

**Student**: Ilyas Mohamed

**Project**: Snoepfabriek van Melle

**Klas**: TTB4-SSD3C

Inhoud

[K1W1 Plannen 3](#_Toc128003780)

[Informatie verzamelen 3](#_Toc128003781)

[Casus: 3](#_Toc128003782)

[Opdracht vastellen 3](#_Toc128003783)

[User story: 3](#_Toc128003784)

[Acceptatiecriteria: 4](#_Toc128003785)

[Defintion of done: 5](#_Toc128003786)

[Conventions: 5](#_Toc128003787)

[Afspraken: 6](#_Toc128003788)

[Planning: 6](#_Toc128003789)

[K1W2 Ontwerpen 7](#_Toc128003790)

[Functioneel ontwerp 7](#_Toc128003791)

[Wireframes: 7](#_Toc128003792)

[Styleguides: 10](#_Toc128003793)

[Technisch ontwerp 11](#_Toc128003794)

[Activitydiagram: 11](#_Toc128003795)

[ERD-diagram: 11](#_Toc128003796)

[K1W3 Realiseren 12](#_Toc128003797)

[K1W4 Testen 12](#_Toc128003798)

[Technische test 12](#_Toc128003799)

[Functionele test 12](#_Toc128003800)

[K1W5 Opleveren 13](#_Toc128003801)

[Optimalisatie 13](#_Toc128003802)

[Presenteren 13](#_Toc128003803)

# K1W1 Plannen

## Informatie verzamelen

### Casus:

Snoepfabriek Van Melle in Breda zou graag een prototype willen hebben van een storingsmonitor. Een dashboard waarin ze de storingen kunnen bijhouden van de machines in de fabriek. De storingen hebben een specifieke status. De storingen moeten door medewerkers gemaakt en afgehandeld kunnen worden.

Bij een storing moet een medewerker gekoppeld kunnen worden die hier vervolgens mee aan de slag dient te gaan. Een medewerker kan een storing afhandelen in het systeem. Een medewerker moet een storing uit het systeem kunnen verwijderen.

Later zal er ook een monitoringssysteem moeten komen waarop men kan zien hoe vaak een bepaalde machine in storing is geweest, en welke medewerker de meeste storingen heeft verholpen enz.

**Het ingeven van een critical failure storing:**

Alleen een voorman medewerker kan een critical storingmelding doen. Een voorman medewerker stelt eerst vast of het een critical failure betreft. Voor veiligheidsissues is het belangrijk dat een machine handmatig uitgezet wordt als het een critical failure betreft. Pas dan kan de melding gedaan worden, anders krijg je een melding: Kan geen critical storing toevoegen; machine nog in werking. Medewerker vult de betreffende storinggegevens in. Als een bepaalde medewerker direct beschikbaar is, koppelt hij daar gelijk de betreffende medewerker aan.

Een medewerker kan maar aan één storing gekoppeld worden tegelijkertijd. Is de medewerker nog gekoppeld aan een andere storing kan deze niet gekoppeld worden. De gekoppelde medewerker krijgt een mail en een sms-notificatie als deze gekoppeld wordt aan een storing.

## Opdracht vastellen

### User story:

* Als medewerker bij Snoepfabriek Van Melle in Breda wil ik een storingsmonitor-dashboard hebben, zodat ik de storingen van de machines in de fabriek kan bijhouden en deze snel en efficiënt kan afhandelen.
* De storingen moeten een specifieke status hebben en kunnen worden toegewezen aan een medewerker die verantwoordelijk is voor het oplossen van de storing. Daarnaast moet het mogelijk zijn om een storing te verwijderen uit het systeem wanneer deze is opgelost.
* Het dashboard moet later uitgebreid worden met een monitoringssysteem waarop we de historie van storingen kunnen zien, zoals hoe vaak een bepaalde machine in storing is geweest en welke medewerker de meeste storingen heeft verholpen.
* Met deze storingsmonitor kunnen we de efficiëntie van onze fabriek verbeteren door storingen snel te signaleren, toewijzen en oplossen.
* Als voorman medewerker wil ik een critical failure storingmelding kunnen ingeven, zodat ik snel actie kan ondernemen bij ernstige storingen.

### Acceptatiecriteria:

* Het dashboard geeft een duidelijk overzicht van alle storingen in de fabriek met een specifieke status.
* Een medewerker kan een storing aanmaken en toewijzen aan een andere medewerker om deze op te lossen.
* Een medewerker kan een storing afhandelen en de status ervan bijwerken.
* Een medewerker kan een storing uit het systeem verwijderen wanneer deze is opgelost.
* Het monitoringssysteem toont de historie van storingen, zoals het aantal storingen per machine en welke medewerker de meeste storingen heeft verholpen.
* Het dashboard en het monitoringssysteem zijn gebruiksvriendelijk en gemakkelijk te begrijpen voor de medewerkers van de fabriek.
* Het systeem is veilig en betrouwbaar, met beperkte toegang tot geautoriseerde gebruikers.
* Het systeem is schaalbaar en kan worden uitgebreid als dat nodig is om aan toekomstige behoeften te voldoen.
* Het systeem wordt regelmatig getest en onderhouden om te zorgen dat het correct functioneert.
* Alleen voorman medewerkers hebben toegang tot de optie om een critical failure storingmelding te doen.
* De voorman medewerker moet kunnen vaststellen of het om een critical failure gaat voordat de melding gedaan kan worden. Als het geen critical failure betreft, kan er geen melding gedaan worden.
* Als het een critical failure betreft, moet de betreffende machine handmatig uitgezet worden voordat de melding gedaan kan worden. Als de machine nog in werking is, krijgt de medewerker een melding dat er geen critical storing toegevoegd kan worden.
* Bij het invoeren van de melding moet de medewerker de benodigde storinggegevens kunnen invullen.
* Als de betreffende medewerker direct beschikbaar is, moet deze aan de storing gekoppeld kunnen worden.
* Een medewerker kan maar aan één storing tegelijkertijd gekoppeld worden. Als de medewerker nog gekoppeld is aan een andere storing, kan deze niet aan de nieuwe storing gekoppeld worden.
* De gekoppelde medewerker moet een mail en sms-notificatie krijgen wanneer deze aan een storing wordt gekoppeld.

### Defintion of done:

* Het storingsmonitor-dashboard is ontworpen en ontwikkeld volgens de specificaties en acceptatiecriteria van de user story.
* De user story is afgerond en goedgekeurd door medestudent.
* De code is getest volgens de testcases die zijn opgesteld in het testplan.
* De code is gereviewd en goedgekeurd door een andere ontwikkelaar.
* De code is gedeployed naar de testomgeving en getest door de testers.
* De code is gedeployed naar de productieomgeving en getest door de gebruikers.
* Er is feedback verzameld van gebruikers en waar nodig zijn aanpassingen gemaakt aan het systeem om het gebruiksvriendelijker te maken.

### Conventions:

* Gebruik duidelijke en betekenisvolle namen voor je variabelen, functies, klassen en bestanden. Maak gebruik van de conventies die Laravel biedt, zoals het gebruik van de 'studly case' voor klassenamen en 'snake case' voor variabelen en functienamen.
* Maak gebruik van de ingebouwde Laravel functionaliteiten om het proces voor het melden en afhandelen van storingen te stroomlijnen. Gebruik bijvoorbeeld de Laravel Resource Controllers en Eloquent ORM om eenvoudig CRUD (Create, Read, Update, Delete) functionaliteit in te bouwen in je applicatie.
* Maak gebruik van de ingebouwde Laravel beveiligingsmechanismen om het systeem veilig te houden. Gebruik bijvoorbeeld Laravel's ingebouwde authenticatie en autorisatie mechanismen om alleen geautoriseerde gebruikers toegang te geven tot het dashboard.
* Gebruik Git voor versiebeheer om je code te beheren en de geschiedenis van wijzigingen bij te houden. Maak regelmatig commits en voeg duidelijke commit messages toe om de wijzigingen in je code te documenteren.
* Schrijf tests om de functionaliteit van je applicatie te verifiëren en ervoor te zorgen dat het correct werkt. Gebruik bijvoorbeeld Laravel's ingebouwde test-framework en PHPUnit om tests te schrijven voor de verschillende functionaliteiten.
* Documenteer je code duidelijk en consistent. Voeg commentaar toe aan je code om uit te leggen wat het doet en waarom bepaalde beslissingen zijn genomen.
* Gebruik Laravel's Blade templating engine om de weergave van je applicatie te scheiden van de logica. Maak gebruik van Blade layouts en partials om herbruikbare codeblokken te maken en houd de weergavelogica overzichtelijk.
* Maak gebruik van de Laravel debug tools, zoals de debugbar en Telescope, om de prestaties van je applicatie te monitoren en eventuele problemen op te sporen.
* Maak gebruik van best practices voor performance en schaalbaarheid, zoals het gebruik van caching en lazy-loading van data, om ervoor te zorgen dat je applicatie snel en efficiënt werkt en gemakkelijk kan worden opgeschaald als dat nodig is.

### Afspraken:

* Tijdig afmelden bij afwezigheid: als een teamlid niet aanwezig kan zijn bij een vergadering of werksessie, moet deze persoon dit zo snel mogelijk laten weten aan de rest van het team. Op deze manier kan het team rekening houden met afwezigheid en eventueel de planning aanpassen.
* Vaststellen van deadlines: het team moet duidelijke deadlines vaststellen voor elk project en elke taak. Dit helpt om ervoor te zorgen dat het werk op tijd wordt afgerond en voorkomt dat teamleden te laat komen.
* Taakverdeling: het is belangrijk om een duidelijke taakverdeling te hebben, zodat iedereen weet wat er van hem of haar wordt verwacht. Dit helpt om ervoor te zorgen dat het werk op tijd wordt afgerond en voorkomt dat teamleden elkaar in de weg zitten.
* Backup-plan: het team moet een backup-plan hebben voor het geval dat een teamlid ziek is of om andere redenen niet in staat is om te werken. Dit kan bijvoorbeeld betekenen dat een ander teamlid taken overneemt of dat de deadline wordt aangepast.
* Communicatie: communicatie is de sleutel tot succesvol samenwerken. Het team moet regelmatig communiceren over de voortgang van het project, problemen die zich voordoen en eventuele wijzigingen in de planning. Dit helpt om ervoor te zorgen dat iedereen op de hoogte is en kan bijdragen aan het succes van het project.

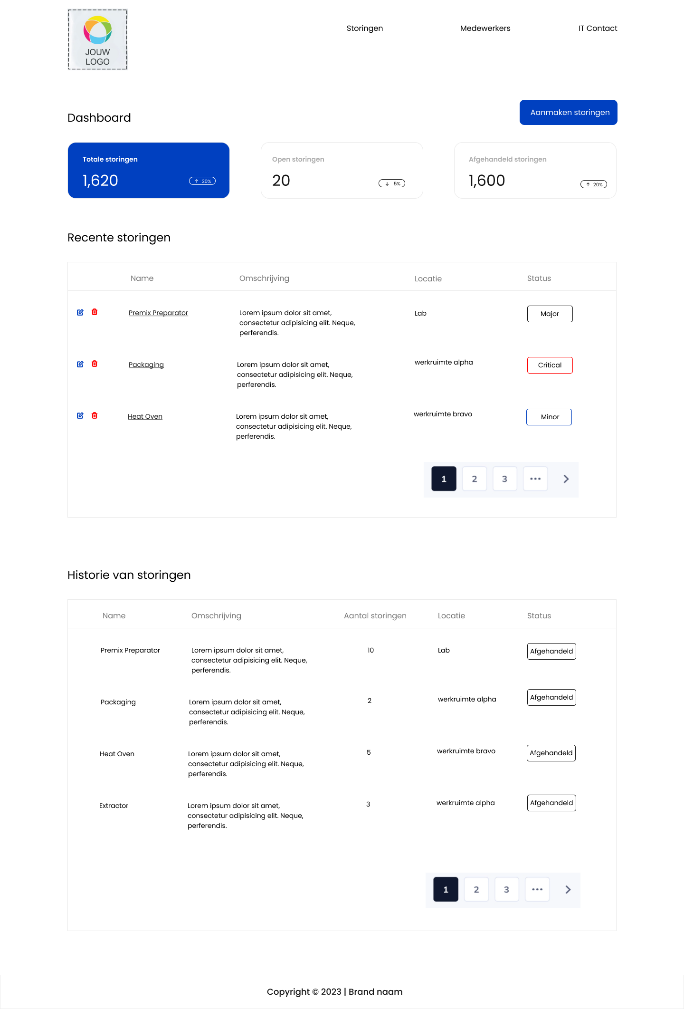
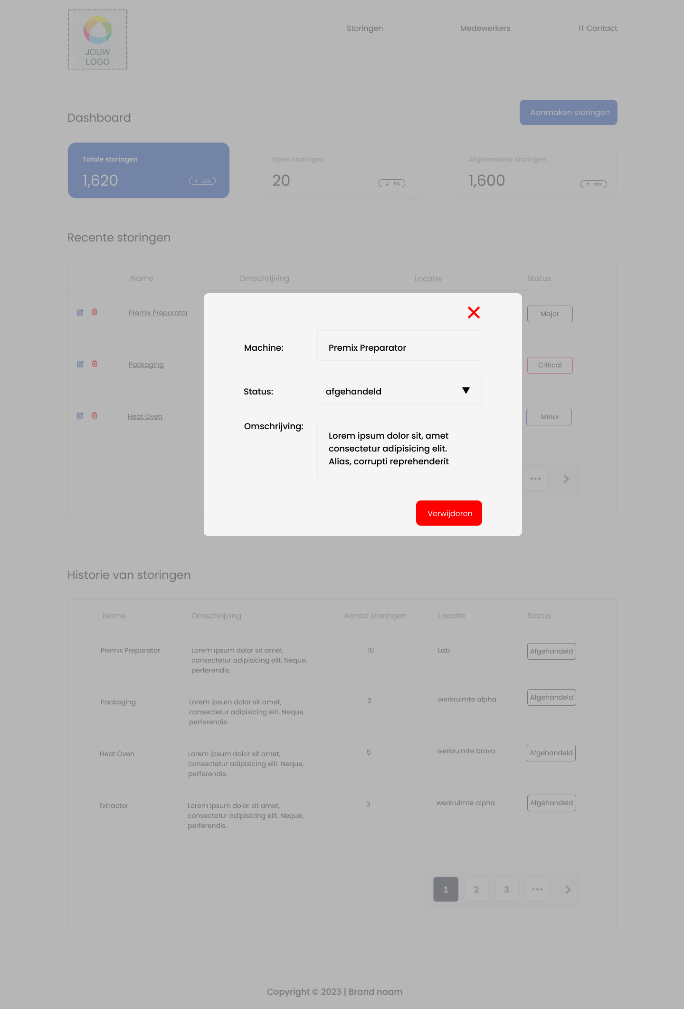
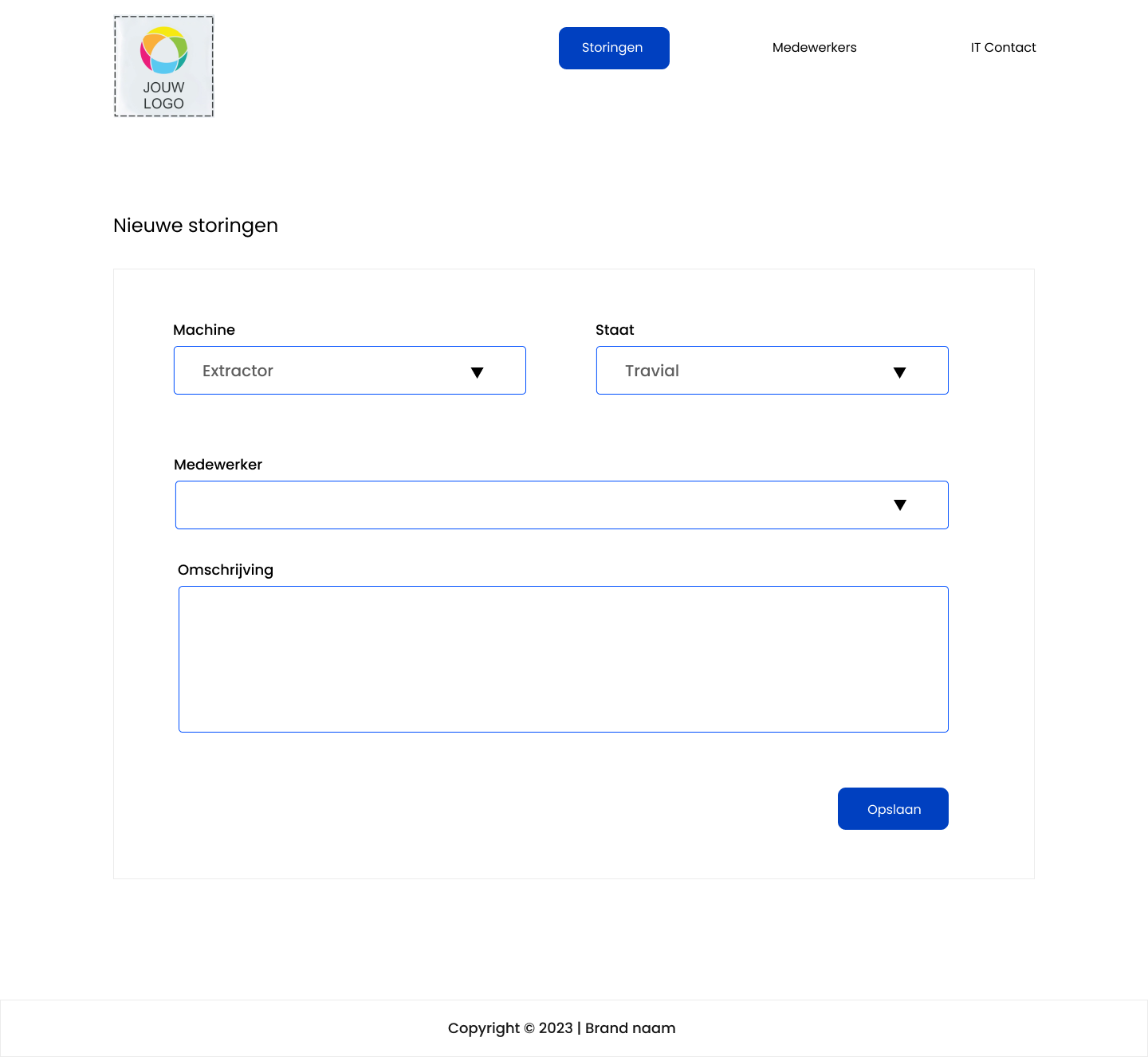
### Planning:

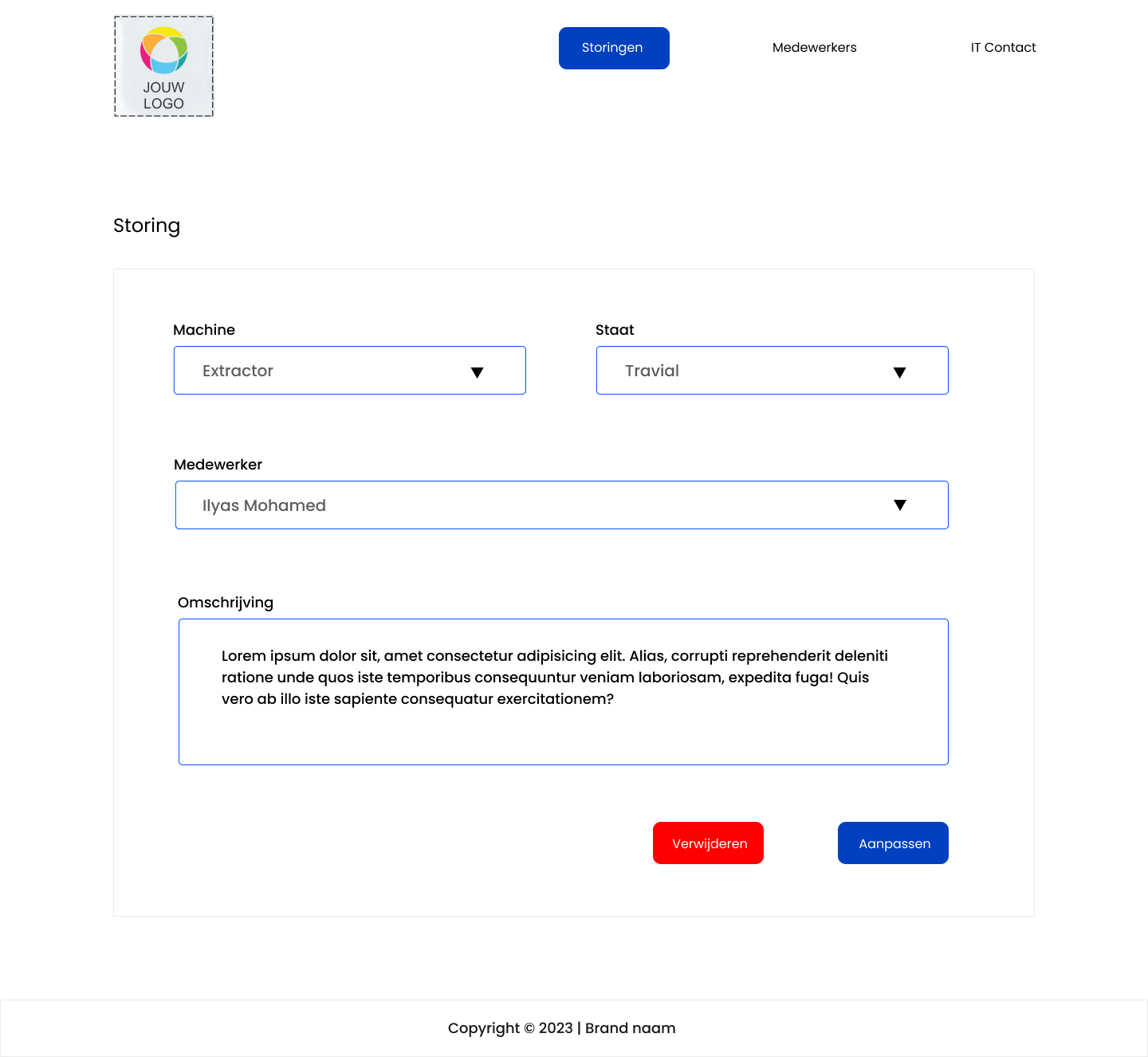
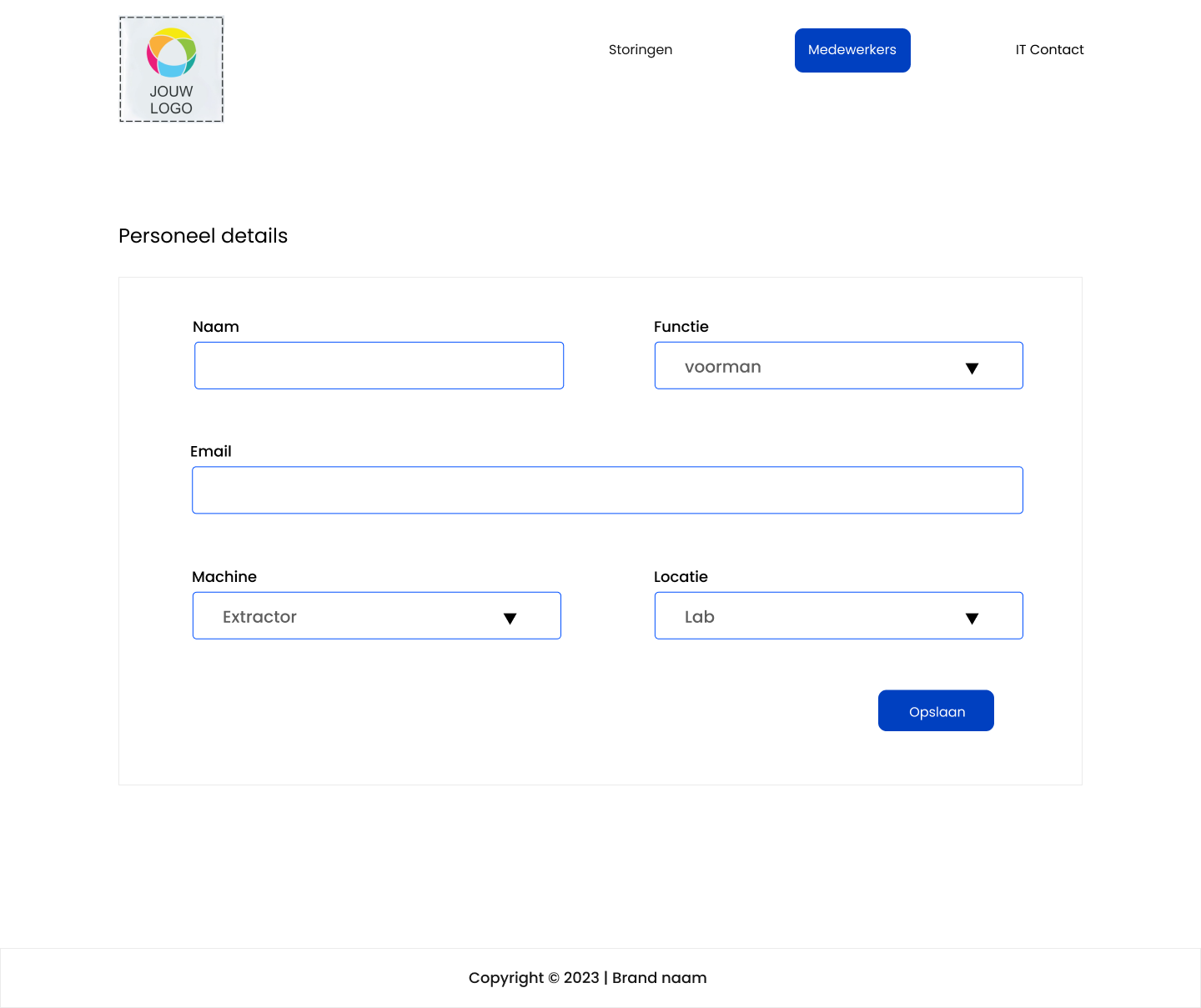
|  |  |
| --- | --- |
| Dag 1 | Identificeren van de entiteiten in de casus |
| Ontwerpen van de ERD |
| Maken van een database omgeving |
| Aanmaken van de tabellen in de database |
| Definiëren van de relaties tussen de tabellen |
| Toevoegen van de initiële dataset aan de database |
| Ontwikkelen van de verbindings tabel tussen medewerkers en storingen |
| Dag 2 | Ontwikkelen van het storingsmonitor-dashboard |
| Toevoegen van de functies om storingen toe te wijzen en te verwijderen |
| Testen van de functionaliteit van het storingsmonitor-dashboard |
| Ontwikkelen van het monitoringssysteem |
| Toevoegen van de mogelijkheid om de historie van storingen te bekijken |
| Testen van de functionaliteit van het monitoringssysteem |
| Toevoegen van de functies om medewerkers toe te wijzen aan storingen |
| Toevoegen van de functie om critical failure storingen te melden |
| Testen van de functionaliteit van de critical failure storingmelding |
| Dag 3 | Technische testen en acceptatie test |
| Dag 4 | Presenteren en opleveren |
|  |  |

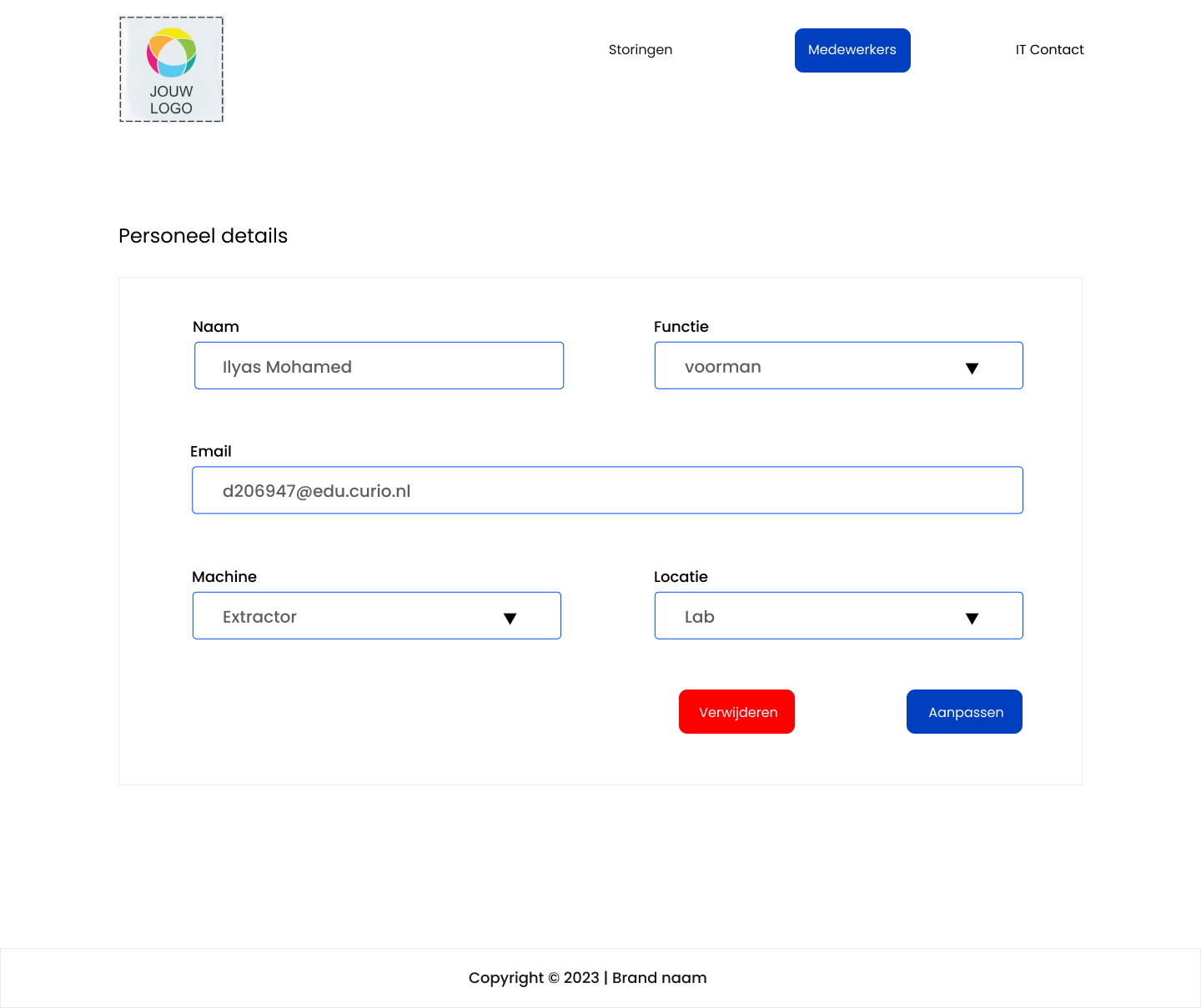
# K1W2 Ontwerpen

## Functioneel ontwerp

### Wireframes:







### Styleguides:

Aa Bb

#161616

#0040C0

#FFFFFF

**Verwijderen**

**Opslaan**

**Minor**

**Critical**

**Major**

**Content:** **’Poppins’, sans-serif**

**Font weight:** Light

Regular

**Trivial**

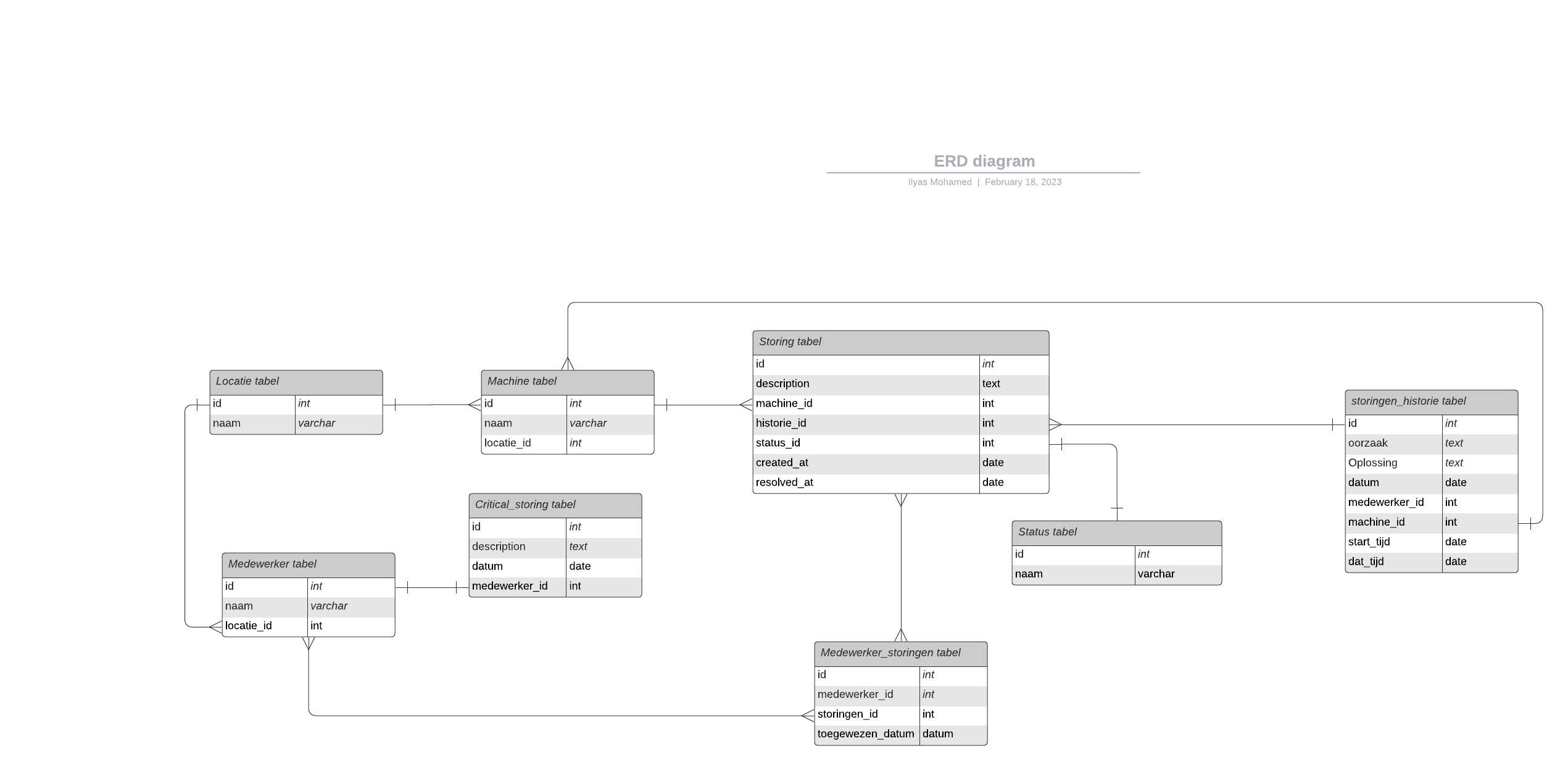
**Bold**

**Black**

## Technisch ontwerp

### Activitydiagram:

### ERD-diagram:



# K1W3 Realiseren

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et rem ea hic perspiciatis dolorum officia, inventore, quod iure doloremque neque aut sit dolores. Architecto possimus delectus mollitia. Dolorum, accusantium ipsa. Maxime repudiandae exercitationem tempore sequi fugiat molestiae excepturi ut illum consectetur aperiam alias expedita, fugit repellendus porro eius quam cum aspernatur autem error rerum quia voluptatem dolorem. Laudantium omnis doloribus maiores tenetur eaque quisquam eos incidunt deserunt praesentium iusto quia, doloremque laborum voluptas, ex dolores reprehenderit necessitatibus at minima nostrum. Odit odio quisquam eius quam et deserunt a, fugiat quo dolore saepe ad dolores illo? Distinctio, a. Blanditiis, labore minus.

# K1W4 Testen

## Technische test

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et rem ea hic perspiciatis dolorum officia, inventore, quod iure doloremque neque aut sit dolores. Architecto possimus delectus mollitia. Dolorum, accusantium ipsa. Maxime repudiandae exercitationem tempore sequi fugiat molestiae excepturi ut illum consectetur aperiam alias expedita, fugit repellendus porro eius quam cum aspernatur autem error rerum quia voluptatem dolorem. Laudantium omnis doloribus maiores tenetur eaque quisquam eos incidunt deserunt praesentium iusto quia, doloremque laborum voluptas, ex dolores reprehenderit necessitatibus at minima nostrum. Odit odio quisquam eius quam et deserunt a, fugiat quo dolore saepe ad dolores illo? Distinctio, a. Blanditiis, labore minus.

## Functionele test

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et rem ea hic perspiciatis dolorum officia, inventore, quod iure doloremque neque aut sit dolores. Architecto possimus delectus mollitia. Dolorum, accusantium ipsa. Maxime repudiandae exercitationem tempore sequi fugiat molestiae excepturi ut illum consectetur aperiam alias expedita, fugit repellendus porro eius quam cum aspernatur autem error rerum quia voluptatem dolorem. Laudantium omnis doloribus maiores tenetur eaque quisquam eos incidunt deserunt praesentium iusto quia, doloremque laborum voluptas, ex dolores reprehenderit necessitatibus at minima nostrum. Odit odio quisquam eius quam et deserunt a, fugiat quo dolore saepe ad dolores illo? Distinctio, a. Blanditiis, labore minus.

# K1W5 Opleveren

## Optimalisatie

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et rem ea hic perspiciatis dolorum officia, inventore, quod iure doloremque neque aut sit dolores. Architecto possimus delectus mollitia. Dolorum, accusantium ipsa. Maxime repudiandae exercitationem tempore sequi fugiat molestiae excepturi ut illum consectetur aperiam alias expedita, fugit repellendus porro eius quam cum aspernatur autem error rerum quia voluptatem dolorem. Laudantium omnis doloribus maiores tenetur eaque quisquam eos incidunt deserunt praesentium iusto quia, doloremque laborum voluptas, ex dolores reprehenderit necessitatibus at minima nostrum. Odit odio quisquam eius quam et deserunt a, fugiat quo dolore saepe ad dolores illo? Distinctio, a. Blanditiis, labore minus.

## Presenteren

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et rem ea hic perspiciatis dolorum officia, inventore, quod iure doloremque neque aut sit dolores. Architecto possimus delectus mollitia. Dolorum, accusantium ipsa. Maxime repudiandae exercitationem tempore sequi fugiat molestiae excepturi ut illum consectetur aperiam alias expedita, fugit repellendus porro eius quam cum aspernatur autem error rerum quia voluptatem dolorem. Laudantium omnis doloribus maiores tenetur eaque quisquam eos incidunt deserunt praesentium iusto quia, doloremque laborum voluptas, ex dolores reprehenderit necessitatibus at minima nostrum. Odit odio quisquam eius quam et deserunt a, fugiat quo dolore saepe ad dolores illo? Distinctio, a. Blanditiis, labore minus.