Technisch ontwerp

Applicatie- en mediaontwikkelaar

Naam: Kaan en Armand

Datum: 23 - 03 - 2017

Versie: 1.0

Technisch ontwerp

Inhoudsopgave

[1 Inleiding 3](#_Toc465334464)

[2 Technische specificaties 3](#_Toc465334465)

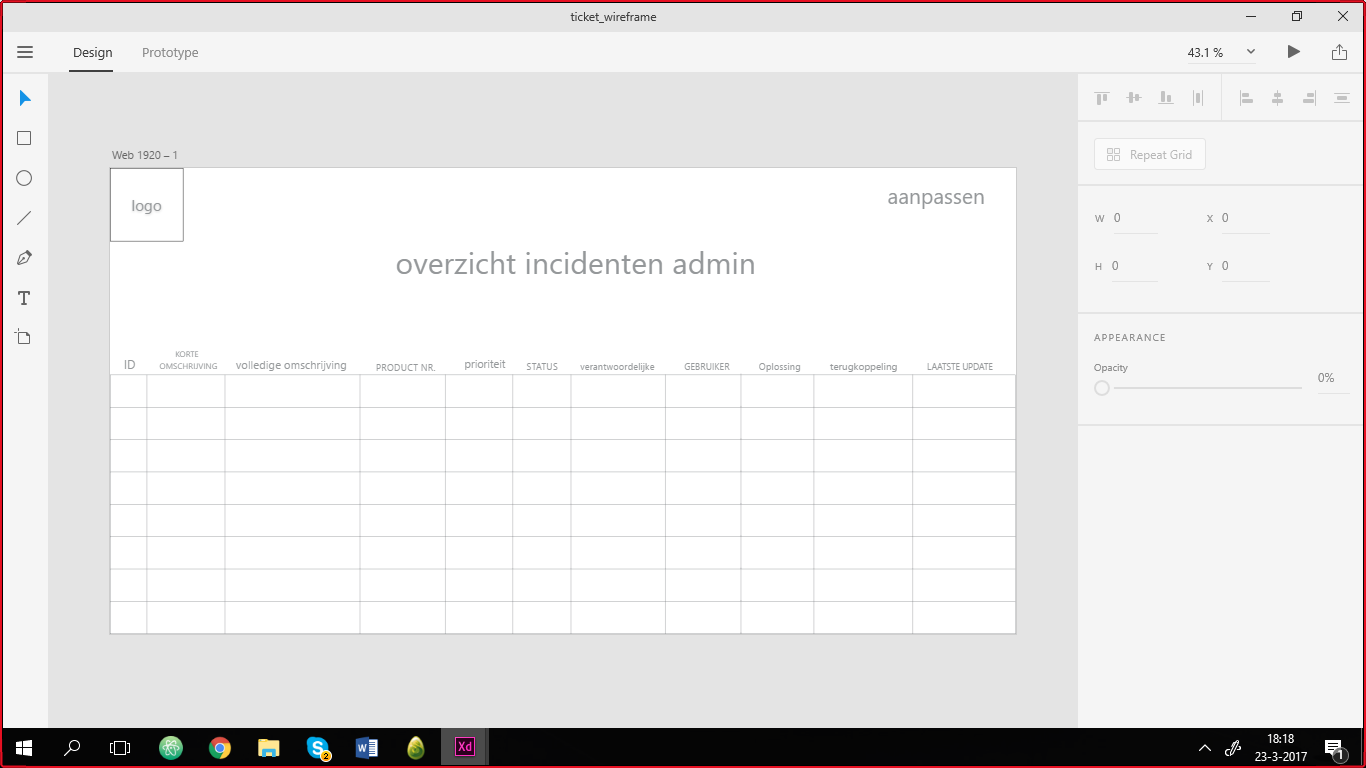
[3 Relationeel datamodel 3](#_Toc465334466)

Sjabloon Technisch ontwerp

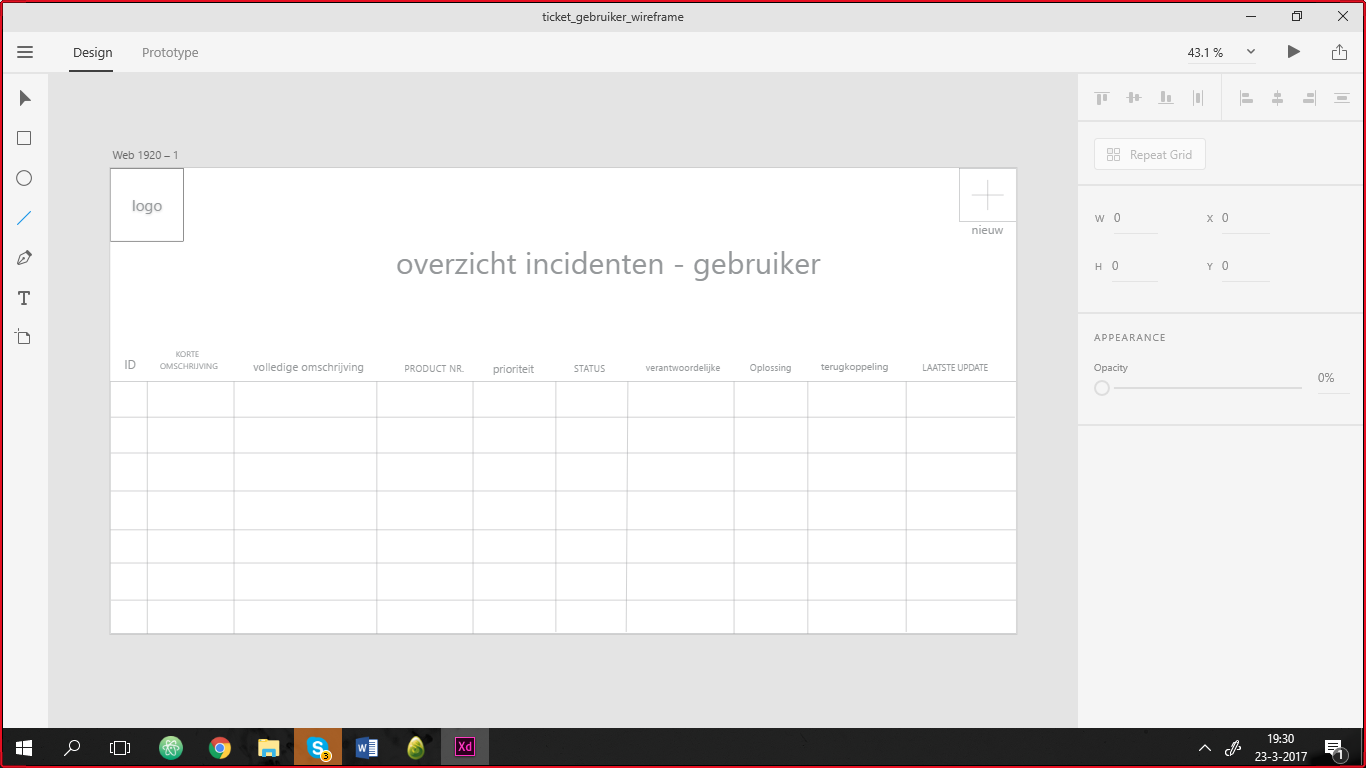
# Inleiding

Dit is het technische ontwerp over het ‘incidenten’ gedeelte

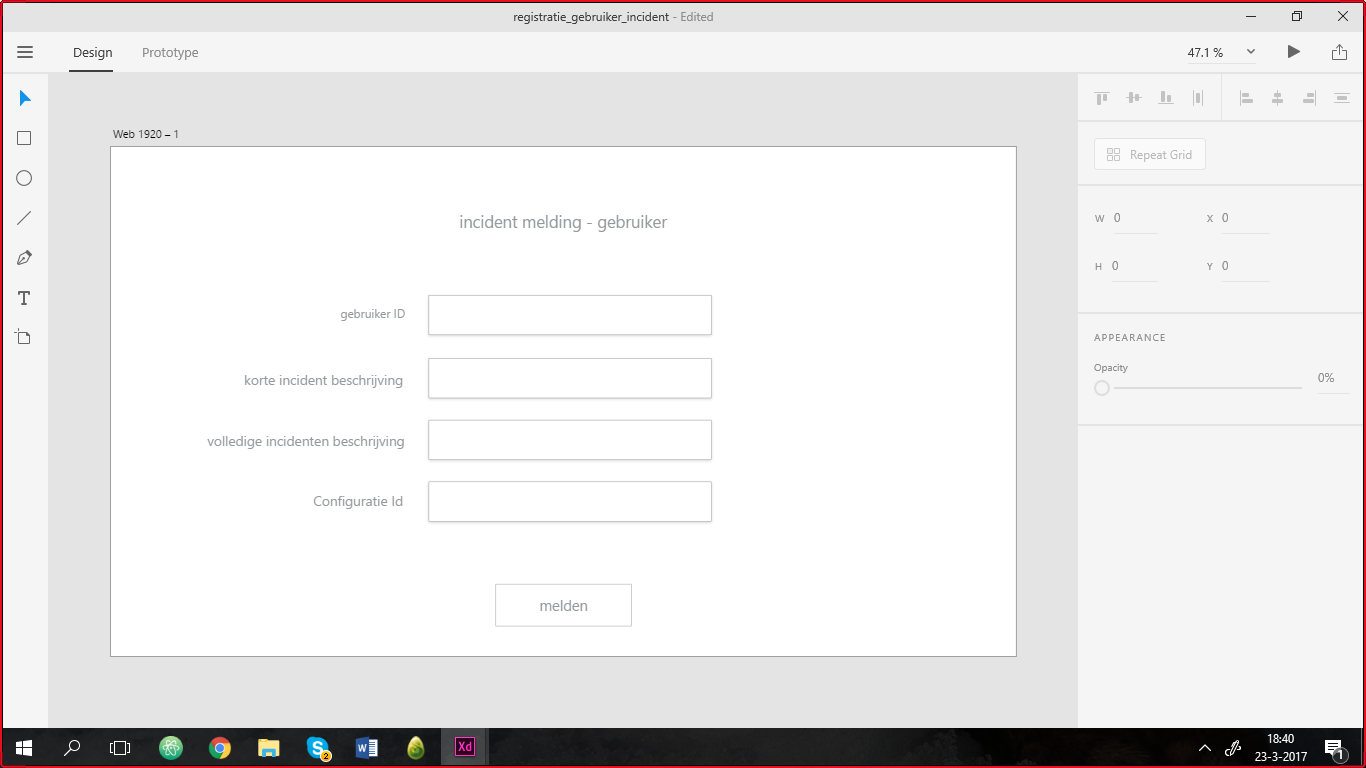
# Technische specificaties



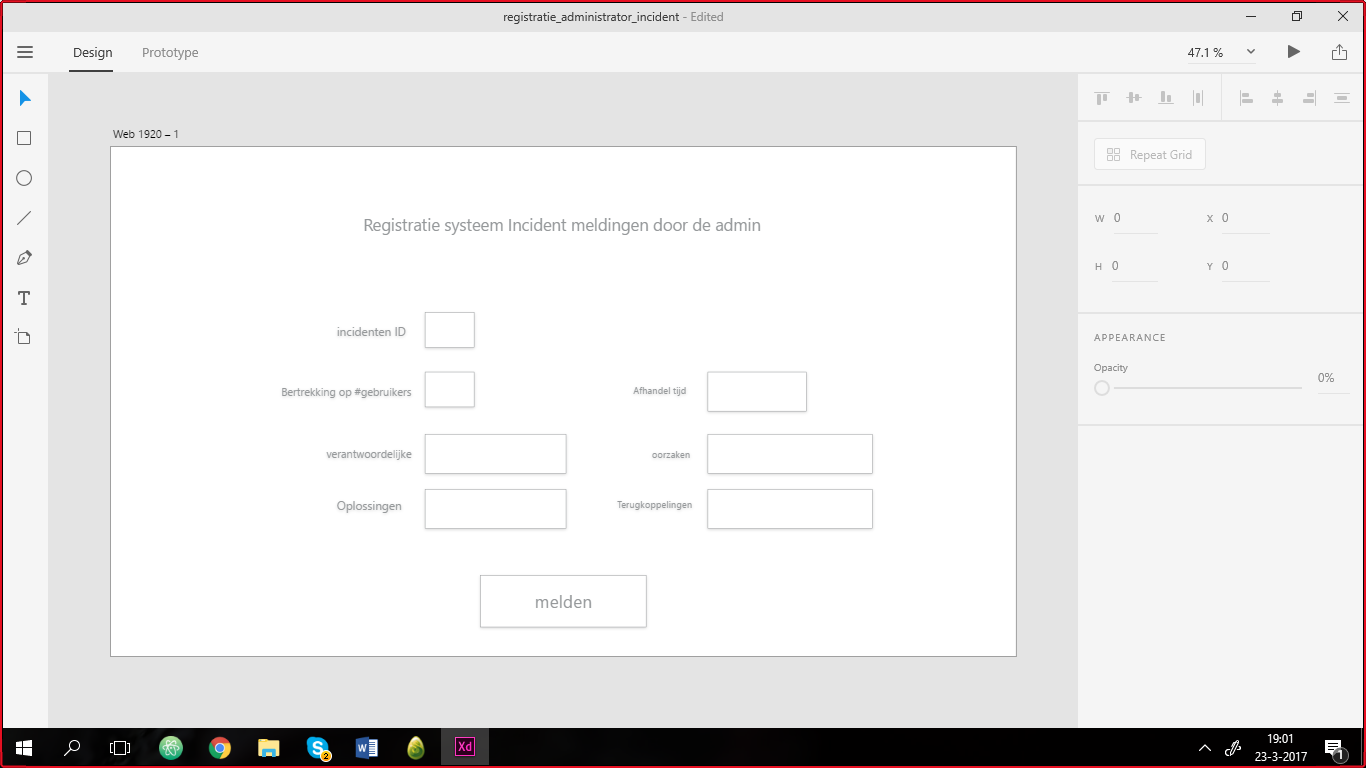
Het overzicht van de incidenten komt in een tabel waarvan de info word gehaald uit de ‘incidenten’ tabel (in de database). De Admin moet de mogelijkheid hebben om het ook nog te kunnen aanpassen dat doe je door de record te selecteren en dan de knop aanpassen aan te klikken. Als je op aanpassen drukt word je doorgeleid naar de incident melden pagina waar je de bestaande incident kan aanpassen.



Dit is het overzicht voor de gebruiker. De gebruiker kan hier alleen zijn eigen gemelde incidenten zien. En een nieuwe maken via de knop recht boven, die leid naar de incident melden pagina.



Dit is een melding formulier voor de gebruiker. Dit formulier komt uiteindelijk in de tabel ‘incidenten’ van de database. Die invoer types zijn gewoon text.



Hier kan de Admin zelf een incident melden. Deze informatie komt uiteindelijk terecht in de database en dus ook in het overzicht ticket meldingen.

**Proces validatie en sanitation**

Alle data die de administratoren en gebruikers versturen gaat eerst door de onderstaande validatie en sanitation functies. Deze functies moeten eerst oke geven voordat de gegevens door php naar de database verstuurd worden.

**Validatie**

1. We valideren de informatie op lengte(bvb niet 20 cijfers in een postcode van 6)
2. We valideren de informatie op type (bvb geen letters in een id naam)
3. We validen de informatie zo dat alleen het tussenvoegsel bij naam niet ingevuld hoeft te zijn.

**Sanitation**

1. We zetten speciale script karakters om in neutrale woorden zodat script injecties sterk bemoeilijkt worden.
2. Verder halen we met functies uit de meeste vakken alle spaties weg.
3. we testen voor mysql commando’s met if else statements.

# Relationeel datamodel

1. Normaliseer de overzichten en eventuele formulieren.

Gebruik hiervoor de 0e tot en met 3e normaalvorm.

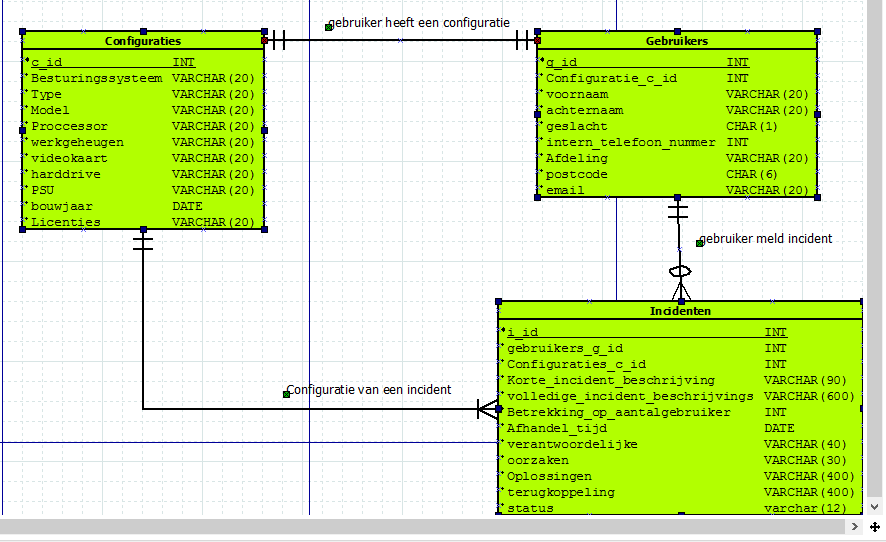
1. Voeg de 3e normaalvormen samen tot een 3e normaalvorm.
2. Maak op basis van de samenvoeging het EntiteitRelatieDiagram (ERD).

Zorg ervoor dat duidelijk naar voren komt:

* + welke entiteiten gebruikt worden;
  + welke eigenschappen of attributen gebruikt worden en bij welke entiteiten zij horen;
  + welke sleutels gebruikt worden;
  + welke relaties er zijn.

1. Maak op basis van het ERD de datadictionary.

Gebruik daarvoor per entiteit het onderstaande schema.



|  |  |  |
| --- | --- | --- |
|  | Incidenten | |
| Definitie |  | |
| Attribuutnaam | Datatype | Toelichting |
| I\_id | INT | id |
| Gebruikers\_g\_id | INT | id |
| Configuratie\_c\_id | INT | id |
| Korte\_incident\_beschrijving | VARCHAR(90) | Korte beschrijving incident |
| Volledige\_incident\_beschrijving | VARCHAR(600) | Volledige omschrijving incident |
| Betrekking\_op\_aantalgebruiker | INT | Aantal betrokken gebruikers |
| Afhandel\_tijd | DATE | afhandelduur |
| verantwoordelijke | VARCHAR(40) | Persoon die er aan werkt |
| oorzaken | VARCHAR(30) | Oorzaak probleem |
| oplossingen | VARCHAR(400) | Oplossing voor het probleem |
| terugkoppeling | VARCHAR(400) |  |
| status | VARCHAR(10) | Status van het probleem |
| Relatie | Toelichting | |
| Configuraties | … | |
| Gebruikers | … | |