

**Student**: Ilyas Mohamed

**Project**: Snoepfabriek van Melle

**Klas**: TTB4-SSD3C

Inhoud

[Realiseert Software (B1-K1) 3](#_Toc129299255)

[B1-K1-W1 Plant werkzaamheden en bewaakt de voortgang 3](#_Toc129299256)

[Casus: 3](#_Toc129299257)

[Opdracht vastellen 3](#_Toc129299258)

[B1-K1-W2 Ontwerpen software 5](#_Toc129299259)

[Functioneel ontwerp 5](#_Toc129299260)

[Technisch ontwerp 9](#_Toc129299261)

[B1-K1-W3 Realiseert software 10](#_Toc129299262)

[B1-K1-W4 Test software 10](#_Toc129299263)

[Accepatie test 10](#_Toc129299264)

[B1-K1-W5 Verbetervoorstellen voor de software 10](#_Toc129299265)

[Verbetervoorstel verslag 10](#_Toc129299266)

[Optimalisatie 10](#_Toc129299267)

[Werkt in een ontwikkelteam (B1-K2) 11](#_Toc129299268)

[B1-K2-W1 Voert overleg 11](#_Toc129299269)

[B1-K2-W2 Presenteert het opgeleverd werk 11](#_Toc129299270)

[B1-K2-W3 Reflecteert op het werk 11](#_Toc129299271)

# Realiseert Software (B1-K1)

## B1-K1-W1 Plant werkzaamheden en bewaakt de voortgang

### 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 1:

Als medewerker 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.

#### Acceptatiecriteria:

* Het dashboard moet een overzicht tonen van alle storingen in de fabriek.
* Het dashboard moet de mogelijkheid bieden om storingen te sorteren op status en machine.
* Het dashboard moet de mogelijkheid bieden om snel een nieuwe storing toe te voegen.
* Het dashboard moet de mogelijkheid bieden om de details van een storing te bekijken en te bewerken.

#### User Story 2:

Als medewerker wil ik de mogelijkheid hebben om een specifieke status toe te kennen aan storingen in de fabriek, zodat ik snel kan zien welke storingen prioriteit hebben. Daarnaast wil ik de mogelijkheid hebben om een storing toe te wijzen aan een specifieke medewerker die verantwoordelijk is voor het oplossen ervan.

#### Acceptatiecriteria:

* Het dashboard moet de mogelijkheid bieden om een specifieke status toe te kennen aan storingen.
* De verschillende statussen moeten duidelijk onderscheidbaar zijn, zodat ik snel kan zien welke storingen prioriteit hebben.
* Ik moet de mogelijkheid hebben om een storing toe te wijzen aan een specifieke medewerker die verantwoordelijk is voor het oplossen ervan.

#### User Story 3:

Als medewerker wil ik de mogelijkheid hebben om de historie van storingen te bekijken via het dashboard, zodat ik trends en patronen kan herkennen en efficiëntieverbeteringen kan identificeren. Dit omvat informatie zoals hoe vaak een bepaalde machine in storing is geweest en welke medewerker de meeste storingen heeft verholpen.

#### Acceptatiecriteria:

* Het dashboard moet de mogelijkheid bieden om de historie van storingen te bekijken.
* Het dashboard moet de mogelijkheid bieden om storingen te filteren op machine en medewerker.
* Het dashboard moet gemakkelijk leesbaar en begrijpelijk zijn, zodat ik trends en patronen kan herkennen en efficiëntieverbeteringen kan identificeren.

#### User Story 4:

Als medewerker wil ik de mogelijkheid hebben om storingen snel toewijzen en oplossen met behulp van de storingsmonitor, zodat we de efficiëntie van onze fabriek kunnen verbeteren.

#### Acceptatiecriteria:

* Het systeem moet een melding genereren wanneer er een nieuwe storing wordt toegevoegd.
* Het systeem moet de mogelijkheid bieden om storingen snel toe te wijzen aan de juiste medewerker.
* Het systeem moet de mogelijkheid bieden om de status van storingen bij te werken.

#### User Story 5:

Als voorman medewerker wil ik een critical failure storingmelding kunnen ingeven, zodat ik snel actie kan ondernemen bij ernstige storingen.

#### Acceptatiecriteria:

* Het systeem moet de mogelijkheid bieden om een storing te markeren als een critical failure.
* Het systeem moet de mogelijkheid bieden om een melding te genereren wanneer er een critical failure storing wordt toegevoegd.
* Het systeem moet de mogelijkheid bieden om een critical failure storing snel toe te wijzen aan de juiste medewerker.

#### 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.

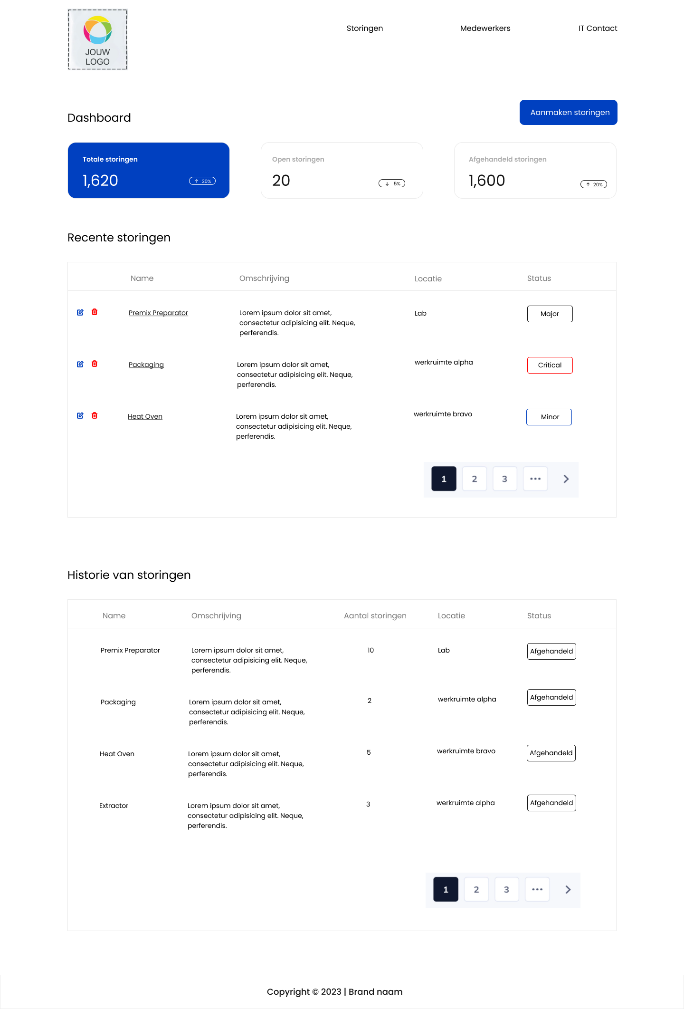
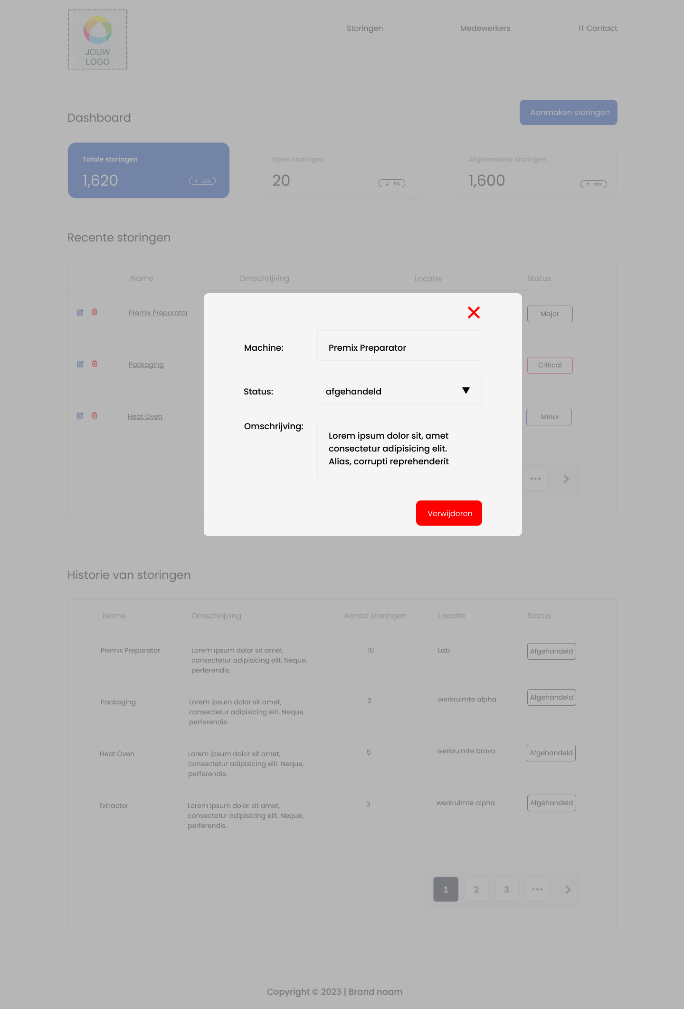
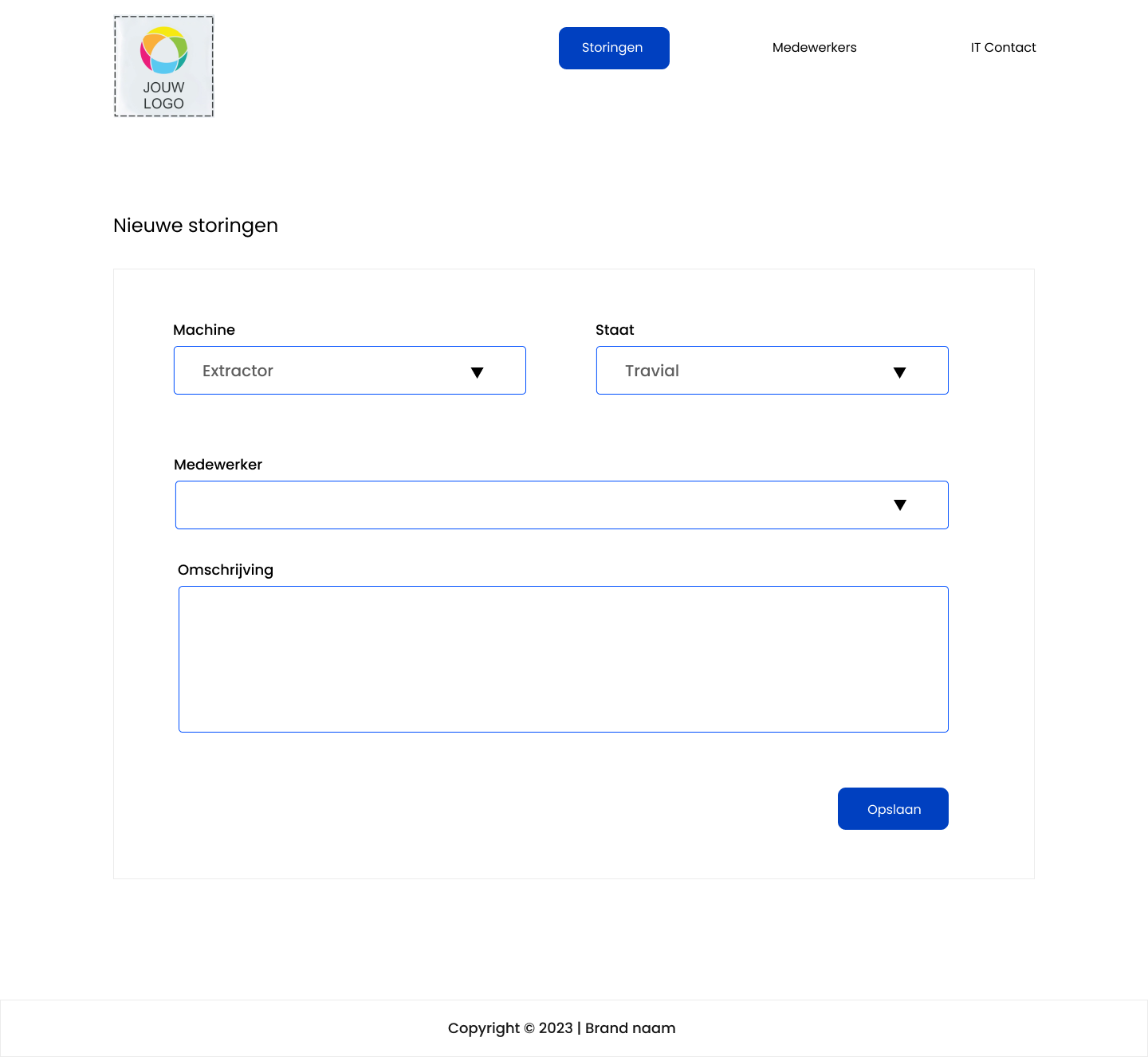
#### Planning:

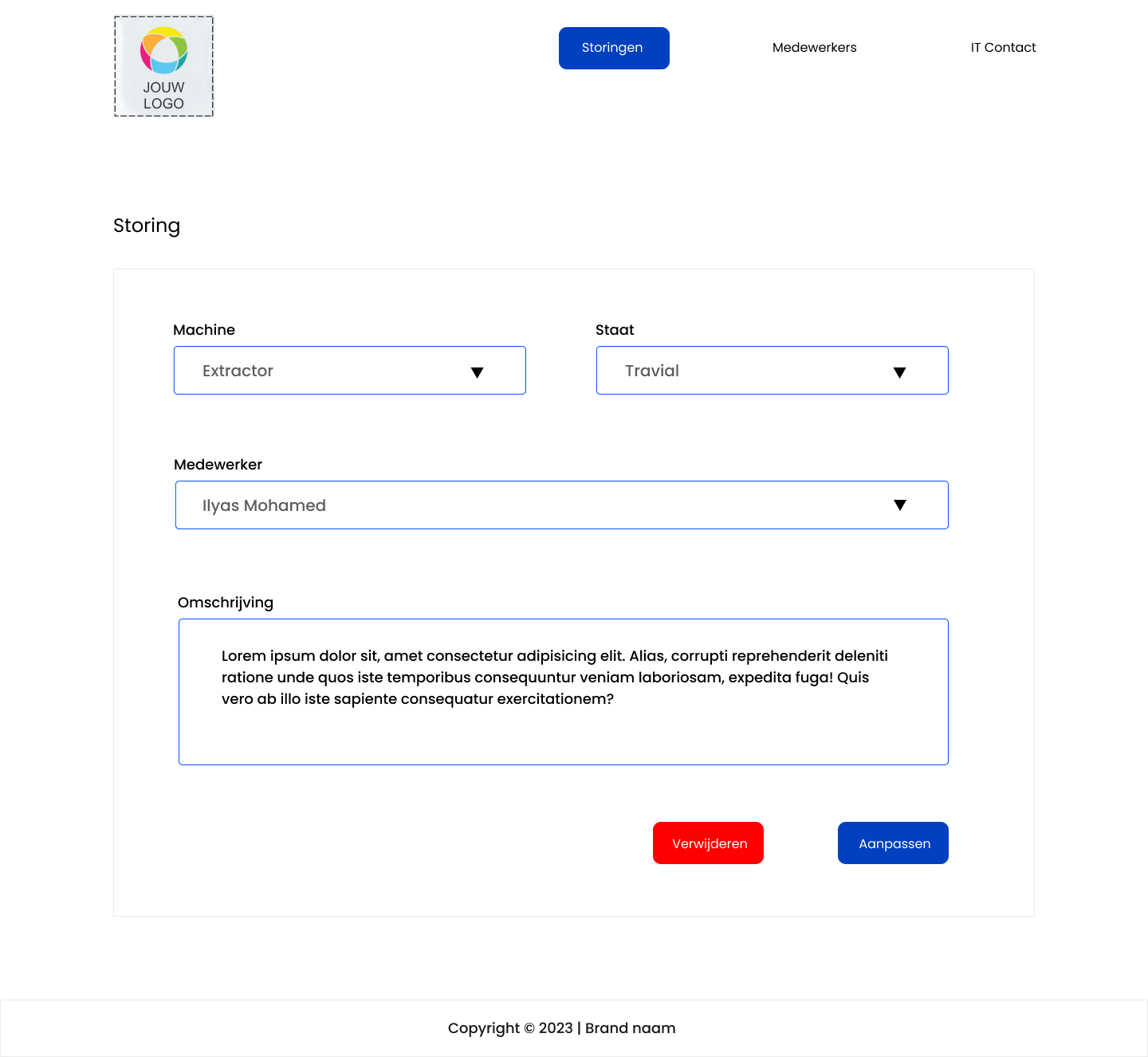
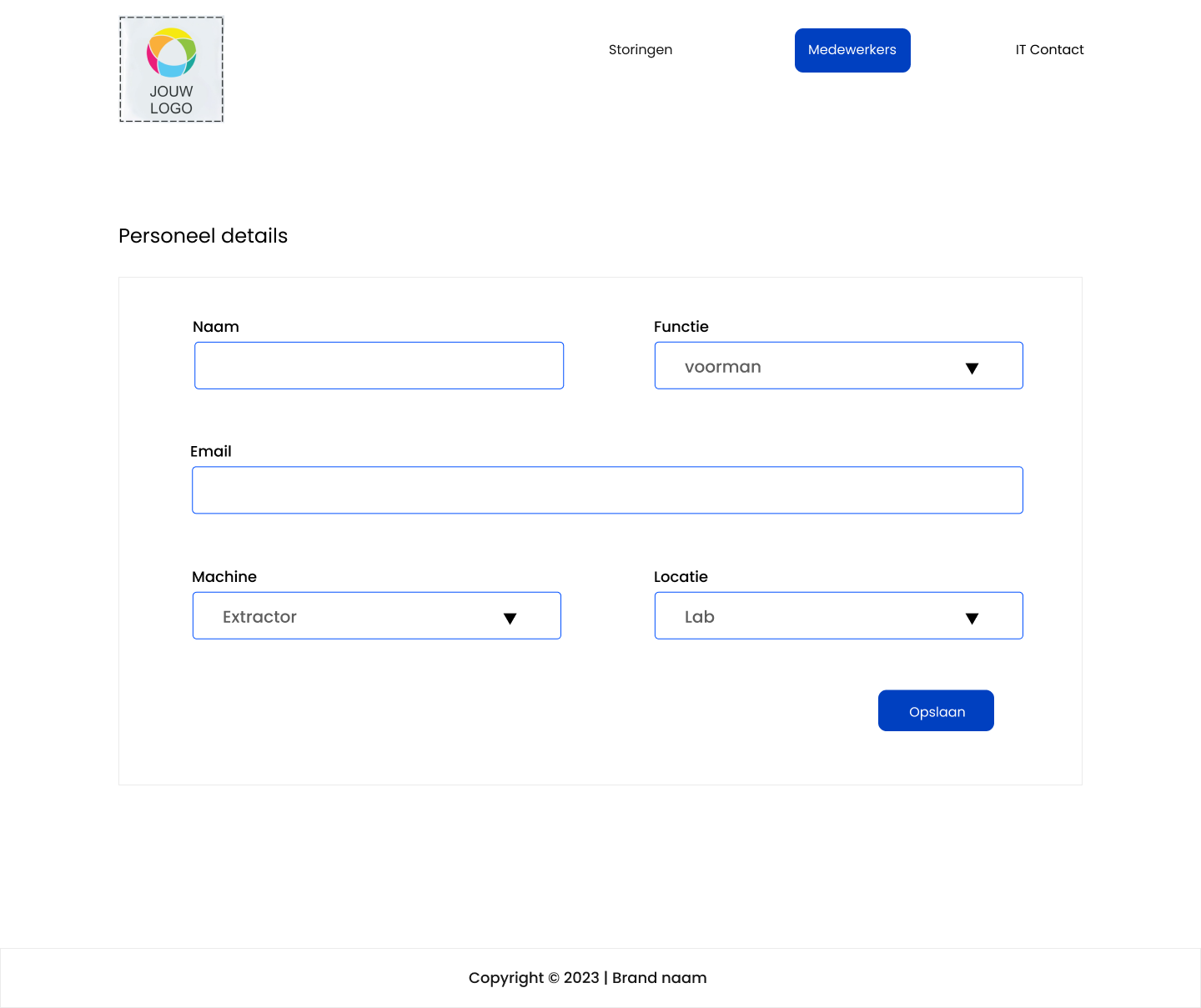
|  |  |
| --- | --- |
| Dag 1 | Identificeren van de entiteiten in de casus |
| Ontwerpen van de ERD |
| Maken van een database omgeving |
| Aanmaken van de tabellen in de database |
| Definiëren van de relaties tussen de tabellen |
| Toevoegen van de initiële dataset aan de database |
| Ontwikkelen 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 | [Inspiraties](https://dribbble.com/shots/20082513-Digital-sales-dashboard-Untitled-UI) Technische testen en acceptatie test |
| Dag 4 | Presenteren en opleveren |
|  |  |

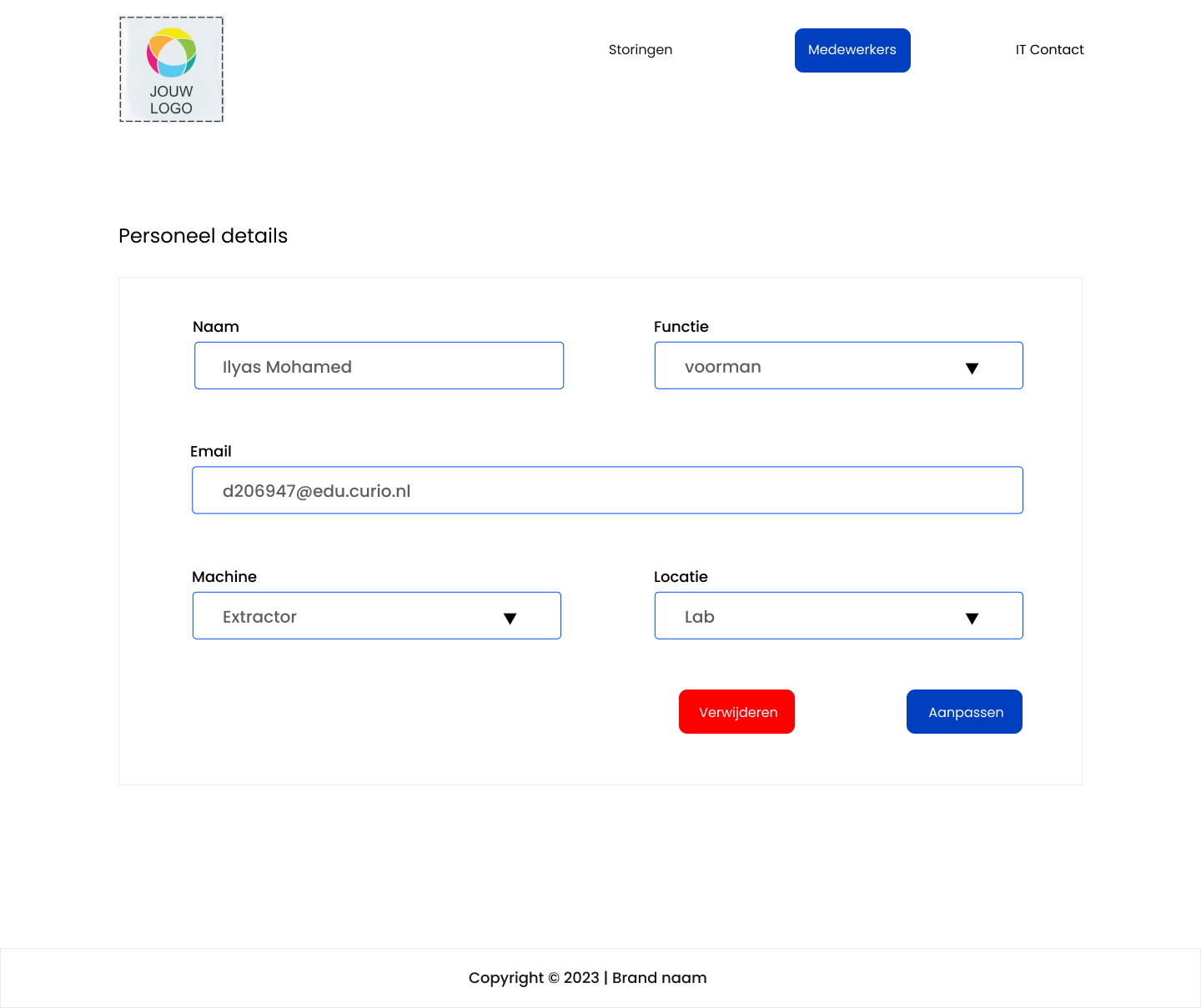
## B1-K1-W2 Ontwerpen software

### Functioneel ontwerp

#### Wireframes:







#### Styleguides:

Aa Bb

#161616

#0040C0

#FFFFFF

**Verwijderen**

**Opslaan**

**Minor**

**Critical**

**Major**

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

**Font weight:** Light

Regular

**Trivial**

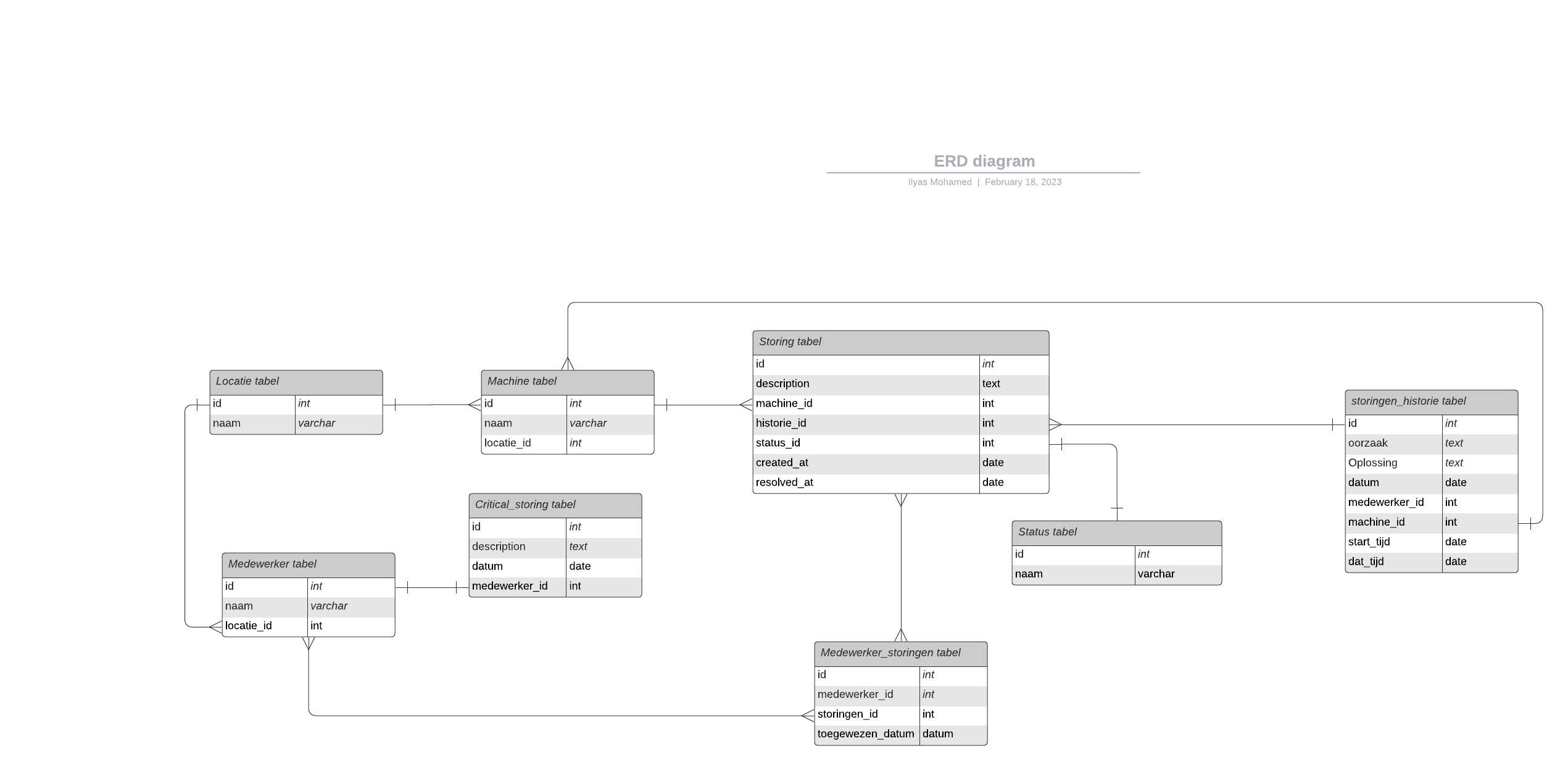
**Bold**

**Black**

### Technisch ontwerp

#### Activitydiagram:

#### ERD-diagram:



## B1-K1-W3 Realiseert software

Coderen!

## B1-K1-W4 Test software

|  |  |  |
| --- | --- | --- |
| **Als medewerker 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.** | | |
|  | **JA** | **NEE** |
| **Homepagina van dashboard** |  |  |
| 1. Open het dashboard en controleer of het een overzicht van alle storingen in de fabriek toont. |  |  |
| 1. Klik op de sorteeroptie voor status en controleer of de storingen worden gesorteerd op status. |  |  |
| 1. Klik op de sorteeroptie voor machine en controleer of de storingen worden gesorteerd op machine. |  |  |
| 1. Voeg een nieuwe storing toe via de daarvoor bestemde knop en controleer of deze direct zichtbaar is in het dashboard. |  |  |
| 1. Klik op een storing om de details te bekijken en controleer of alle relevante informatie wordt getoond. |  |  |
| 1. Bewerk de details van een storing en controleer of de wijzigingen correct worden opgeslagen en zichtbaar zijn in het dashboard. |  |  |
| **Opmerkingen:** |  |  |

## B1-K1-W5 Verbetervoorstellen voor de software

### Verbetervoorstel verslag

### Optimalisatie

Lighthouse

# Werkt in een ontwikkelteam (B1-K2)

## B1-K2-W1 Voert overleg

Laten zien

## B1-K2-W2 Presenteert het opgeleverd werk

* Bereid een demo voor presentatie
* Zorg voor een heldere presentatie
* Bespreek uw beslissingen: Bespreek de redenen achter de beslissingen.
* Demonstreer de code
* Beantwoord vragen

## B1-K2-W3 Reflecteert op het werk

Retropectieve verslag