery driver from the array.



Projects / sindre-gangeskar-sp1

Schedule Board Sprint

Create Clear Table Row functionality

☑ ~~SGS-21~~ ✓ SG

Add visual confirmation of deletion

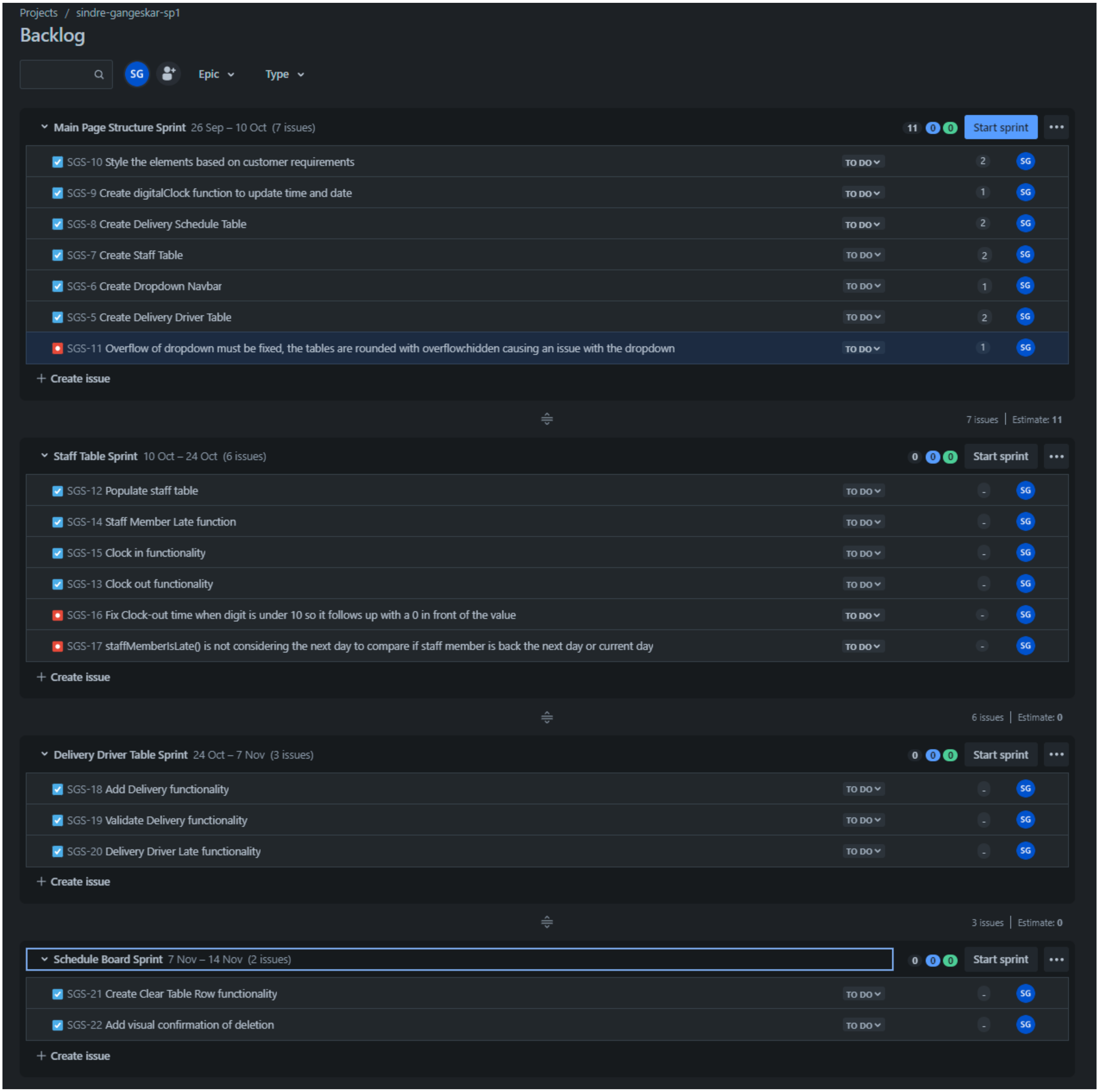
☑ ~~SGS-22~~ ✓ SG

(Open console to see some useful logging when creating drivers and clocking staff)

Conclusion

While fun, some challenges were faced. It was a good learning experience, including the fail-fast SCRUM mindset and using Jira gave me a perspective on how useful these tools really are.

Backlog



Projects / sindre-gangeskar-sp1

Backlog

SG Epic ⌄ Type ⌄

Main Page Structure Sprint 26 Sep – 10 Oct (7 issues) 11 0 0 Start sprint •••

☑ SGS-10 Style the elements based on customer requirements	TO DO ⌄	2	SG
☑ SGS-9 Create digitalClock function to update time and date	TO DO ⌄	1	SG
☑ SGS-8 Create Delivery Schedule Table	TO DO ⌄	2	SG
☑ SGS-7 Create Staff Table	TO DO ⌄	2	SG
☑ SGS-6 Create Dropdown Navbar	TO DO ⌄	1	SG
☑ SGS-5 Create Delivery Driver Table	TO DO ⌄	2	SG
🟥 SGS-11 Overflow of dropdown must be fixed, the tables are rounded with overflow:hidden causing an issue with the dropdown	TO DO ⌄	1	SG

+ Create issue

7 issues | Estimate: 11

Staff Table Sprint 10 Oct – 24 Oct (6 issues) 0 0 0 Start sprint •••

☑ SGS-12 Populate staff table	TO DO ⌄	-	SG
☑ SGS-14 Staff Member Late function	TO DO ⌄	-	SG
☑ SGS-15 Clock in functionality	TO DO ⌄	-	SG
☑ SGS-13 Clock out functionality	TO DO ⌄	-	SG
🟥 SGS-16 Fix Clock-out time when digit is under 10 so it follows up with a 0 in front of the value	TO DO ⌄	-	SG
🟥 SGS-17 staffMemberIsLate() is not considering the next day to compare if staff member is back the next day or current day	TO DO ⌄	-	SG

+ Create issue

6 issues | Estimate: 0

Delivery Driver Table Sprint 24 Oct – 7 Nov (3 issues) 0 0 0 Start sprint •••

☑ SGS-18 Add Delivery functionality	TO DO ⌄	-	SG
☑ SGS-19 Validate Delivery functionality	TO DO ⌄	-	SG
☑ SGS-20 Delivery Driver Late functionality	TO DO ⌄	-	SG

+ Create issue

3 issues | Estimate: 0

Schedule Board Sprint 7 Nov – 14 Nov (2 issues) 0 0 0 Start sprint •••

☑ SGS-21 Create Clear Table Row functionality	TO DO ⌄	-	SG
☑ SGS-22 Add visual confirmation of deletion	TO DO ⌄	-	SG

+ Create issue