

Technisch ontwerp

Asset tracking

Maart 2020

# 

# 

# Voorwoord

Dit technisch ontwerp bevat informatie over de technische aspecten van ons project: asset tracking.

Huidige situatie: er is bijvoorbeeld een botenverhuurbedrijf en die wordt opgebeld door een klant dat ze motor pech hebben. Het verhuurbedrijf wil een monteur het water op sturen om de motor te repareren of om de boot naar de haven te slepen. De monteur moet uren zoeken op het water naar een stilliggende boot omdat de locatie omschrijving van de klanten erg vaag was.

Gewenste situatie: botenverhuurbedrijf kan zo kijken waar een boot ligt zodat hij gelijk de monteur naar de juiste locatie kan sturen. Op die manier hoeft er niet meer gezocht worden naar de boot maar is het duidelijk waar die zich bevindt. Dat bespaart geld en tijd.

In dit voorbeeld gebruik ik botenverhuurbedrijf om een duidelijke schets te maken van hoe het product in de praktijk gebruikt zou worden. Het product is natuurlijk niet alleen maar werkbaar met boten. De asset tracker kan ook werken met auto’s, mensen, dieren, fietsen etcetera.

# 

# Inhoudsopgave

[**Voorwoord**](#_t4ebjjtroxpo) **1**

[**Inhoudsopgave**](#_kcgtui27zbj2) **2**

[**Samenvatting**](#_e0u8nlwvtsto) **4**

[**Plan van aanpak**](#_dhtpx9b1jqxz) **4**

[Fase 1 Onderzoeksfase](#_49m6wt282b0j) 4

[Fase 2 Projectplan](#_1qlz591pxbqq) 4

[Fase 3 Functioneel & technische ontwerp](#_r36b0vd3d2rm) 5

[Fase 4: Sprint 1](#_1f489vvnvnd4) 5

[Fase 5: Sprint 2](#_ul420qyap9oy) 6

[Fase 6: Sprint 3](#_lzn29v5nq6cj) 6

[Fase 7: Sprint 4](#_rcfd33s3dj1p) 6

[Fase 8: Sprint 5](#_cpq7x2n5diay) 7

[Fase 9: Sprint 6](#_514qockp4ow1) 7

[Fase 10: Eindgesprek](#_neo7k1uf1vrn) 7

[**Ontwikkelomgeving**](#_pvb3z420aagr) **7**

[Technische infrastructuur](#_wv2x5ub2fk6t) 7

[Uitleg sleutels](#_hajhgkq2kz9q) 7

[Normalisatie](#_l7glbosbo1m1) 11

[Sequentiediagrammen](#_cnfga2dpwqa0) 12

[Level: algemeen](#_fj6youhnofl) 12

[Account aanmaken](#_6d1twjtrpzc2) 12

[Login](#_2dn7ykhbu3lo) 14

[Uitloggen](#_u1930jfc41cg) 16

[Level: gebruiker](#_eq28t1tm8chk) 17

[Asset toevoegen](#_70eamlusmdc8) 17

[Asset bewerken](#_euzpdewwrg6l) 19

[Asset verwijderen](#_7q3cqediy4wo) 21

[Asset tonen](#_djk34tdeupm2) 22

[Op home scherm](#_9pbsez1sz28q) 22

[Op leaflet pagina](#_o62j4wffqce) 23

[Asset afgelegde route tonen](#_kpghgl3neuy) 24

[Profiel bewerken](#_z4n3m863pwan) 25

[Level: admin](#_tgq02zine8xo) 27

[Gebruiker aanmaken](#_yspwtsewlyww) 27

[Gebruiker bewerken](#_yw7xwg6oousy) 29

[Gebruiker verwijderen](#_kpthqh4ysrr4) 31

[Gebruiker tonen](#_fb64a1kjlt8o) 32

[Activiteitendiagrammen](#_7wu9hj1ijhio) 33

[Level: algemeen](#_cne26spgt1h2) 33

[Account aanmaken](#_iaj0xaibime1) 33

[Inloggen](#_vmpv88a5wre5) 36

[Uitloggen](#_hu4iz0kho8sj) 38

[Level: gebruiker](#_bk6v1hc70dxf) 39

[Asset toevoegen](#_uodrov3krvm9) 39

[Asset bewerken](#_81fbwbyg2djg) 41

[Asset verwijderen](#_fgna9vgm1tcj) 43

[Asset tonen](#_m8lhqddat7bf) 44

[Op homepagina](#_ykmgef38zj3g) 44

[Op leaflet map](#_bpnau76rmqdn) 45

[Asset afgelegde route tonen](#_gdczu9on2oi7) 46

[Gebruiker bewerken](#_w0ykuxw182tg) 47

[Level: admin](#_scl223nknquj) 49

[Gebruiker aanmaken](#_fqyb1t84ec6t) 49

[Gebruiker bewerken](#_z5kp4tpaicf7) 52

[Gebruiker verwijderen](#_gf0k85p2jea2) 55

[Gebruiker tonen](#_5kef9wdpqcxd) 56

[Klassendiagramm](#_4dt5uwooo0v2) 56

[ERD](#_9li856aao7cr) 57

[Ontwikkeltools](#_ygzrhvbzb8b2) 57

[**Beveiliging**](#_jowl09jbbem) **58**

[Autorisatie](#_vf5zohc3fb7r) 58

[Ongewenst gebruik van de applicatie](#_fnedvyzdn4qi) 58

[**Beheer**](#_3hj9ju3zwsy4) **58**

[Back-up](#_b4ddhensczfo) 58

[Content](#_kzcl21a8thk7) 59

# 

# Samenvatting

Ik ga een systeem maken waar gebruikers hun assets kunnen tracken. Het systeem zal communiceren met GPS trackers zodat de locatie van de GPS tracker getoond kan worden op een wereldkaart. Op die manier krijg je een systeem waar gebruikers gemakkelijk zijn/haar eigendommen kunnen volgen.

Het systeem is het beste uit te leggen met een duidelijk voorbeeld van hoe het project tot de praktijk gebracht zou kunnen worden. Je hebt een bedrijf in het noorden van Nederland en werknemers gaan met een auto van het werk naar het zuiden om een klant te helpen. Om ze dan nog wel een beetje te kunnen controleren wil je kijken waar ze zich bevinden. Met de Asset tracking is dat mogelijk. Je kunt naar een website gaan waar de auto zijn locatie getoond wordt op de wereldkaart. Op die manier kun je als baas je werknemers toch nog in de gaten houden als ze ver weg zijn.

# 

# Plan van aanpak

Dit is een opsomming van alle mijlpaalproducten. In de [planning](https://docs.google.com/spreadsheets/d/1516glDooZqnw4MMh0REchuEw1doTuStN_Aq4DitDpNg/edit?usp=sharing) staat wanneer de documenten worden opgeleverd

## Fase 1 Onderzoeksfase

Lijstje van GPS trackers met voor en nadelen. Op die manier krijgt de opdrachtgever een duidelijke beeld van de keuzes qua GPS trackers. Vervolgens kan de opdrachtgever gemakkelijk een keuze maken tussen de GPS trackers.

## Fase 2 Projectplan

Het projectplan is samengesteld om goed in kaart te brengen wat mijn opdracht is voor het maken van de asset tracking ten behoeve van bedrijven (en particulieren). In dit projectplan leg ik voor mijn opdrachtgevers Arno Flapper, Cees Nieboer, Frans de Boer en Richard kingma (hierna te noemen: opdrachtgevers) vast waar dit projectplan betrekking op heeft.

## Fase 3 Functioneel & technische ontwerp

Het technisch ontwerp bevat informatie over de technische aspecten van ons project: de Updatetracker.

Het functioneel ontwerp bevat informatie over de functionele aspecten van ons project: de Update Tracker.

## Fase 4: Sprint 1

In fase 4 ga ik bezig met het maken van paginas. Ik zal de paginas zowel stylen als functioneel maken. In fase 4 maak ik de volgende paginas:

* Inlog pagina
  + Inloggegevens valideren d.m.v vergelijking gegevens database
  + Functioneel werkend
  + Styling
* Account aanmaken pagina
  + Account gegevens valideren d.m.v. vergelijking gegevens database
  + Functioneel werkend
  + Styling
* Home scherm
  + Gebruiker zijn assets tonen in tabel
  + Functioneel werkend
  + Styling
* Kaarten pagina
  + Functioneel werkend gemaakt in Fase 7: Sprint 4
  + Styling
* Bewerk asset pagina
  + Asset gegevens updaten in database
  + Functioneel werkend
  + Styling
* Bewerk profiel pagina
  + Account gegevens valideren en doorvoeren naar database
  + Functioneel werkend
  + Styling
* Voeg asset toe pagina
  + Asset gegevens controleren d.m.v. API
  + Styling asset toevoegen
* Home scherm admin
  + Systeem zijn gebruikers tonen in tabel
  + Functioneel werkend
  + Styling
* Bewerk gebruiker pagina admin
  + Gebruiker gegevens updaten in database
  + Functioneel werkend
  + Styling
* Bewerk profiel pagina admin
  + Account gegevens valideren en doorvoeren naar database
  + Functioneel werkend
  + Styling
* Voeg gebruiker toe pagina admin
  + Gebruiker gegevens controleren
  + Styling asset toevoegen

## Fase 5: Sprint 2

In sprint 2 maak ik het testplan, test scenarios en het testrapport in sprint 2. Het project zal niet getest worden in sprint 2 dat gebeurd namelijk in Fase 8: Sprint 5.

## Fase 6: Sprint 3

In sprint 3 maak ik het testplan, test scenario's en het testrapport voor de acceptatietest . Het project zal niet getest worden in sprint 3 dat gebeurd namelijk in Fase 8: Sprint 5.

Test documentatie maken

## Fase 7: Sprint 4

In sprint 4 maak ik de kaarten pagina functioneel werkend. Met behulp van de GPS API zal de gebruiker zijn assets getoond worden op de pagina. Ook zal de gebruiker zijn de afgelegde route kunnen bekijken van zijn asset.

## Fase 8: Sprint 5

In sprint 5 zal het project getest worden door een aangewezen testteam. Indien nodig wordt er gehertest en dus ook documentatie herschreven. Dat zal gebeuren totdat het project geen fouten meer heeft. Ook zal in sprint 5 de acceptatietest worden uitgevoerd. Indien de opdrachtgever niet tevreden is wordt het project aangepast en zal er nog een keer getest worden.

## Fase 9: Sprint 6

In sprint 6 maak ik de opleverdocumentatie. Denk aan Rapport onderhoud applicatie en Rapportgegevens documenteren en archiveren. Ook zal ik de puntjes op de i zetten qua styling van de paginas.

## Fase 10: Eindgesprek

In de laatste fase heb ik een eindgesprek met de opdrachtgever en word ik beoordeeld op de kwaliteit van mijn opgeleverde product. Ook zal ik mijn project presenteren.



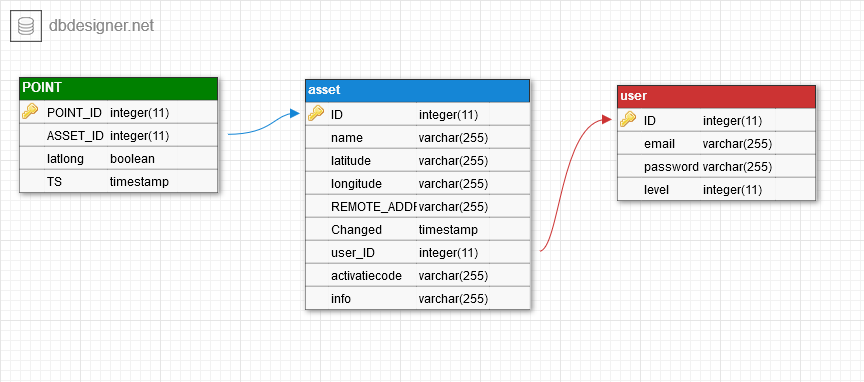
# Ontwikkelomgeving

## Technische infrastructuur

Hieronder is een afbeelding die het databasemodel beschrijft. Het laat de relaties zien van de database. Ook staat er bij wat voor data type het is. Er is te zien dat de tabel ‘location’ en ‘Track’ een connectie met de tabel ‘user’ hebben via de kolom ‘ user\_ID’ en ‘ID’ . Verder heeft de tabel ‘POINT’ een connectie met de tabel ‘Track’ via de kolom ‘Track\_ID’. De pijlen geven de relaties tussen de tabellen aan.

### Uitleg sleutels

* ‘location’ bevat 1 sleutel uit andere tabellen namelijk ‘user\_ID’ die koppelt welke user een GPS heeft.
* De tabel ‘Track’ bevat 1 sleutel uit andere tabellen namelijk ‘user\_ID’ die koppelt de user aan de afgelegde route.
* De tabel ‘POINT’ bevat 1 sleutel uit andere tabellen namelijk ‘TRACK\_ID’ die koppelt afgelegde route aan de coördinaten die bij de route horen.



Verder leg ik nog uit welke data er in de tabellen komt te staan.

user:

|  |  |  |
| --- | --- | --- |
| Name | Type | Beschrijving |
| ID | Int(11) | ID van Users |
| email | Varchar(255) | Het email address van de user |
| Password | varchar(255) | Het wachtwoord van de user. Het wachtwoord wordt van te voren geencrypt door het systeem met de php functie password\_hash. |
| Level | Integer(11) | Hier komt het level te staan van een user. Het is later nog uit te breiden met een role tabel zodat je meerdere levels kunt inbouwen. |

asset:

|  |  |  |
| --- | --- | --- |
| Name | Type | Beschrijving |
| ID | Int(11) | ID van GPS tracker |
| name | Varchar(255) | De naam gegeven door de gebruiker voor de GPS tracker/ asset |
| latitude | varchar(255) | De huidige latitude locatie van de GPS locatie. Als het niet de huidige is dan is het de laatst bekende. |
| Longitude | varchar(255) | De huidige longitude locatie van de GPS locatie. Als het niet de huidige is dan is het de laatst bekende. |
| REMOTE\_ADDR | varchar(255) | Remote address van gebruiker waarmee hij de website bezoekt. |
| Changed | timestamp | Wanneer de row wordt aangepast veranderd de timestamp om veranderingen bij te kunnen houden |
| user\_ID | int(11) | Koppel ID voor tabel user via deze ID weet het systeem van wie de GPS tracker/ asset is. |
| activatiecode | varchar(255) | Code die je krijgt bij het aanschaffen van GPS tracker. Deze code is nodig om ermee te communiceren. |
| info | Varchar(255) | Tekst waar gebruiker nog eigen info kan toevoegen om de GPS tracker / asset te omschrijven |

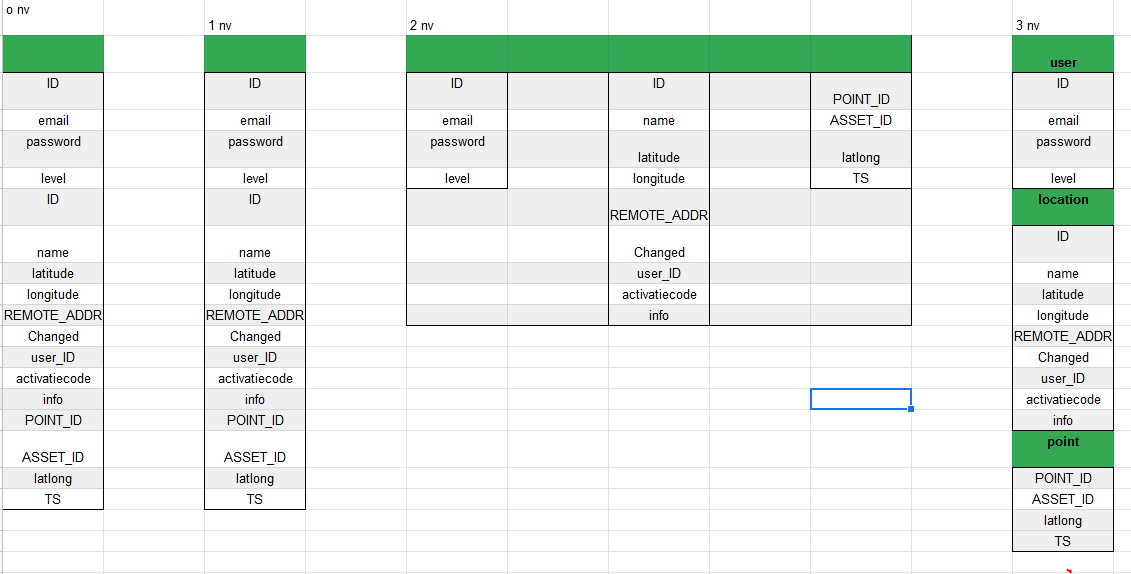
point:

|  |  |  |
| --- | --- | --- |
| Name | Type | Beschrijving |
| POINT\_ID | Int(11) | ID van lat long punt |
| ASSET\_ID | int(11) | Koppel ID voor tabel asset via deze ID weet het systeem bij welke asset het bijbehorende punt hoort |
| latlong | geometry | Latitude, Longitude punt wordt opgeslagen zodat de route uitgestippeld kan worden |
| TS | timestamp | Timestamp dient voor het vastleggen van tijdstip van wanneer GPS tracker op die locatie was |

### 

### Normalisatie

Normalisatie is een techniek bij het ontwerpen van databases. Het heeft twee doelen: opslagruimte besparen en voorkomen dat we meerdere keren dezelfde data vastleggen.



1e nv: Herhalende groepen worden in een aparte tabel gezet (hier niet het geval).

2e nv: Er zijn afzonderlijke tabellen gemaakt voor sets met waarden die op meerdere records van toepassing zijn.

3e nv: Velden die niet belangrijk zijn voor de sleutel zijn geëlimineerd.

## 

## 

## Sequentiediagrammen

Hieronder zie je de verschillende sequentiediagrammen die de interactie tussen objecten laat zien. Door deze diagrammen wordt het duidelijk welke stappen het systeem volgt als het systeem een functie uitvoert.

### Level: algemeen

Het systeem kent meerdere levels oftewel rollen. Level: gebruiker heeft toegang tot andere functionaliteiten dan level: admin. Level algemeen is de categorie waar zowel level admin als level gebruiker bij kan.

#### Account aanmaken

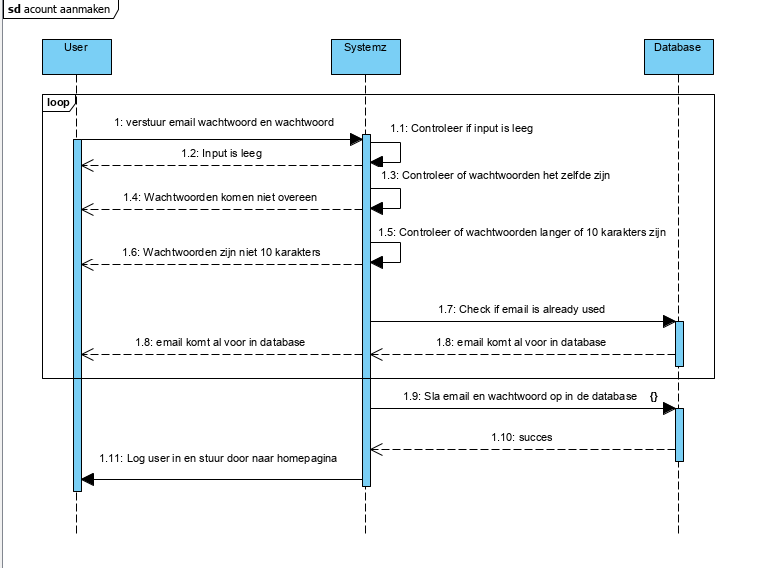
De gebruiker voert zijn e mail, wachtwoord en wachtwoord in. Vervolgens drukt de gebruiker op sign up. Als de gebruiker zijn gegevens goed invult wordt zijn account aangemaakt en wordt de gebruiker doorgestuurd naar de homepagina.

De stappen die de gebruiker neemt

1. Gebruiker verstuurt email, wachtwoord en wachtwoord naar systeem
2. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
3. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
4. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
5. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
6. Als alle gegevens goedgekeurd zijn slaat het systeem het email adres en het wachtwoord op van de gebruiker.
7. Systeem maakt sessie aan en stuurt gebruiker naar homepagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord



#### 

#### Login

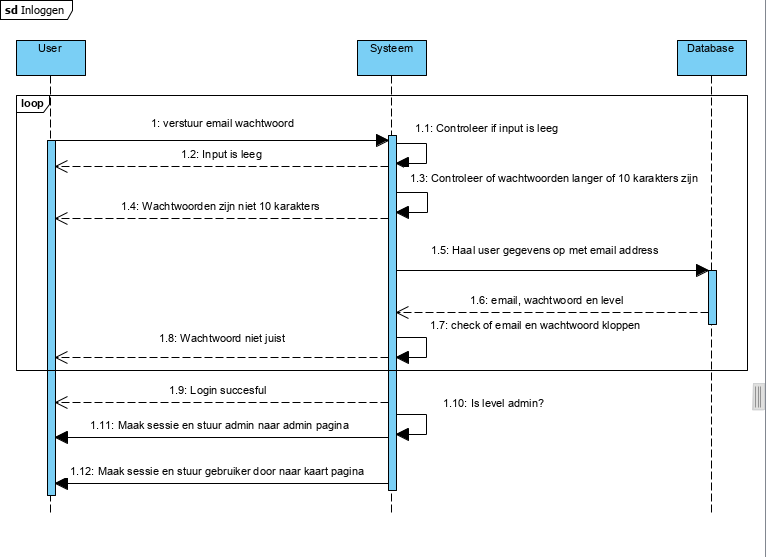
Als de gebruiker in het bezit is van een account kan hij/zij zijn e mailadres en wachtwoord invullen. Vervolgens kan de gebruiker op login drukken als de gegevens juist zijn dan wordt er een sessie aangemaakt en kan de gebruiker in het systeem. Anders krijgt de gebruiker een foutmelding.

De stappen die de gebruiker neemt

1. Gebruiker verstuurt email, wachtwoord naar systeem
2. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
3. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
4. Systeem haalt user gegevens van ingevulde email address uit database
5. Systeem controleert of email en wachtwoord kloppen
   1. Zo niet “wachtwoord niet juist”
6. Als alle gegevens correct zijn “login successful”
7. Systeem kijk of level ‘admin’ is
   1. Zo ja maak sessie en stuur door naar admin pagina
8. Systeem maakt sessie aan en stuurt gebruiker naar kaartpagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters



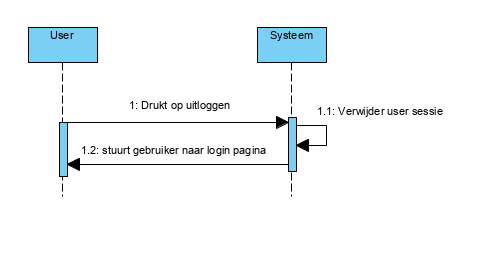
#### 

#### Uitloggen

Als de gebruiker is ingelogd dan kan de gebruiker op de uitlog knop drukken. Vervolgens beëindigt het systeem de sessie en stuurt het systeem de gebruiker door naar de inlogpagina.

De stappen die de gebruiker neemt

1. Gebruiker drukt op uitloggen
2. Systeem verwijderd user sessie
3. Systeem stuurt gebruiker naar inlogpagina.



### 

### Level: gebruiker

Het systeem kent meerdere levels oftewel rollen. Level: gebruiker heeft toegang tot andere functionaliteiten dan level: admin.

#### Asset toevoegen

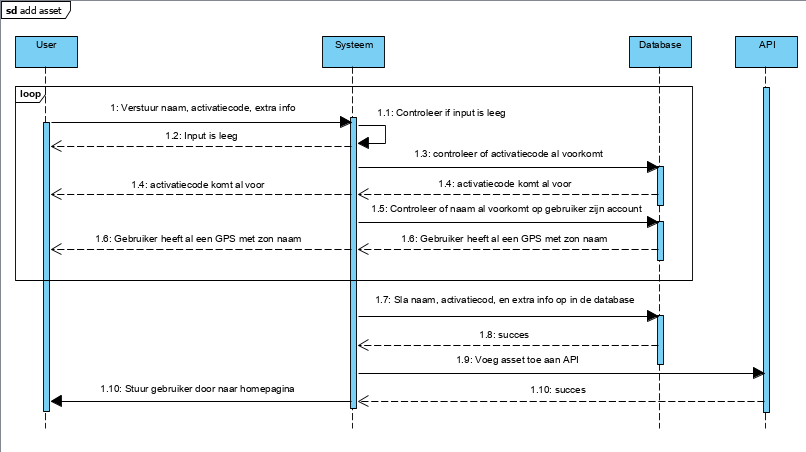
De gebruiker drukt op ‘voeg asset toe’. Er komt een form tevoorschijn waar de gebruiker de asset zijn naam, activatiecode en overige info invult. Vervolgens drukt de gebruiker op ‘add’. Als de gegevens kloppen wordt de asset toegevoegd aan het systeem. Anders krijgt de gebruiker een foutmelding

De stappen die de gebruiker neemt.

1. Gebruiker verstuurt naam, activatiecode en extra info naar systeem
2. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
3. Systeem controleert of activatiecode al voorkomt in database
   1. Zo ja “activatiecode komt al voor”
4. Systeem controleert of naam al voorkomt in database
   1. Zo ja “Gebruiker heeft al een GPS met zon naam”
5. Als alle gegevens goedgekeurd zijn slaat het systeem de naam, activatiecode en de extra info op van de asset.
6. Systeem stuurt gebruiker naar homepagina.

De eisen waar de asset gegevens aan moeten voldoen

* Asset name
  + Moet uniek zijn op de gebruiker zijn account
  + Mag niet leeg zijn
* Activatie code
  + Mag niet door een andere gebruiker al gebruikt worden
  + Mag niet leeg zijn
* Other GPS info
  + Mag niet leeg zijn



#### 

#### Asset bewerken

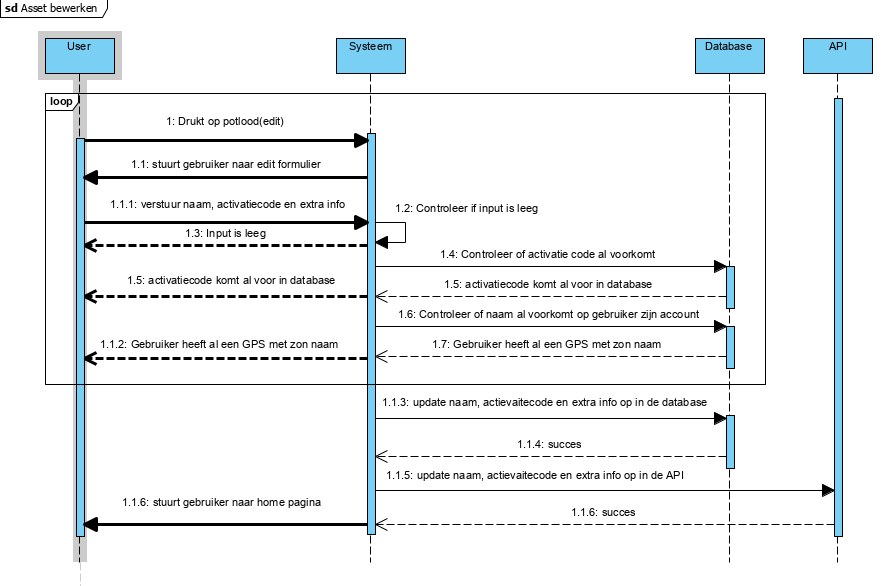
De gebruiker drukt op ‘edit (het potlood)’. Gebruiker wordt doorgestuurd naar een invulformulier. Daar kan de gebruiker de asset name, activatiecode en de extra info aanpassen. Vervolgens drukt de gebruiker op ‘edit’. Als de gegevens kloppen wordt de asset aangepast. Anders krijgt de gebruiker een foutmelding.

De stappen die de gebruiker neemt

1. Gebruiker drukt op potlood (edit)
2. Systeem stuurt gebruiker door naar edit formulier
3. Gebruiker verstuurt naam, activatiecode en extra info naar systeem
4. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
5. Systeem controleert of activatiecode al voorkomt in database
   1. Zo ja “activatiecode komt al voor”
6. Systeem controleert of naam al voorkomt in database
   1. Zo ja “Gebruiker heeft al een GPS met zon naam”
7. Als alle gegevens goedgekeurd zijn slaat het systeem de naam, activatiecode en de extra info op van de asset.
8. Systeem stuurt gebruiker naar homepagina.

De eisen waar de asset gegevens aan moeten voldoen

* Asset name
  + Moet uniek zijn op de gebruiker zijn account
  + Mag niet leeg zijn
* Activatie code
  + Mag niet door een andere gebruiker al gebruikt worden
  + Mag niet leeg zijn
* Other GPS info
  + Mag niet leeg zijn



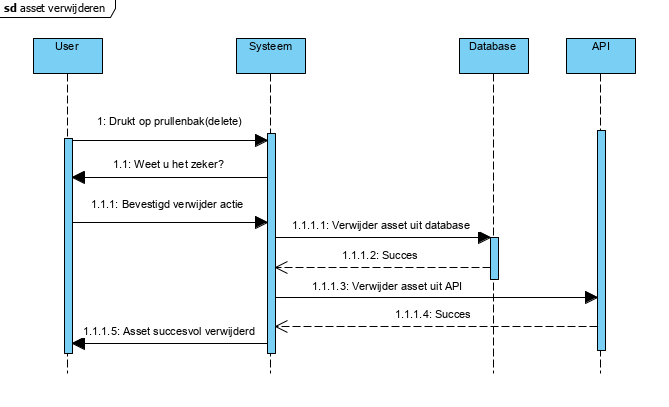
#### 

#### Asset verwijderen

De gebruiker drukt op ‘delete (de prullenbak)’. Gebruiker wordt gevraagd of hij het zeker weet. Als de gebruiker bevestigd wordt de asset verwijderd. Anders wordt de asset niet verwijderd.

De stappen die de gebruiker neemt

1. Gebruiker drukt op prullenbak (delete)
2. Systeem stuurt gebruiker de vraag “weet u het zeker?”
3. Gebruiker bevestigt verwijder actie
4. Systeem verwijderd asset uit database
5. Systeem “asset succesvol verwijderd”



#### 

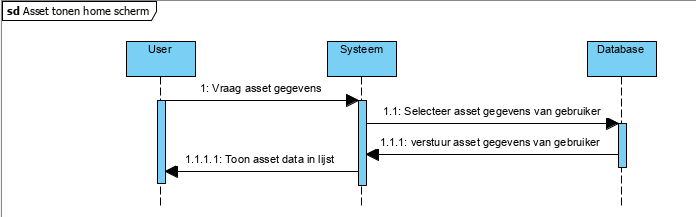
#### Asset tonen

##### Op home scherm

Als de gebruiker is ingelogd kan de gebruiker zijn assets bekijken op het home scherm. Vanuit het home scherm heeft de gebruiker de mogelijkheid om meer assets toe te voegen of om assets te bewerken/ verwijderen.

De stappen die de gebruiker neemt

1. Gebruiker komt op home pagina
2. Systeem selecteert asset gegevens van gebruiker
3. Systeem toont asset gegevens van gebruiker in tabel vorm



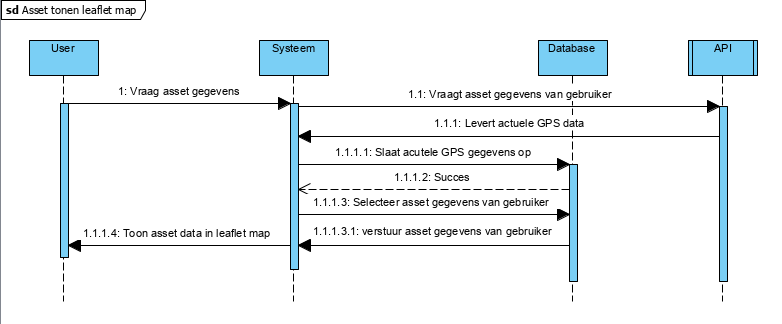
##### 

##### Op leaflet pagina

Als de gebruiker is ingelogd kan de gebruiker zijn assets bekijken op het leaflet map. Vanuit de leaflet map heeft de gebruiker de mogelijkheid om de live locatie van de asset te bekijken.

De stappen die de gebruiker neemt

1. Gebruiker komt op kaartpagina
2. Systeem selecteert asset gegevens van gebruiker
3. Systeem toont asset gegevens van gebruiker in leaflet map



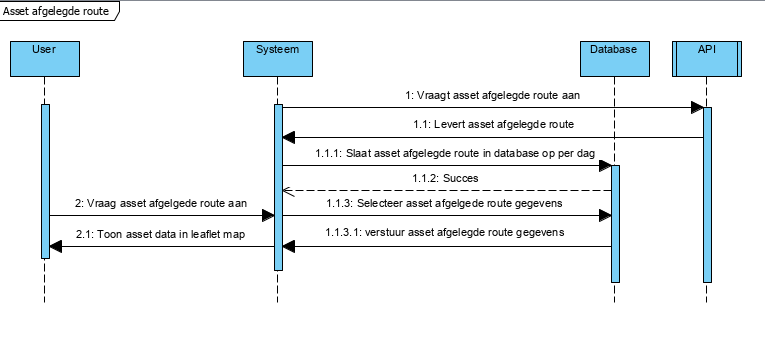
#### 

#### Asset afgelegde route tonen

Als de gebruiker is ingelogd kan de gebruiker zijn assets bekijken op het leaflet map. Vanuit de leaflet map heeft de gebruiker de mogelijkheid om de afgelegde route van een asset.

De stappen die de gebruiker neemt

1. Gebruiker vraag asset afgelegde route aan
2. Systeem selecteert asset afgelegde route gegevens van gebruiker
3. Systeem toont asset afgelegde route gegevens van gebruiker in leaflet map



#### 

#### Profiel bewerken

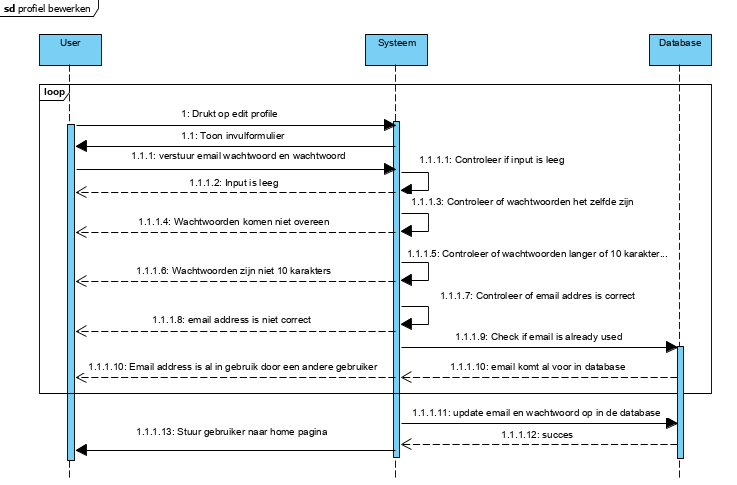
De gebruiker drukt op ‘edit profile’. Gebruiker krijgt een invulformulier voor zich. Daar kan de gebruiker zijn e mailadres en wachtwoord aanpassen. Vervolgens drukt de gebruiker op ‘update’. Als de gegevens kloppen wordt de gebruiker zijn accountgegevens aangepast. Anders krijgt de gebruiker een foutmelding

De stappen die de gebruiker neemt

1. Gebruiker drukt op edit profile
2. Systeem toont invulformulier
3. Gebruiker verstuurt email, wachtwoord en wachtwoord
4. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
5. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
6. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
7. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
8. Systeem controleert of email address valiede is
   1. Zo niet “email address is niet correct”
9. Als alle gegevens goedgekeurd zijn update het systeem het email adres en het wachtwoord in de database
10. Systeem stuurt gebruiker naar homepagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord



### 

### Level: admin

Het systeem kent meerdere levels oftewel rollen. Level: admin heeft toegang tot andere functionaliteiten dan level: gebruiker.

#### Gebruiker aanmaken

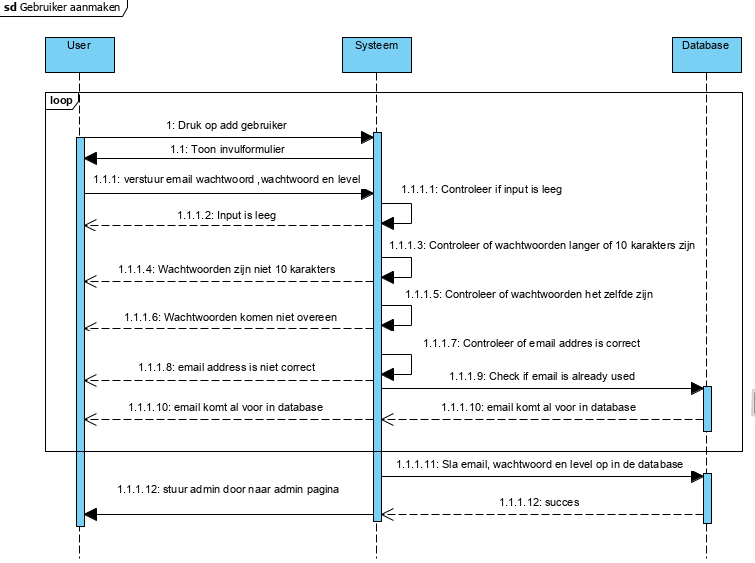
De admin drukt op ‘voeg gebruiker toe’ en voert een gebruiker zijn e mail, wachtwoord en wachtwoord in. Vervolgens drukt de admin op ‘add’. Als de admin de gegevens goed invult wordt de gebruiker zijn account aangemaakt en wordt de admin doorgestuurd naar de homepagina.

De stappen die de admin neemt

1. Admin drukt op ‘add gebruiker’
2. Systeem toont invulformulier
3. Admin verstuurt email, wachtwoord, wachtwoord en level naar systeem
4. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
5. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
6. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
7. Systeem controleert of email address valiede is
   1. Zo niet “email address is niet correct”
8. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
9. Als alle gegevens goedgekeurd zijn slaat het systeem het email adres, wachtwoord en het level op in de database.
10. Systeem stuurt admin naar admin pagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord
* Level
  + Wanneer niet aangevinkt normale gebruiker anders admin



#### 

#### Gebruiker bewerken

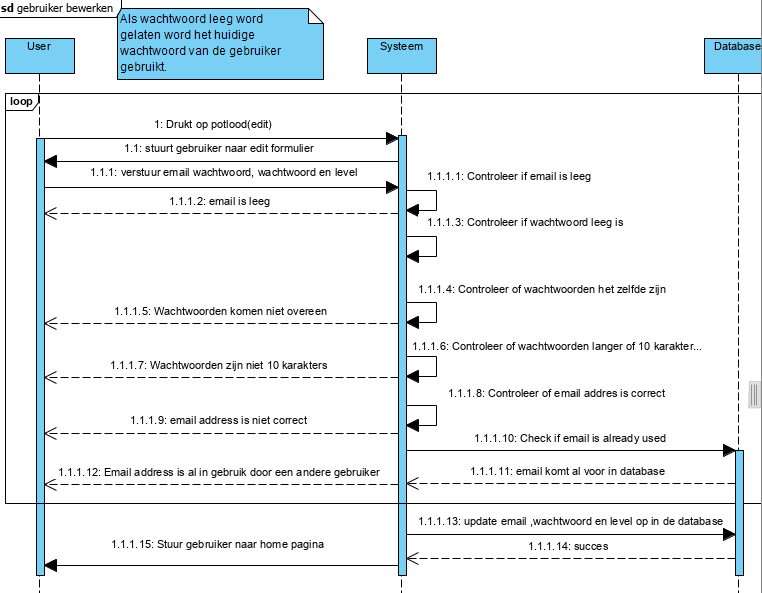
De admin drukt op ‘edit (het potlood)’. Admin wordt doorgestuurd naar een invulformulier. Daar kan de admin het email adres, wachtwoord en het level aanpassen. Vervolgens drukt de admin op ‘edit’. Als de gegevens kloppen wordt de gebruiker aangepast. Anders krijgt de admin een foutmelding.

De stappen die de admin neemt

1. Admin drukt op potlood(edit)
2. Systeem stuurt admin naar edit formulier
3. Admin verstuurt email, wachtwoord, wachtwoord en level
4. Systeem controleert of email leeg is
   1. Zo ja “email is leeg”
5. Systeem controleert of wachtwoorden leeg zijn
   1. Zo ja gebruik huidige wachtwoord van gebruiker
6. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
7. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
8. Systeem controleert of email address valiede is
   1. Zo niet “email address is niet correct”
9. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
10. Als alle gegevens goedgekeurd zijn update het systeem het e mailadres, het wachtwoord en het level van de gebruiker in de database
11. Systeem stuurt admin naar homepagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Als het veld wordt leeg gelaten gebruikt het systeem het huidige wachtwoord van het account. Op die manier kan de admin gebruikers aanpassen zonder zijn wachtwoord te weten.
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Als het veld wordt leeg gelaten gebruikt het systeem het huidige wachtwoord van het account. Op die manier kan de admin gebruikers aanpassen zonder zijn wachtwoord te weten.
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord
* Level
  + Wanneer niet aangevinkt normale gebruiker anders admin



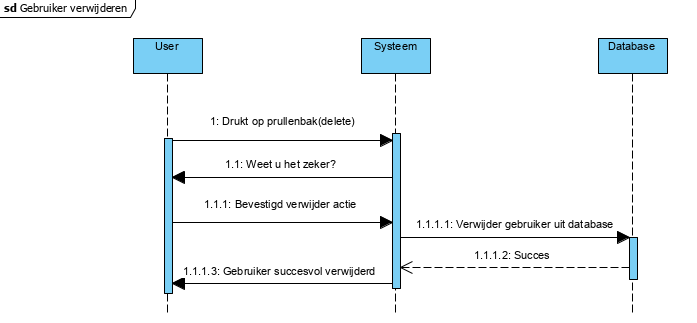
#### 

#### Gebruiker verwijderen

De admin drukt op ‘delete (de prullenbak)’. Admin wordt gevraagd of hij het zeker weet. Als de admin bevestigd wordt de gebruiker verwijderd. Anders wordt de gebruiker niet verwijderd.

De stappen die de admin neemt

1. Admin drukt op prullenbak (delete)
2. Systeem stuurt admin de vraag “weet u het zeker?”
3. Admin bevestigt verwijder actie
4. Systeem verwijderd gebruiker uit database
5. Systeem “gebruiker succesvol verwijderd”



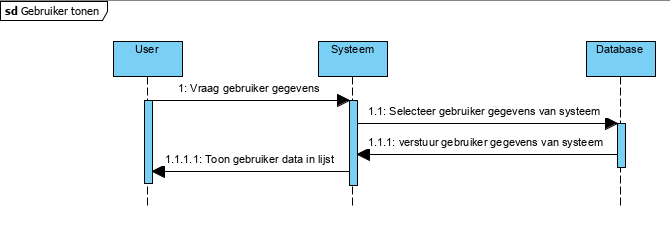
#### 

#### Gebruiker tonen

Als de admin is ingelogd kan de admin de gebruikers bekijken op zijn home scherm. Vanuit het home scherm heeft de admin de mogelijkheid om meer gebruikers toe te voegen of om gebruikers te bewerken/ verwijderen.

De stappen die de admin neemt

1. Admin komt op home pagina
2. Systeem selecteert gebruiker gegevens van systeem
3. Systeem toont gebruiker gegevens van het systeem in tabelvorm



## 

## Activiteitendiagrammen

Hieronder zie je de verschillende activiteitendiagrammen die de interactie tussen objecten laat zien. Door deze diagrammen wordt het duidelijk welke stappen het systeem volgt als het systeem een functie uitvoert.

### Level: algemeen

Het systeem kent meerdere levels oftewel rollen. Level: gebruiker heeft toegang tot andere functionaliteiten dan level: admin. Level algemeen is de categorie waar zowel level admin als level gebruiker bij kan.

#### Account aanmaken

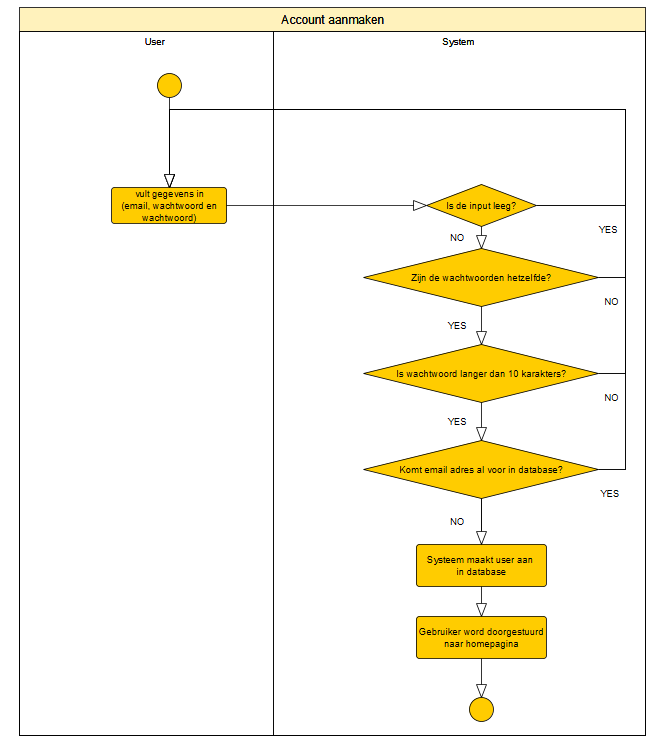
De gebruiker voert zijn e mail, wachtwoord en wachtwoord in. Vervolgens drukt de gebruiker op sign up. Als de gebruiker zijn gegevens goed invult wordt zijn account aangemaakt en wordt de gebruiker doorgestuurd naar de homepagina.

De stappen die de gebruiker neemt

1. Gebruiker verstuurt email, wachtwoord en wachtwoord naar systeem
2. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
3. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
4. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
5. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
6. Als alle gegevens goedgekeurd zijn slaat het systeem het email adres en het wachtwoord op van de gebruiker.
7. Systeem maakt sessie aan en stuurt gebruiker naar homepagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord



#### Inloggen

Als de gebruiker in het bezit is van een account kan hij/zij zijn e mailadres en wachtwoord invullen. Vervolgens kan de gebruiker op login drukken als de gegevens juist zijn dan wordt er een sessie aangemaakt en kan de gebruiker in het systeem. Anders krijgt de gebruiker een foutmelding.

De stappen die de gebruiker neemt

1. Gebruiker verstuurt email, wachtwoord naar systeem
2. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
3. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
4. Systeem haalt user gegevens van ingevulde email address uit database
5. Systeem controleert of email en wachtwoord kloppen
   1. Zo niet “wachtwoord niet juist”
6. Als alle gegevens correct zijn “login successful”
7. Systeem kijk of level ‘admin’ is
   1. Zo ja maak sessie en stuur door naar admin pagina
8. Systeem maakt sessie aan en stuurt gebruiker naar kaartpagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters

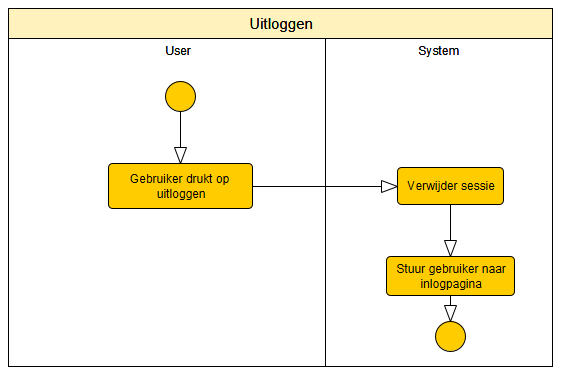
#### 

#### Uitloggen

De stappen die de gebruiker neemt

1. Gebruiker drukt op uitloggen
2. Systeem verwijderd user sessie
3. Systeem stuurt gebruiker naar inlogpagina.

Als de gebruiker is ingelogd dan kan de gebruiker op de uitlog knop drukken. Vervolgens beëindigt het systeem de sessie en stuurt het systeem de gebruiker door naar de inlogpagina.



### 

### Level: gebruiker

#### Asset toevoegen

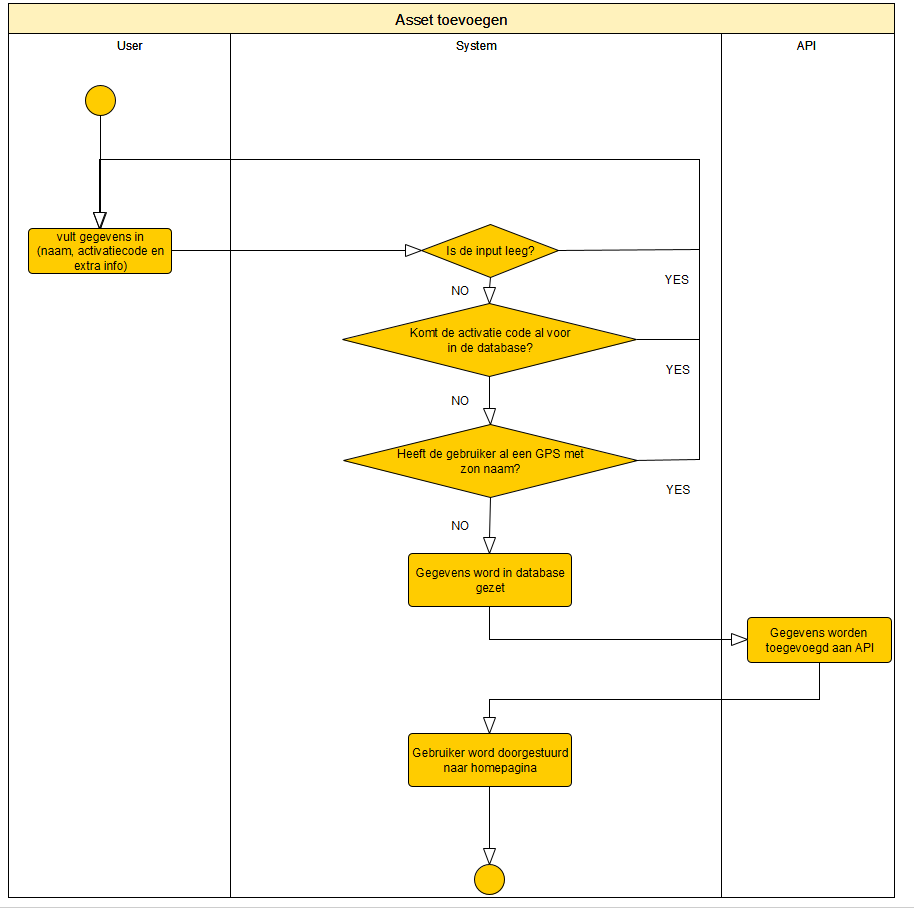
De gebruiker drukt op ‘voeg asset toe’. Er komt een form tevoorschijn waar de gebruiker de asset zijn naam, activatiecode en overige info invult. Vervolgens drukt de gebruiker op ‘add’. Als de gegevens kloppen wordt de asset toegevoegd aan het systeem. Anders krijgt de gebruiker een foutmelding.

De stappen die de gebruiker neemt.

1. Gebruiker verstuurt naam, activatiecode en extra info naar systeem
2. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
3. Systeem controleert of activatiecode al voorkomt in database
   1. Zo ja “activatiecode komt al voor”
4. Systeem controleert of naam al voorkomt in database
   1. Zo ja “Gebruiker heeft al een GPS met zon naam”
5. Als alle gegevens goedgekeurd zijn slaat het systeem de naam, activatiecode en de extra info op van de asset.
6. Systeem stuurt gebruiker naar homepagina.

De eisen waar de asset gegevens aan moeten voldoen

* Asset name
  + Moet uniek zijn op de gebruiker zijn account
  + Mag niet leeg zijn
* Activatie code
  + Mag niet door een andere gebruiker al gebruikt worden
  + Mag niet leeg zijn
* Other GPS info
  + Mag niet leeg zijn



#### 

#### Asset bewerken

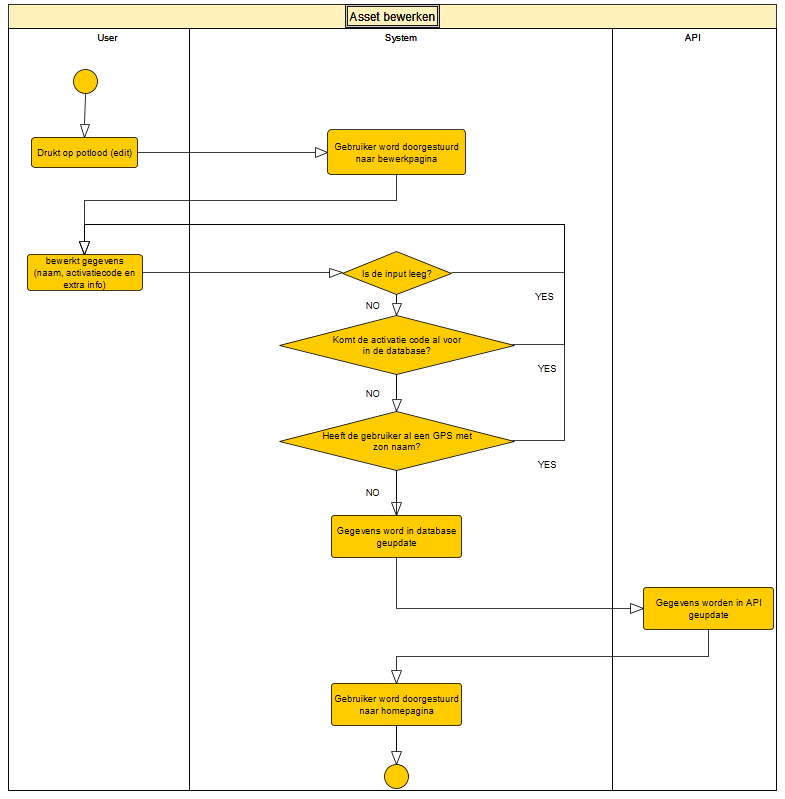
De gebruiker drukt op ‘edit (het potlood)’. Gebruiker wordt doorgestuurd naar een invulformulier. Daar kan de gebruiker de asset name, activatiecode en de extra info aanpassen. Vervolgens drukt de gebruiker op ‘edit’. Als de gegevens kloppen wordt de asset aangepast. Anders krijgt de gebruiker een foutmelding.

De stappen die de gebruiker neemt

1. Gebruiker drukt op potlood (edit)
2. Systeem stuurt gebruiker door naar edit formulier
3. Gebruiker verstuurt naam, activatiecode en extra info naar systeem
4. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
5. Systeem controleert of activatiecode al voorkomt in database
   1. Zo ja “activatiecode komt al voor”
6. Systeem controleert of naam al voorkomt in database
   1. Zo ja “Gebruiker heeft al een GPS met zon naam”
7. Als alle gegevens goedgekeurd zijn slaat het systeem de naam, activatiecode en de extra info op van de asset.
8. Systeem stuurt gebruiker naar homepagina.

De eisen waar de asset gegevens aan moeten voldoen

* Asset name
  + Moet uniek zijn op de gebruiker zijn account
  + Mag niet leeg zijn
* Activatie code
  + Mag niet door een andere gebruiker al gebruikt worden
  + Mag niet leeg zijn
* Other GPS info
  + Mag niet leeg zijn



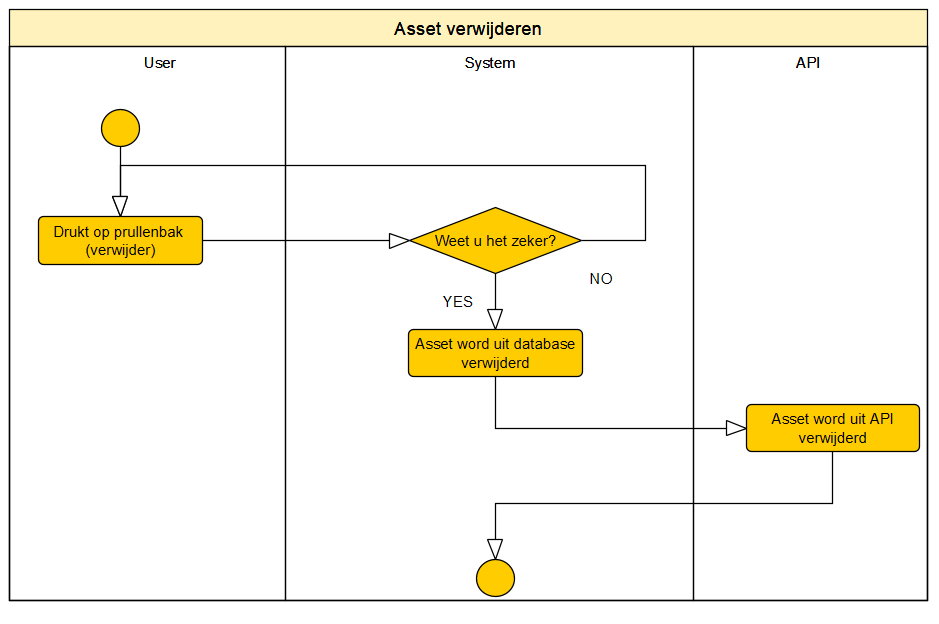
#### 

#### Asset verwijderen

De gebruiker drukt op ‘delete (de prullenbak)’. Gebruiker wordt gevraagd of hij het zeker weet. Als de gebruiker bevestigd wordt de asset verwijderd. Anders wordt de asset niet verwijderd.

De stappen die de gebruiker neemt

1. Gebruiker drukt op prullenbak (delete)
2. Systeem stuurt gebruiker de vraag “weet u het zeker?”
3. Gebruiker bevestigt verwijder actie
4. Systeem verwijderd asset uit database
5. Systeem “asset succesvol verwijderd”



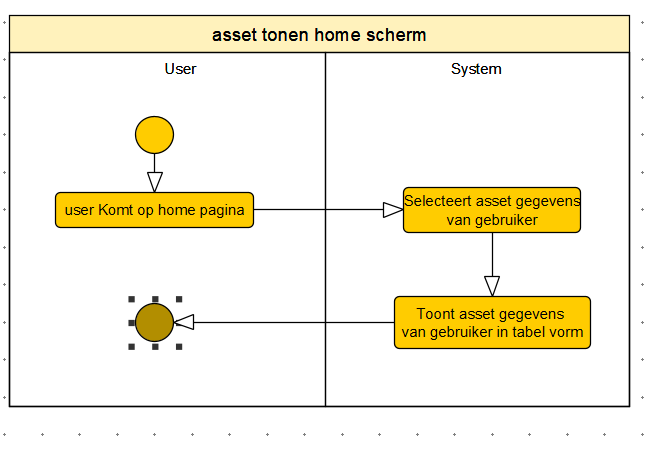
#### Asset tonen

##### Op homepagina

Als de gebruiker is ingelogd kan de gebruiker zijn assets bekijken op het home scherm. Vanuit het home scherm heeft de gebruiker de mogelijkheid om meer assets toe te voegen of om assets te bewerken/ verwijderen.

De stappen die de gebruiker neemt

1. Gebruiker komt op home pagina
2. Systeem selecteert asset gegevens van gebruiker
3. Systeem toont asset gegevens van gebruiker in tabel vorm



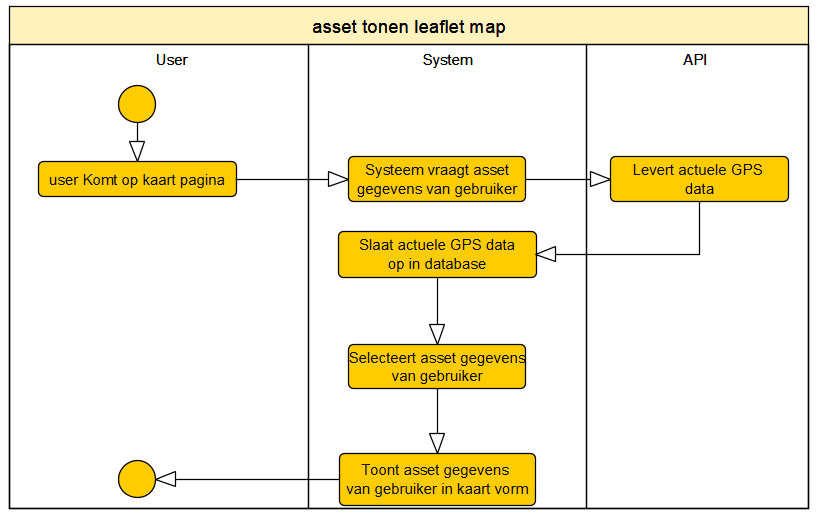
##### 

##### Op leaflet map

Als de gebruiker is ingelogd kan de gebruiker zijn assets bekijken op het leaflet map. Vanuit de leaflet map heeft de gebruiker de mogelijkheid om de live locatie van de asset te bekijken.

De stappen die de gebruiker neemt

1. Gebruiker komt op kaartpagina
2. Systeem selecteert asset gegevens van gebruiker
3. Systeem toont asset gegevens van gebruiker in leaflet map



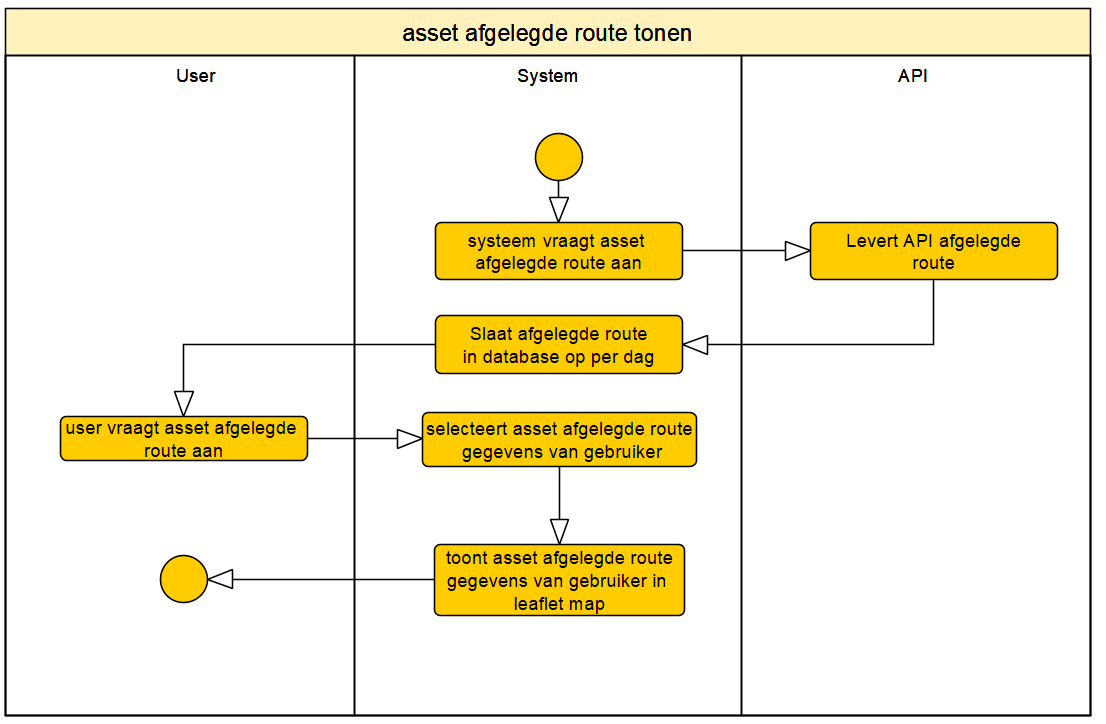
#### 

#### Asset afgelegde route tonen

Als de gebruiker is ingelogd kan de gebruiker zijn assets bekijken op het leaflet map. Vanuit de leaflet map heeft de gebruiker de mogelijkheid om de afgelegde route van een asset.

De stappen die de gebruiker neemt

1. Gebruiker vraag asset afgelegde route aan
2. Systeem selecteert asset afgelegde route gegevens van gebruiker
3. Systeem toont asset afgelegde route gegevens van gebruiker in leaflet map



#### 

#### Gebruiker bewerken

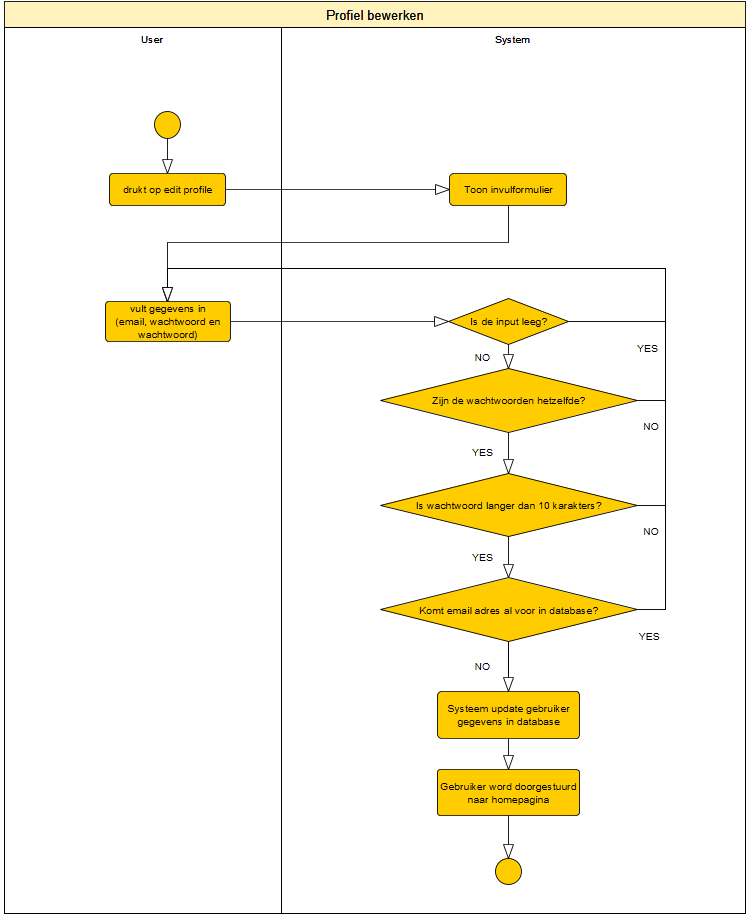
De gebruiker drukt op ‘edit profile’. Gebruiker krijgt een invulformulier voor zich. Daar kan de gebruiker zijn e mailadres en wachtwoord aanpassen. Vervolgens drukt de gebruiker op ‘update’. Als de gegevens kloppen wordt de gebruiker zijn accountgegevens aangepast. Anders krijgt de gebruiker een foutmelding

De stappen die de gebruiker neemt

1. Gebruiker drukt op edit profile
2. Systeem toont invulformulier
3. Gebruiker verstuurt email, wachtwoord en wachtwoord
4. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
5. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
6. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
7. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
8. Systeem controleert of email address valiede is
   1. Zo niet “email address is niet correct”
9. Als alle gegevens goedgekeurd zijn update het systeem het email adres en het wachtwoord in de database
10. Systeem stuurt gebruiker naar homepagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord



### Level: admin

Het systeem kent meerdere levels oftewel rollen. Level: admin heeft toegang tot andere functionaliteiten dan level: gebruiker.

#### Gebruiker aanmaken

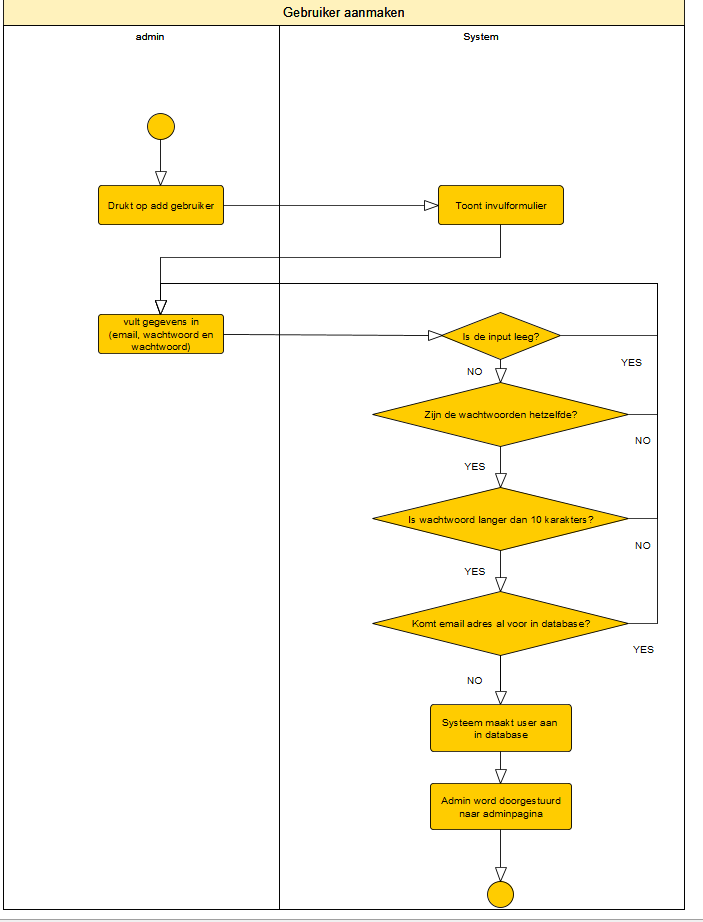
De admin drukt op ‘voeg gebruiker toe’ en voert een gebruiker zijn e mail, wachtwoord en wachtwoord in. Vervolgens drukt de admin op ‘add’. Als de admin de gegevens goed invult wordt de gebruiker zijn account aangemaakt en wordt de admin doorgestuurd naar de homepagina.

De stappen die de admin neemt

1. Admin drukt op ‘add gebruiker’
2. Systeem toont invulformulier
3. Admin verstuurt email, wachtwoord, wachtwoord en level naar systeem
4. Systeem controleert of input leeg is
   1. Zo ja “input is leeg”
5. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
6. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
7. Systeem controleert of email address valiede is
   1. Zo niet “email address is niet correct”
8. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
9. Als alle gegevens goedgekeurd zijn slaat het systeem het email adres, wachtwoord en het level op in de database.
10. Systeem stuurt admin naar admin pagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord
* Level
  + Wanneer niet aangevinkt normale gebruiker anders admin



#### Gebruiker bewerken

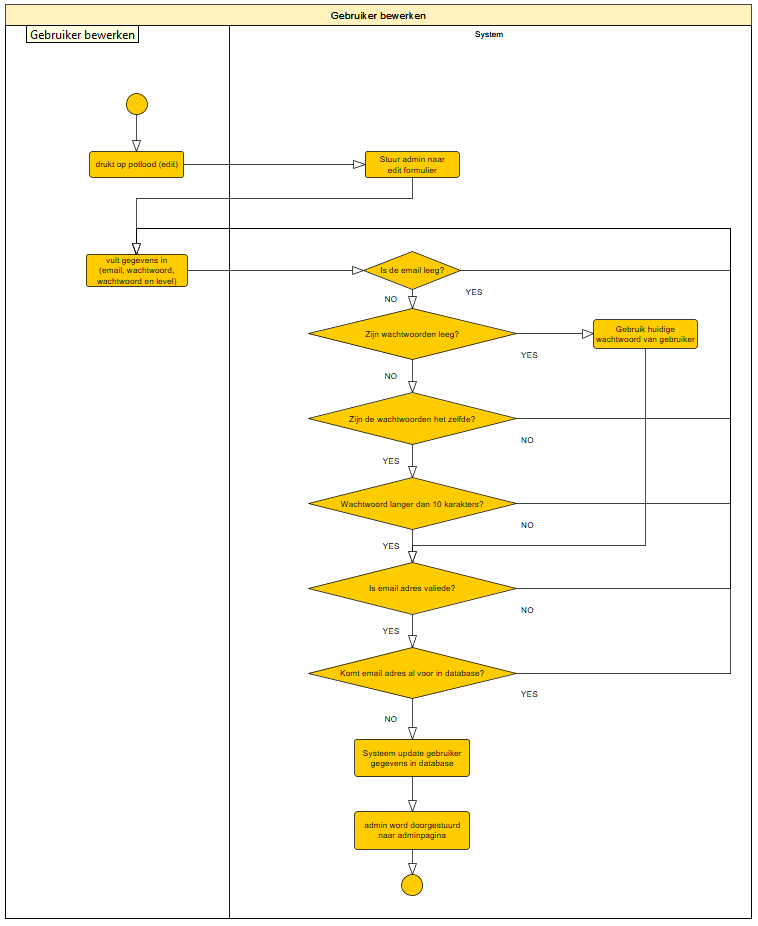
De admin drukt op ‘edit (het potlood)’. Admin wordt doorgestuurd naar een invulformulier. Daar kan de admin het email adres, wachtwoord en het level aanpassen. Vervolgens drukt de admin op ‘edit’. Als de gegevens kloppen wordt de gebruiker aangepast. Anders krijgt de admin een foutmelding.

De stappen die de admin neemt

1. Admin drukt op potlood(edit)
2. Systeem stuurt admin naar edit formulier
3. Admin verstuurt email, wachtwoord, wachtwoord en level
4. Systeem controleert of email leeg is
   1. Zo ja “email is leeg”
5. Systeem controleert of wachtwoorden leeg zijn
   1. Zo ja gebruik huidige wachtwoord van gebruiker
6. Systeem controleert of wachtwoorden hetzelfde zijn
   1. Zo niet “wachtwoorden komen niet overeen”
7. Systeem controleert of wachtwoorden langer of 10 karakters zijn
   1. Zo niet “wachtwoorden zijn niet 10 karakters”
8. Systeem controleert of email address valiede is
   1. Zo niet “email address is niet correct”
9. Systeem controleert of email al voor komt in database
   1. Zo ja “email komt al voor in database”
10. Als alle gegevens goedgekeurd zijn update het systeem het e mailadres, het wachtwoord en het level van de gebruiker in de database
11. Systeem stuurt admin naar homepagina.

De eisen waar de accountgegevens aan moeten voldoen

* Email adres
  + Mag niet in gebruik zijn door een andere gebruiker
  + Moet een email adres zijn
* Wachtwoord
  + Als het veld wordt leeg gelaten gebruikt het systeem het huidige wachtwoord van het account. Op die manier kan de admin gebruikers aanpassen zonder zijn wachtwoord te weten.
  + Langer dan 10 karakters
* Tweede wachtwoord
  + Als het veld wordt leeg gelaten gebruikt het systeem het huidige wachtwoord van het account. Op die manier kan de admin gebruikers aanpassen zonder zijn wachtwoord te weten.
  + Moet hetzelfde zijn als het eerste ingevulde wachtwoord
* Level
  + Wanneer niet aangevinkt normale gebruiker anders admin

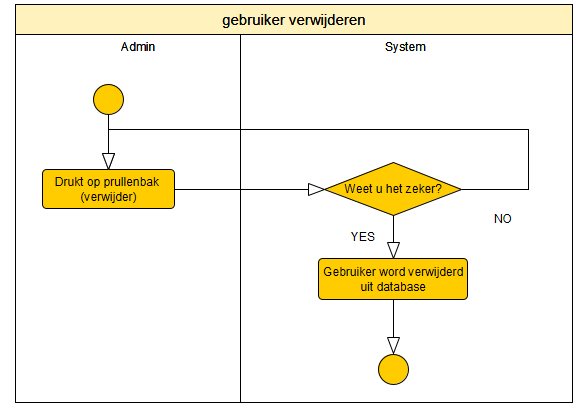


#### Gebruiker verwijderen

De admin drukt op ‘delete (de prullenbak)’. Admin wordt gevraagd of hij het zeker weet. Als de admin bevestigd wordt de gebruiker verwijderd. Anders wordt de gebruiker niet verwijderd.

De stappen die de admin neemt

1. Admin drukt op prullenbak (delete)
2. Systeem stuurt admin de vraag “weet u het zeker?”
3. Admin bevestigt verwijder actie
4. Systeem verwijderd gebruiker uit database
5. Systeem “gebruiker succesvol verwijderd”



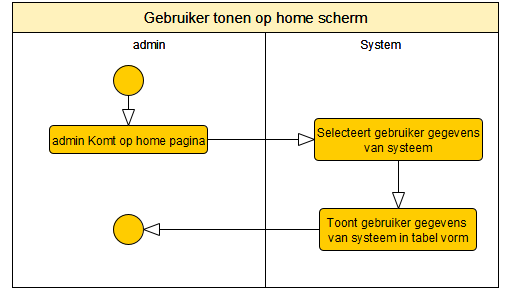
#### 

#### Gebruiker tonen

Als de admin is ingelogd kan de admin de gebruikers bekijken op zijn home scherm. Vanuit het home scherm heeft de admin de mogelijkheid om meer gebruikers toe te voegen of om gebruikers te bewerken/ verwijderen.

De stappen die de admin neemt

1. Admin komt op home pagina
2. Systeem selecteert gebruiker gegevens van systeem
3. Systeem toont gebruiker gegevens van het systeem in tabelvorm



## Klassendiagramm

Ik heb geen klassendiagram want dat is niet van toepassing op mijn project. In het ERD zal goed omschreven worden wat de relaties zijn tussen de software en de user. Daardoor wordt een klassendiagram overbodig.

## 

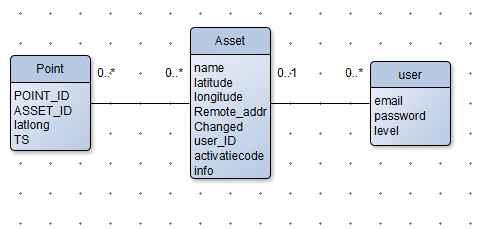
## ERD

Hier bij de ERD laat ik zien wat de relatie is tussen de point, de asset en de user.

* user heeft 0 of meer assets gekoppeld aan zich.
* Asset heeft 0 of 1 user aan zich gekoppeld. En 0 of meer points aan zich gekoppeld.
* Point heeft 0 of meer assets aan zich gekoppeld.

De 0 of meer betekent de ‘0..\*’ in het ERD.

De 0 of 1 betekent de “0..1” in het ERD.



## 

## Ontwikkeltools

Voor het programmeren gebruik ik de code editor Sublime text 3. Ik heb deze code editor gekozen omdat ik er goede ervaring mee heb.

Verder gebruik ik xampp om het project lokaal te testen.

PHPMyAdmin gebruik ik om een database te maken. Die heb ik nodig om data op te slaan.

Om verder het systeem te testen worden zoveel mogelijk web browsers gebruikt. Zoals Chrome, firefox, edge en opera.

Verder gebruik ik een GPS tracker met een werkende simkaart die gekoppeld is aan een API.



# Beveiliging

## Autorisatie

Het systeem is ingedeeld in verschillende levels. Level gebruiker en admin.

* Level gebruiker kan andere dingen binnen het systeem dan level admin. Zo kan level gebruiker alleen maar eigen data aanpassen
* Level admin kan zowel eigen als andermans data aanpassen.

Bij het inloggen wordt gecontroleerd welk level de gebruiker heeft. Als de gebruiker level admin heeft wordt hij doorgestuurd naar de admin pagina. Anders is het een normale gebruiker en wordt de gebruiker doorgestuurd naar zijn homepagina waar een overzicht van zijn assets zijn.

## Ongewenst gebruik van de applicatie

Alles wat in de database wordt opgeslagen is van te voren gecontroleerd door het systeem. Op die manier komt er geen data in de database die er niet thuis hoort.

Het systeem controleert de gegevens eerst op een vriendelijke manier met javascript. Als de gebruiker die controle probeert te omzeilen worden de gegevens gecontroleerd door php. Die controle is niet omzeilbaar.

Met vriendelijke controle wordt bedoeld de manier van melden. Met de javascript controle kan de gebruiker zijn gegevens aanpassen (ze blijven ingevuld in de invulvelden). Bij de PHP controle zijn de gegevens al verstuurd naar de backend en dus verdwenen uit de invulvelden. Als er dan dus een foutmelding optreedt zal de gebruiker de gegevens opnieuw moeten invullen.

# 

# Beheer

## Back-up

Back-ups van code en documentatie worden geupload naar [github](https://github.com/Jurjen-V/Asset-tracking). Op die manier is er altijd een versie om terug te vallen als de applicatie helemaal corrupt geraakt is. Verder is het duidelijk gemaakt in de planning wanneer ik wat ga doen. Het risico dat ik niet door kan als er iemand niet is vervalt omdat ik als enige aan dit project werk.

Het back-uppen van de database leg ik bij de opdrachtgever neer. Als de database eens in de zoveel tijd een back up moet maken zal de opdrachtgever dat zelf moeten doen dat komt omdat de opdrachtgever zelf de hosting regelt. Na het maken van het project ben ik niet meer verantwoordelijk als de database data verloren gaat. Er zal natuurlijk wel een blanco versie van de database zijn waar altijd op teruggevallen kan worden.

## Content

Content op de website wordt beheerd door elke gebruiker apart. Een gebruiker kan als enige zijn content aanpassen. Een admin kan alleen de accountgegevens aanpassen van een gebruiker.

Verder wordt er content geleverd door een API van derden. Dat gebeurt door een API call te maken en de data op te slaan in de database.

De content die wordt geüpload door de gebruikers wordt van te voren wel gecontroleerd door het systeem. Het systeem controleert met javascript (front-end) en met PHP (back-end) op die manier kan de gebruiker de controle niet omzeilen.