



BED 2022 - Semester Project

SYSTEM WDT

Dennis Lars William Hagström



Table of Contents

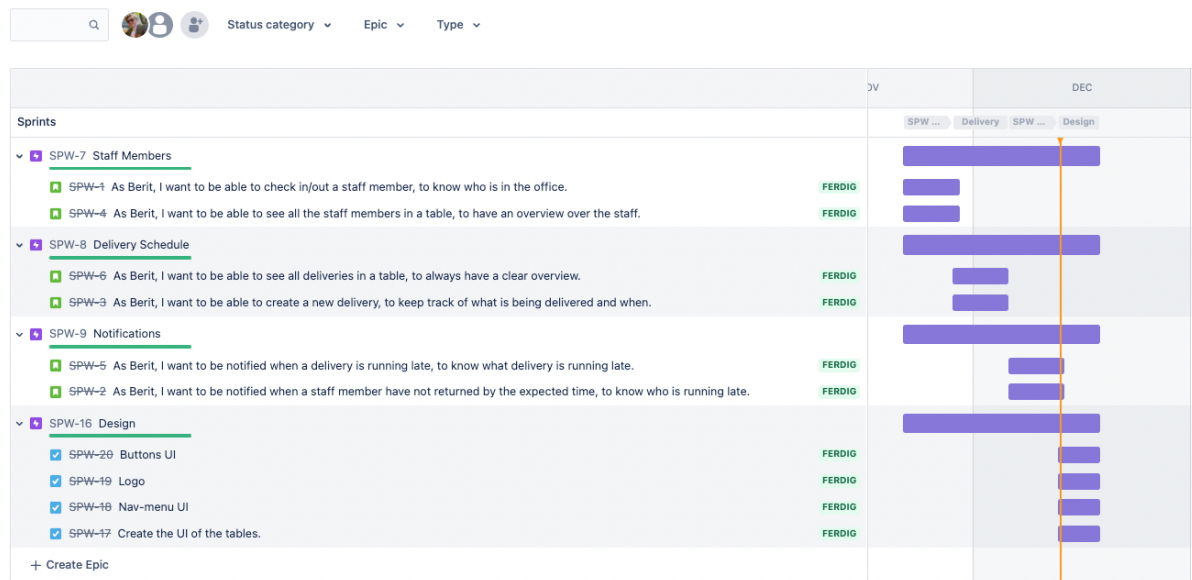
Roadmap:	2
Commits:	3
Stories:	4
Epic:	6
Management of the project:	7
Branching Strategy:	7
Challenges:	7
Solutions:	8

I have decided to not include screenshots of every single Story, Issue, Epic, but instead I have included something of everything, to show my understanding of how a project should be managed.

Roadmap & Sprints:

Projects / Semester Project - WDT

Roadmap



This is my Roadmap for the semester project. In this screenshot you can see all the stories and tasks that has been a part of each of the Epics and Sprints.

I decided to make one Sprint for each Epic, since that made the most sense in this project being only one developer on the team.
























Unfortunately, I do not have any screenshots of my Backlog when all the stories, issues and planned sprints were there. But each Epic was a sprint, with a sprint goal.

The sprint goal for the sprint "Staff Members" was "Berit should be able to see all the staff members, and clock them in and out of the office".

The sprint goal for the "Notification" sprint was "Berit should be notified when a delivery or staff member is running late".

Commits:

Commits

Search commits <input type="text"/>			All branches
Author	Commit	Message	
	9131f5f	Application Folder and Documentation Folder	
	1ed91c3	MERGED Merged dev into main	
	83f39b8	Application done	
	7ac3f90	Minor fixes to the code.	
	3947d4d	MERGED Merged SPW-16 into dev	
	cfd0837	Design finished	
	375a46a	Rounded corners on table	
	3c0c397	MERGED Everything working as intended, design remaining, but the application is useable.	
	16afb3c	Clean up in the code	
	17df384	MERGED Merged feature/SPW-5 into dev	
	15e7ee7	Notification when delivery is running late	
	9b11642	MERGED Merged feature/SPW-2 into dev	
	7ff8856	See and get notification when staff member is running late.	
	53e2655	MERGED Merged feature/SPW-6 into dev	
	44c0bd5	All deliveries in table	
	8c51410	MERGED Merged feature/SPW-3 into dev	
	0d536eb	Berit can now create a new delivery	
	8d232c3	Added validation to the inputs for scheduling	
	ca46807	MERGED Merged feature/SPW-1_1 into dev	
	31f0e6e	Checkin and CheckOut working in the staffTable	
	4c408ba	MERGED Merged feature/spw-4 into dev	
	9a40371	All staffmembers in table	
	9705e23	Project Init	

In this screenshot you can see all the commits that has been made on the project.

The commits have descriptive names of what has been done.

Stories:

SPW-8 /
SPW-6

1

...

X

As Berit, I want to be able to see all deliveries in a table, to always have a clear overview.

...

Ferdig ▾

✓ Utført

Beskrivelse

Add a description...

Child issues

Order by ▾ ... +

100% Done

SPW-23

Populate the table w...

0,5

FERDIG ▾

SPW-24

Clear checked deliver...

2

FERDIG ▾

Details ^

Tildelt

Dennis Lars William Hagström

Etiketter

None

Sprint

None +1

Story point estimate

5

Development

1 branch

1 commit 7 days ago

1 pull request MERGED

Innmelder

Dennis Lars William Hagström

Automation

Rule executions

«As Berit, I want to be able to see all deliveries in a table, to always have a clear overview».

This story was a part of the “Delivery Schedule” Epic, and “Delivery” sprint.

You can see that this story has one branch, called “feature/SPW-6”, it has one commit, and one pull request as well.

Child Issues for this story:

- Populate table with created delivery object(s) (Estimated time 0,5 hours).
- Clear checked delivery from table and remove object. (Estimated time 2hours).

SPW-7 /
SPW-4

1

As Berit, I want to be able to see all the staff members in a table, to have an overview over the staff.



Ferdig

✓ Utført

Beskrivelse

Add a description...

Child issues

Order by

100% Done

SPW-10	Set up the table.	0,5	FERDIG
SPW-11	API Request to get u...	0,5	FERDIG
SPW-12	Create staff member...	0,5	FERDIG
SPW-13	Populate the table w...	0,5	FERDIG

Details

Tildelt Dennis Lars William Hagström

Etiketter None

Sprint None +1

Story point estimate 5

Development 1 branch

Create commit

Create pull request

Innmelder Dennis Lars William Hagström

Automation Rule executions

Here is another story related to the “Staff Members” Epic. This story has 4 child issues, which all has about 0.5 hours of work. This story does not have a description since the name of the story itself is descriptive enough.

Child issues:

- Set up the table.
- API Request to get users.
- Create staff member objects from class.
- Populate the table with objects data.

Epic:

SPW-8   1   ... 

Delivery Schedule

   ...

Gjøremaal ▾





Beskrivelse

Berit should be able to create new deliveries, and also see all the created deliveries in a table.






Child issues

Order by ▾ ... +

 100% Done

-  **SPW-6** As Berit, I want to be a... 5  **FERDIG** ▾
-  **SPW-3** As Berit, I want to be a... 8  **FERDIG** ▾

Details

Tildelt	 Dennis Lars William Hagström
Etiketter	None
Start date	21. nov. 2022
Forfallsdato	18. des. 2022
Development	 Create branch  Create commit ▾
Innmelder	 Dennis Lars William Hagström
Automation	 Rule executions

This is the “Delivery Schedule” Epic. In this Epic there is a description, two stories with story points, and who the story is assigned to.

Management of the project:

Using Jira and the methodology Scrum for this project has really helped me a lot with keeping a structured workflow and overview throughout the project. I used Bitbucket for this project, which made it possible for me to create new branches of my stories and epics. This was helpful to separate things from each other, and to know what I was working on.

I then cloned the Bitbucket repository, and pushed it into a Git Repo, since there was a requirement to use GitHub in this project.

I have not done sprint retrospectives and sprint reviews since I have been the only one taken part of this project. Otherwise, I would have had both a sprint review with the scrum master, product owner and rest of the team, and a sprint retrospective with the scrum team.

Branching Strategy:

My branching strategy was based on **Main -> Dev -> Feature**.

For every new Story in each sprint, I made a new branch with the "feature/SPW-{id}" as the name directly from the Story card. These feature branches were then merged with the "Dev" branch when they were finished. When everything was done and working, I merged the "Dev" with the "Main" branch.

Challenges:

What I found to be the most challenging was the timestamps. I know that JavaScript is not the best at working with dates and timestamps unless you are using libraries which can help with that. For this project I decided to not use anything else than the built in function "Date()" to solve this.

Another thing I found challenging was the rounded corners on table with a border, that one was not an easy one in my opinion. I managed to solve it by adding a separate div under the table, which then has a border around the table, with a border-radius.

Solutions:

For the tables I chose to use checkboxes, so that you can select more than one element to manipulate. Reason for choosing checkboxes was because in the future when there are a lot of staff members, you appreciate it being a button where you can “check all”. Today it just says “check”, but in the future it would be a “check all” checkbox here as well.

Since we are working without a database in this project, I decided to make a “fake”-database named “db”. In this database I have split the staff members and delivery driver objects into separate arrays, “staffMembers” and “deliveryDrivers”.

To be able to know which object I am working with inside the tables and throughout the project, I made a function “randomId()” to generate a random id from 0-10000 for each object. These id’s are then pushed to another array called “all_id”. If they do not already include in the array, then it would have to generate a new one before pushing it to the array. This is why I have a while loop inside the “randomId()” function.

The class StaffMember which extends class Employee has three methods. The first method is the “checkOut()” method, which is used to check members out of the office, and gather the information needed to calculate however long they are staying out.

The second method is the “checkIn()” method, where the receptionist can check in staff members when they arrive back to the office.

The third method, “staffMemberIsLate()”, similar to the method “deliveryDriverIsLate()”, found in the DeliveryDriver instance as well, are functions which are triggered if a staff member or delivery driver is running late, in other words, the estimated return time has passed the current time. These methods take in the object which is late, and pass the object

to the “spawnXToast”, which then spawn the toast with information about the delivery driver or staff member.

I also have two functions checking if there is a “Return Time” or and “Estimated Return Time” set on the staff member objects or the delivery driver objects. These two functions are put into the built in “setInterval()” function which runs every second, just like the “digitalClock()” function, which also updates every second with the help of a “setInterval()” function.

In my validation of the input from the “Schedule Delivery Table”, I made a separate function which validates the data when submitted. The value of the inputs are sent to my “validateDelivery()” function, which has an array called “errorList”. If there is an error in the validation, a message will be pushed to the errorList array before continuing validating the rest. When all the validation is done, the errorList will then be returned. I will check if the length of the array returned from the “validateDelivery()” function is equal to 0. If that is true, the validation will be passed, if not I will show the errors in an alert to the users so he or she knows exactly what was wrong in the input fields.