

## PROJECT A – Twitter Zuil

Datum : 2020-11-02  
Versie : 1.0  
Auteur : Taha Genc  
1788230

Datum	Versie	Omschrijving
2020-11-02	1.0	Basis versie. Alles is ingevuld

## INHOUDSOPGAVE

1 Inleiding.....	3
2 Functionaliteit.....	3
1 Gedrag.....	4
1.1 BPMN.....	4
1.2 Use case diagram.....	5
1.3 Actoren.....	5
1.4 Use case templates.....	6
2 Datamodel.....	7
2.1 Conceptueel datamodel.....	7
2.2 Logisch datamodel.....	7
2.3 Fysiek datamodel.....	7
3 Technologieën.....	8
3.1 Python.....	8
3.2 Twitter.....	8
3.3 Database.....	8
4 Twitter koppeling.....	8
5 Gebruikers handleiding.....	9
1. Feedback invoeren.....	9
2. Voorwaarden bericht.....	9
1. Inloggen.....	10
2. Autoqueue.....	10
3. Moderators inzien/toevoegen/verwijderen.....	11
4. Locaties inzien/toevoegen.....	11
6 Referenties.....	12

# 1 Inleiding

De Nederlandse Spoorwegen vinden het erg belangrijk dat zij goed kunnen communiceren met hun klanten én van hun klanten goede of minder goede ervaringen horen. Het lijkt de directie een goed plan dat klanten hun opmerkingen/complimenten via een computer, aanwezig op elk het station, kunnen invoeren en dat deze opmerkingen dan zichtbaar worden in die stationshal.

## 2 Functionaliteit

Het systeem zal uit 3 modules bestaan met daarnaast een database om data op te slaan. Gebruikers/Reizigers zullen alleen interacteren met module 1, de zuil waarop berichten ingevoerd worden in de database. Moderators van de NS zullen gebruikmaken van module 2, de moderating module. Module 3 zal worden gestart op een scherm in een stationshal zonder gebruikers.

Hieronder worden modules in details beschreven.

Module 1 zuil:

Gebruikers kunnen gebruik maken van een computer in de stationshallen, op deze computer staat module 1. De enige taak van deze module is het bericht, naam en tijd op te slaan in de database. Voordat de data wordt opgeslagen in de database moet er controles worden toegepast op de invoer. In het geval van het bericht mag deze maximaal 140 characters lang zijn. De naam is niet verplicht. Standaard staat de naam als "anoniem" ingevuld, maar de gebruiker mag de naam aanpassen.

Module 2 moderating:

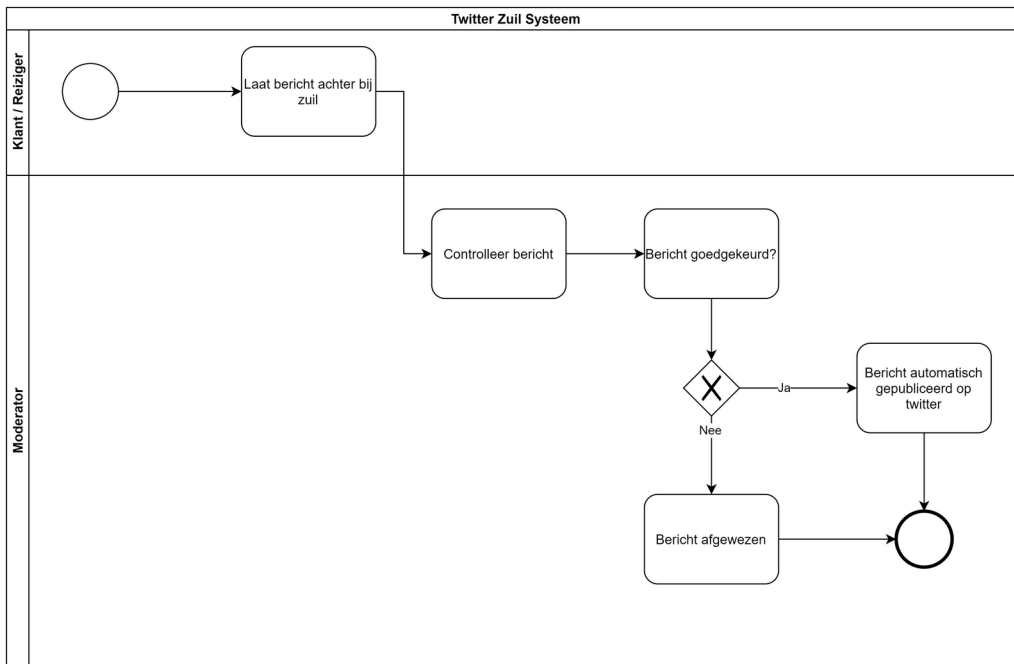
Moderators van de NS keuren de berichten die gebruikers invoeren. Deze module haalt automatisch de oudste ongekeurde bericht uit het systeem en weergeeft het voor de moderator. Wanneer de moderator een bericht goedgekeurd wordt de moderatie opgeslagen in de database. Waarna het automatisch ook op Twitter wordt geplaatst. Wanneer de moderator het bericht afkeurt wordt de moderatie in de database opgeslagen en niet op Twitter geplaatst. De module haalt zolang er ongekeurde berichten zijn berichten op en de moderators keuren het. Wanneer er geen berichten meer zijn om te keuren krijgen de moderators daarover een melding.

Module 3 Scherm:

Deze module staat aan op een groot scherm in een stationshal. Deze module weergeeft goedgekeurde berichten weer op de scherm.

# 1 Gedrag

## 1.1 BPMN



### Process beschrijving:

Het process begint bij de klant, hierna genoemd reiziger. De reiziger vult een bericht in bij een computer zuil.

Het bericht is maximaal 140 tekens lang, er wordt ook om de naam gevraagd. De naam mag leeg gelaten worden, waarna er automatisch "anoniem" wordt bijgevoeld. Nadat het bericht in de database is opgeslagen, is het de beurt aan de moderator om het bericht goed te keuren.

Als de moderator het bericht goedkeurt, wordt automatisch het bericht gepubliceerd op twitter, en wordt op een groot scherm in de stations hal getoond.

Als het bericht wordt afgekeurd wordt er niets meer met het bericht gedaan.

### Actoren:

klant/reiziger, moderator

### ICT infrastructuur:

computer zuil voor de klant  
computer voor de moderator  
groot beeldscherm in de stations hal  
een database

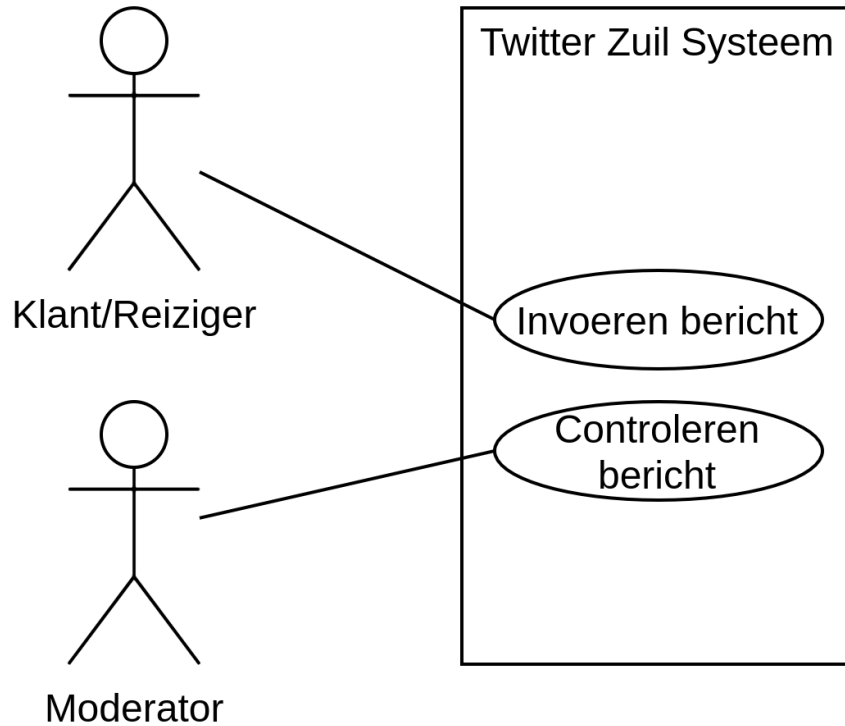
### Diagram structuur:

pool: Twitter Zuil Systeem  
lanes: klant/reiziger, moderator

### Diagram gateways:

- Exclusive gateway

## 1.2 Use case diagram



## 1.3 Actoren

Actor	Beschrijving
<b>Klant / Reiziger</b>	Is de actor die gebruikt maakt van NS diensten. In de stations hallen bij computer zuilen start de actor de usecase: invoeren bericht.
<b>Moderator</b>	Is de actor die berichten keurt. De moderator heeft eigen waarden waarmee de berichten goedgekeurd of afgekeurd worden. Zie: controleren bericht

## 1.4 Use case templates

**ID: 1**            **Naam:** Invoeren bericht

**Actoren:** Reiziger/Klant

**Samenvatting:** De actor voert een bericht in bij een computer zuil. Het bericht is maximaal 140 tekens lang. Er wordt ook om een naam gevraagd(i), dit is maximaal 32 tekens lang. Na het invullen, wordt het bericht opgeslagen in de database en wordt de actor bedankt om de feedback.

**Precondities:**

**Scenario/Stappenplan:**

1. De actor voert het bericht in(i)
2. Als het bericht langer dan 140 characters is.
  - 2.1. Een melding waarschuwt dat het bericht te lang is, en ingekort moet worden
3. De actor voert een naam in
4. Als de naam langer dan 32 characters is.
  - 4.1. Een melding waarschuwt dat de naam te lang is, en ingekort moet worden.
5. Als de naam korter dan 1 character is.
  - 5.1. De naam wordt automatisch ingevuld met de waarde "anoniem"
6. Het systeem probeert alle waarden op te slaan in de database
7. Als het bericht niet wordt opgeslagen in de database.
  - 7.1 Toon een bericht dat het bericht niet opgeslagen is en dat de actor het later opnieuw moet proberen.
  - 7.2 De usecase eindigt met postconditie 2
6. De actor wordt bedankt voor het bericht
7. De usecase eindigt met postconditie 1

**Postcondities:**

1. Een bericht, opgeslagen in de database
2. Geen bericht

**ID: 2**            **Naam:** Controleren bericht

**Actoren:** Moderator

**Samenvatting:** De actor start de automatische bericht controle wachtrij(i). Het systeem haalt de oudste bericht op uit de database. De actor controleert of het bericht goedgekeurd moet worden. De actor keurt het bericht goed of niet goed(i). Het systeem haalt automatisch weer de volgende oudste bericht op

**Precondities:**

**Scenario/Stappenplan:**

1. De actor start de automatische bericht controle wachtrij(i)
2. Het systeem haalt de oudste niet gekeurde bericht op.
3. Als er geen berichten meer zijn.
  - 3.1. Het systeem meldt dat er geen berichten meer zijn in de wachtrij
  - 3.2. De usecase eindigt met postconditie 1
4. De actor keurt het bericht(i)
5. Als het bericht is goedgekeurd.
  - 5.1 Het bericht wordt opgeslagen met de goedgekeurde status in de database
  - 5.2 Het bericht wordt gepubliceerd op twitter
  - 5.3 Het systeem gaat verder met stap 2
6. Als het bericht is afgekeurd
  - 6.1 Het bericht wordt opgeslagen met de afgekeurde status in de database
  - 6.2 Het systeem gaat verder met stap 2

**Postcondities:**

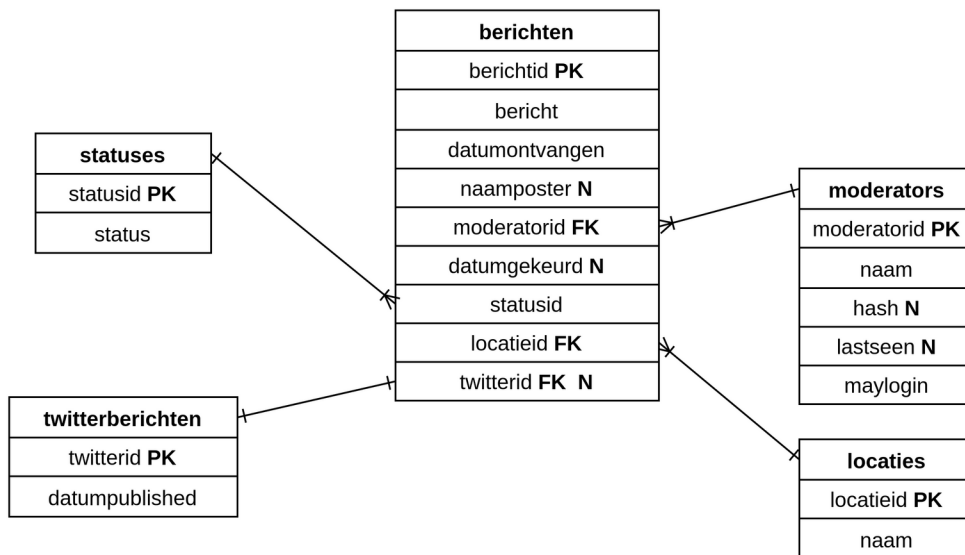
1. Eventueel gekeurde berichten

## 2 Datamodel

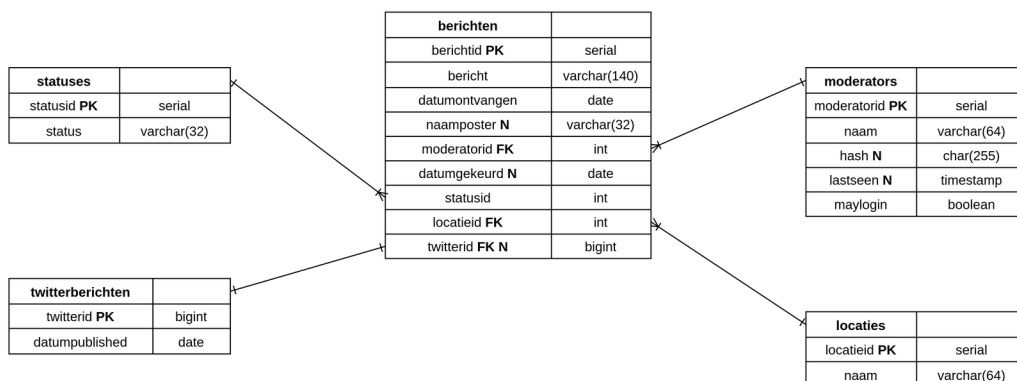
### 2.1 Conceptueel datamodel

Entiteit	Beschrijving
<b>berichten</b>	Hierin worden de berichten opgeslagen met bijbehorende data zoals naam, datum, status en locatie
<b>moderators</b>	Moderator namen en wachtwoord hashes worden in dit tabel opgeslagen
<b>locaties</b>	EIkk station krijgt een eigen id
<b>statuses</b>	Er is er voor gekozen om bericht statusen in een apart tabel op te slaan om het wat overzichtelijker te maken
<b>twitterberichten</b>	De twitter id en post time worden hierin opgeslagen

### 2.2 Logisch datamodel



### 2.3 Fysiek datamodel



## 3 Technologieën

### 3.1 Python

Python is de programmeer taal die gebruikt wordt in dit project. Python is een simpel maar krachtig programmeer taal en beschikt over een groot bibliotheek aan externe software, ookwel libraries genoemd. Voor de koppeling tussen python en de web browser maken wij gebruik van Flask. Dit is een simpele library waarmee we de “backend” van onze project gaan schrijven. Flask zorgt ook overigens voor de HTML templating.

### 3.2 Twitter

Twitter is de online berichten platform waar geaccepteerde berichten op worden gepubliceerd. Twitter maakt een API beschikbaar voor programmeurs om makkelijk data uit te wisselen tussen beiden systemen. De formaat waarmee de datauitwisseling plaatsvind is JSON. Twitter verseist stricte authenticatie protocollen. Hierom gebruiken we de Python library TwitterAPI gemaakt door Geduldig. Dit maakt de koppeling tussen Twitter en python makkelijker.

### 3.3 Database

Dit project maakt gebruik van de database software Postgres. Dit is een oud maar robuust database systeem waar we onze informatie, waaronder berichten, moderator informatie en twitter informatie opslaan. Voor de koppeling tussen Postgres en Python maken we gebruik van de psycopg2 software library in python waarmee we makkelijk data kunnen uitwisselen tussen de twee technologieën.

## 4 Twitter koppeling

In dit project maken wij gebruik van de quick messaging diens Twitter. Twitter bied hun eigen software interfaces aan via twitter.com en hun diverse mobiele applicaties. Om met software niveau te werken met Twitter wordt er een API aangeboden, momenteel versie 1.1. Om gebruik te kunnen maken van de Twitter API moet er eerst een aanvraag gedaan worden. Nadat deze is goedgekeurd krijgen wij 5 sleutels om met de API te kunnen communiceren. Deze zijn als volgt:

- TWITTER\_APIKEY
- TWITTER\_APISECRET  
Deze twee sleutels bieden ons toegang tot de API
- TWITTER\_BEARER  
Hiervan maken wij geen gebruik
- TWITTER\_ACCESSTOKEN
- TWITTER\_ACCESSECRET  
Deze twee sleutels bieden ons toegang om acties uit te voeren

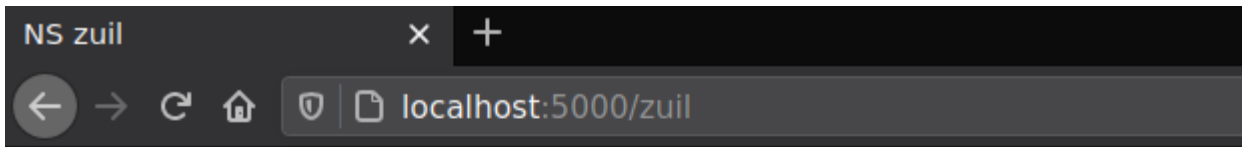
Met de bovenstaande vier sleutels kunnen wij verbinding maken en data wijzigen via de API. Deze applicatie maakt alleen gebruik van de *statuses/update* functie om berichten te plaatsen. En eventueel later van de *statuses/destroy/:id* functie om berichten te verwijderen via de moderator portal. De berichten die worden geplaatst zien er als volgt uit:

Lorem ipsum dolor sit amet | Door eiusmod op Utrecht Centraal



## 5 Gebruikers handleiding

# Gebruikershandleiding zuil



## Welkom op Amsterdam Centraal

Heeft u feedback voor ons?

Feedback wordt openbaar gemaakt op Twitter na goedkeuring

### 1. Feedback invoeren

Het achterlaten van een bericht op de NS zuil is extreem simpel. Vul uw bericht in die aan de onderstaande voorwaarden voldoen. Als u wilt kunt u uw naam achterlaten. De naam staat standaard op *anoniem*. Berichten die succesvol worden opgeslagen, worden na goedkeuring van een moderator gepubliceerd op Twitter en op de stations scherm. Berichten die niet worden goedgekeurd worden alsnog opgeslagen.

### 2. Voorwaarden bericht

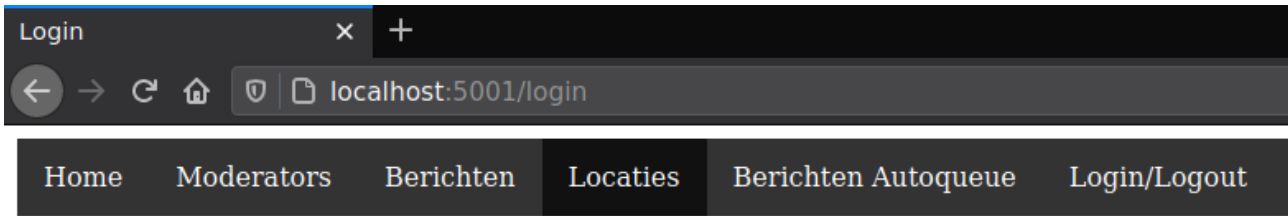
Het bericht moet aan enkele voorwaarden voldoen. Die worden hieronder beschreven.

- Het bericht is tussen 16 en 140 characters lang
- Uw naam is tussen de 3 en 32 characters lang

Als deze voorwaarden niet voldaan worden, wordt uw bericht niet opgeslagen.

# Moderator handleiding

## 1. Inloggen



## Please login to access home page

Start de web pagina, en login met de gebruikersnaam en wachtwoord die u van uw leidinggevende heeft gekregen. Bent u uw wachtwoord vergeten? Vraag de leidinggevende op verdere hulp. Na een succesvol inlog poging wordt u automatisch doorverwezen naar de Home pagina.

## 2. Autoqueue

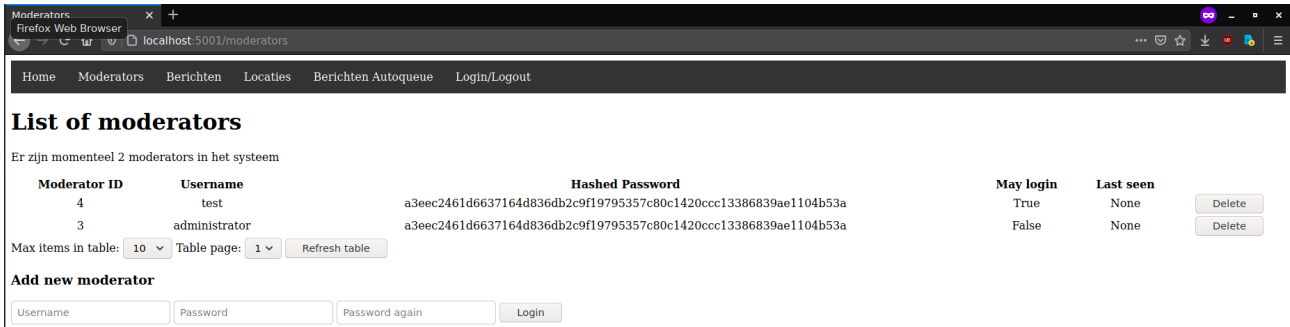


Om de autoqueue te starten klikt u vanaf de Home pagina de autoqueue knop aan op de bovenstaande navigatie balk. De meest oudste bericht die nog niet is gekeurd wordt opgehaald om te worden gemodereerd. Als de naam en bericht inhoud aan de gestelde voorwaarden voldoen mag er op de *goedkeuren en publiceren* knop aangeklikt worden. Dit bericht wordt dan op goedgekeurd gezet onder uw naam en wordt er geprobeerd om te publiceren op twitter. Als de naam **of** bericht inhoud **niet** aan gestelde voorwaarden voldoen dient er op de *afkeuren* knop geklikt te worden.

Wanneer een bericht is gekeurd wordt automatisch de volgende bericht opgehaald. Als er geen berichten meer zijn om goed te keuren wordt u doorverwezen naar de Home pagina met de melding dat er geen berichten meer zijn om goed te keuren. U kunt het later opnieuw proberen.

Wanneer de moderator wilt stoppen met berichten modereren moet er altijd op de knop *klik hier om te stoppen* gebruikt worden.

### 3. Moderators inzien/toevoegen/verwijderen

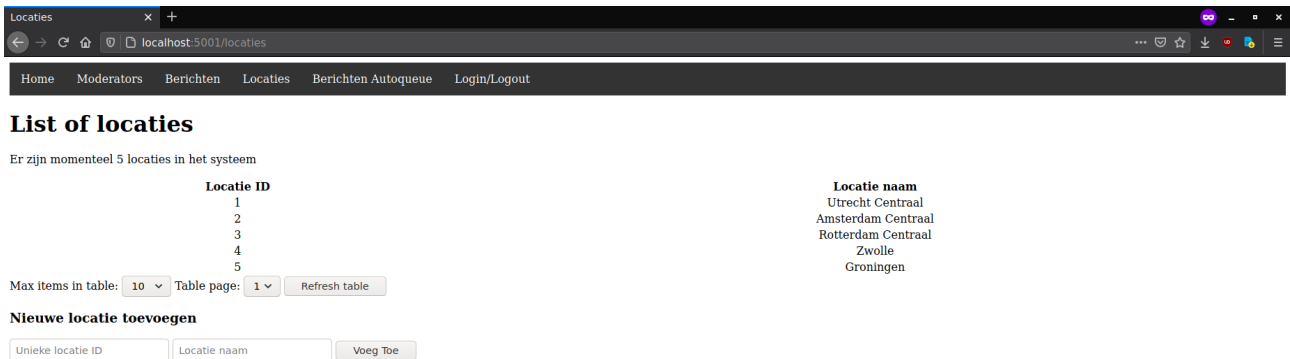


Met de menu optie moderators kunt u alle moderators inzien, toevoegen en verwijderen.

Om een moderator toe te voegen vult u onderaan de kopje “add new moderator” een gebruikersnaam en twee keer een wachtwoord in. Wanneer een moderator is toegevoegd mag die gelijk inloggen in de inlog scherm.

Om een moderator te verwijderen klikt u op de “delete” knop achter de moderator die verwijderd dient te worden. **LET OP** als een moderator berichten heeft goedgekeurd kan de account niet verwijderd worden. De moderator account wordt niet verwijderd maar mag na de actie niet meer inloggen.

### 4. Locaties inzien/toevoegen



Met de menu optie locaties kunt u alle locaties inzien en locaties toevoegen. Elk locatie heeft een eigen uniek ID. Om een locatie toe te voegen vult u aan de onderkant een unieke locatie ID in, dit kan het beste een 4 cijferig postcode zijn. Een locatie ID mag geen spaties of letters bevatten, alleen cijfers. Na de locatie ID vult u de stations naam in. Dit moet er goed uitzien met hoofdletters.

## 6 Referenties

Flask Quickstart Guide, <https://flask.palletsprojects.com/en/1.1.x/quickstart/>

OpenWeatherMap Current weather data, <https://openweathermap.org/current>

Psycopg documentation, <https://www.psycopg.org/docs/>

Twitter API documentation, <https://developer.twitter.com/en/docs/twitter-api>

TwitterAPI Geduldig documentation, <http://geduldig.github.io/TwitterAPI/>