FONTYS

ICT

Challenge 1 ICT & Software Slimme verlichting

ICT & Software

Challenge 1 van Mika Kleinbergen

Voor het vak ICT & Software semester 2

Fontys Hogescholen, ICT

18-02-2025

Docent Meulenbroeks, Marco

Document versiebeheer

Versie	Auteur	Datum	Wijziging
0.1	Mika	18-02-2025	Stap 1: analyse en stap 2
0.2	Mika	22-02-2025	Stap 3: database ontwerp
1.0	Mika	22-02-2025	Stap 4 & 5
1.1	Mika	24-02-2025	Feedback
1.2	MIka	8-03-2025	Iteraties
1.3	Mika	16-03-2025	User requirements

Inhoud

Stap 1: Analyse

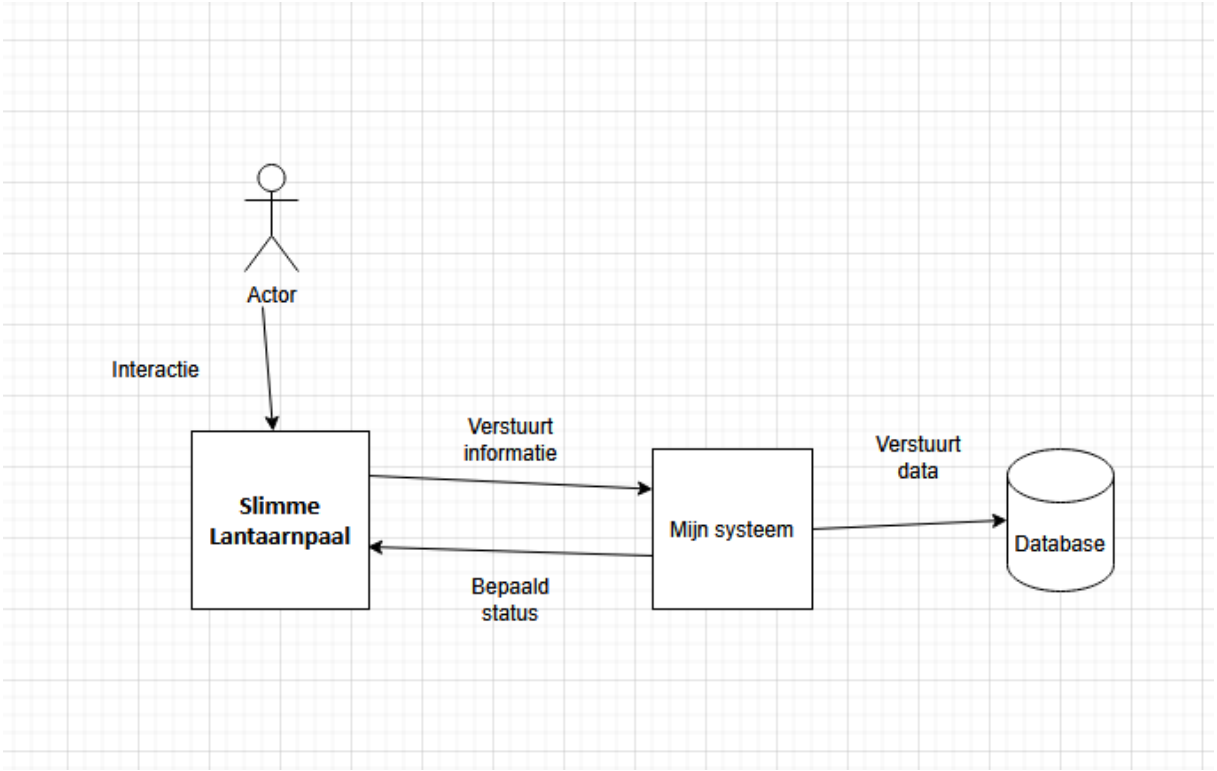
Basisfunctionaliteiten van het systeem:

Het systeem dat ik ga maken moet licht kunnen meten en beweging kunnen detecteren. Op basis van deze gegevens moet er een keuze gemaakt worden of het licht aan of uit gaat. Er moet data opgeslagen worden over de status van de lantaarnpaal. Er moet een eenvoudige interface zijn om informatie op te halen.

Benodigde gegevens:

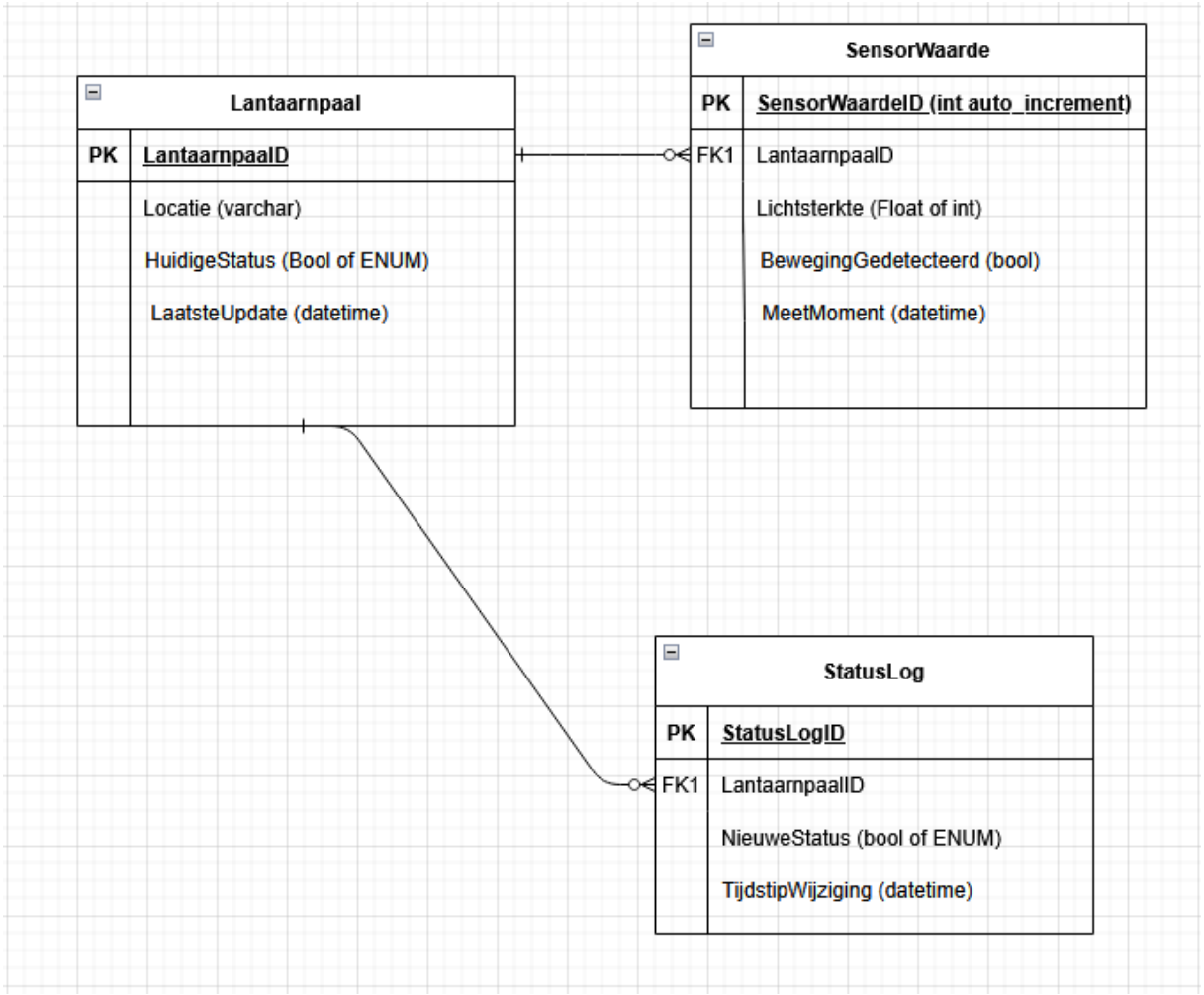
Ik heb thuis een Micro:bit gevonden. De Micro:bit heeft een led sensor die lichtsterkte kan meten. Ik heb een PIR sensor besteld om ook beweging te meten. Daarnaast moet er data opgeslagen worden over de status van de lamp.

Concept Diagram:



Stap 2: Conceptueel model ERD

Belangrijke gegevens zijn de lichtsterkte en de beweging. Deze gegevens moeten ook de tijd bevatten. En het moet duidelijk zijn om welke lantaarnpaal het gaat.



Stap 3: Databaseontwerp

```
CREATE DATABASE IF NOT EXISTS Lantaarnpaal;

USE Lantaarnpaal;

CREATE TABLE Lantaarnpaal (

        LantaarnpaalID INT AUTO_INCREMENT PRIMARY KEY,

        HuidigeStatus ENUM('Aan', 'Uit') NOT NULL,

        LaatsteUpdate DATETIME DEFAULT CURRENT_TIMESTAMP

);

CREATE TABLE SensorWaarde (

        SensorWaardeID INT AUTO_INCREMENT PRIMARY KEY,

        LantaarnpaalID INT,

        Lichtsterkte INT NOT NULL,

        BewegingGedetecteerd BOOLEAN NOT NULL,

        MeetMoment DATETIME DEFAULT CURRENT_TIMESTAMP,

        FOREIGN KEY (LantaarnpaalID) REFERENCES Lantaarnpaal(LantaarnpaalID) ON DELETE
CASCADE

);

CREATE TABLE LogStatus (

        LogStatusID INT AUTO_INCREMENT PRIMARY KEY,

        LantaarnpaalID INT,

        NieuweStatus ENUM('Aan', 'Uit') NOT NULL,

        TijdstipWijziging DATETIME DEFAULT CURRENT_TIMESTAMP,

        FOREIGN KEY (LantaarnpaalID) REFERENCES Lantaarnpaal(LantaarnpaalID) ON DELETE
CASCADE

);
```

Door ChatGPT aangepast voor Microsoft SQL server:

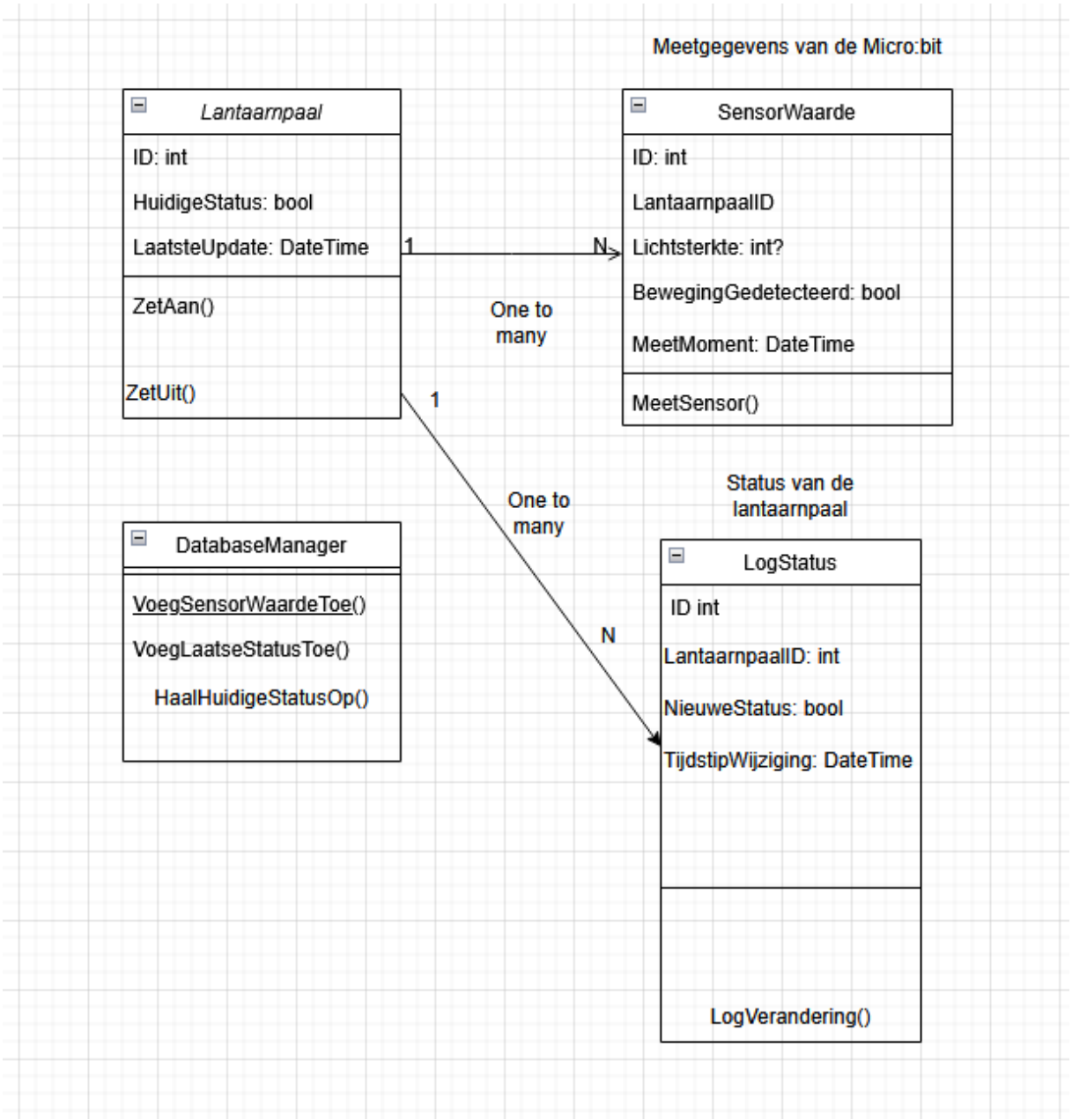
```
-- Maak de database als deze nog niet bestaat
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'Lantaarnpaal')
BEGIN
	CREATE DATABASE Lantaarnpaal;
END
GO
USE Lantaarnpaal;
GO

-- Maak de Lantaarnpaal tabel
CREATE TABLE Lantaarnpaal (
	LantaarnpaalID INT IDENTITY(1,1) PRIMARY KEY,
	HuidigeStatus BIT NOT NULL DEFAULT 0, -- 1 = AAN, 0 = UIT
	LaatsteUpdate DATETIME DEFAULT GETDATE()
);
GO

-- Maak de SensorWaarde tabel
CREATE TABLE SensorWaarde (
	SensorWaardeID INT IDENTITY(1,1) PRIMARY KEY,
	LantaarnpaalID INT,
	Lichtsterkte INT NOT NULL,
	BewegingGedetecteerd BIT NOT NULL DEFAULT 0,
	MeetMoment DATETIME DEFAULT GETDATE(),
	FOREIGN KEY (LantaarnpaalID) REFERENCES Lantaarnpaal(LantaarnpaalID) ON DELETE CASCADE
);
GO

-- Maak de LogStatus tabel
CREATE TABLE LogStatus (
	LogStatusID INT IDENTITY(1,1) PRIMARY KEY,
	LantaarnpaalID INT,
	NieuweStatus BIT NOT NULL DEFAULT 0, -- 1 = AAN, 0 = UIT
	TijdstipWijziging DATETIME DEFAULT GETDATE(),
	FOREIGN KEY (LantaarnpaalID) REFERENCES Lantaarnpaal(LantaarnpaalID) ON DELETE CASCADE
);
GO
```

Stap 4: Klassendiagram



Stap 5: Prototype

Code om lichtniveau van Microbit te lezen.

```
from microbit import *
import utime

while True:
    lichtniveau = display.read_light_level()  # Meet lichtsterkte (0-255)
    print(lichtniveau)  # Stuur waarde naar de seriële poort
    utime.sleep(1)  # Wacht 1 seconde
```

Stap 6: Iteraties

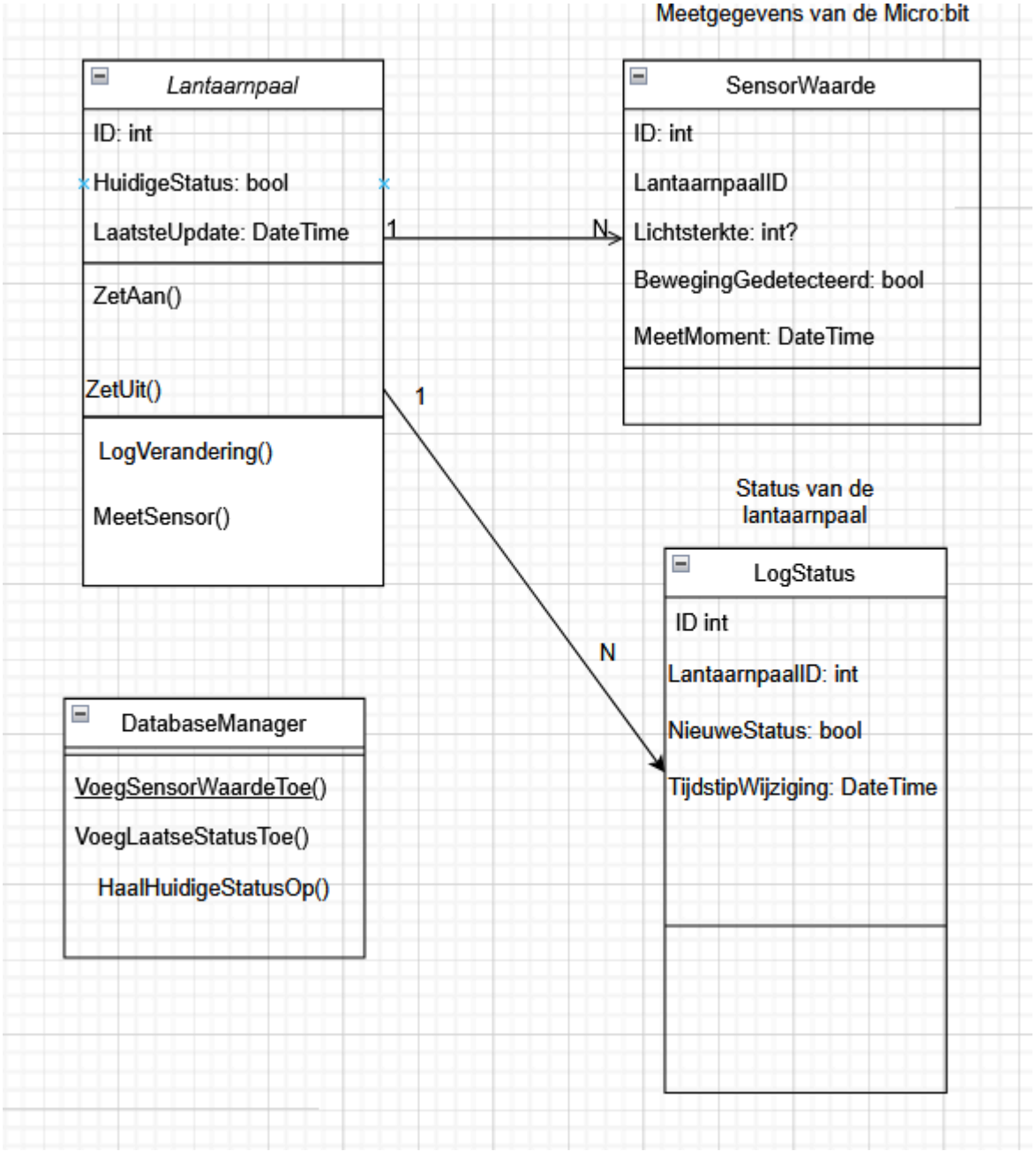
Feedback gekregen 24-02-2025:

- Code: Relaties in klassen moeten terugkomen.
- Klassendiagram: Functies horen bij klassen.
- Database model: One to many moet weg.
- User requirements: Requirements prioriteiten.
- Database: huidige status en tijdstip wijziging zijn overbodig.
- Concept diagram: Actor bij het systeem.

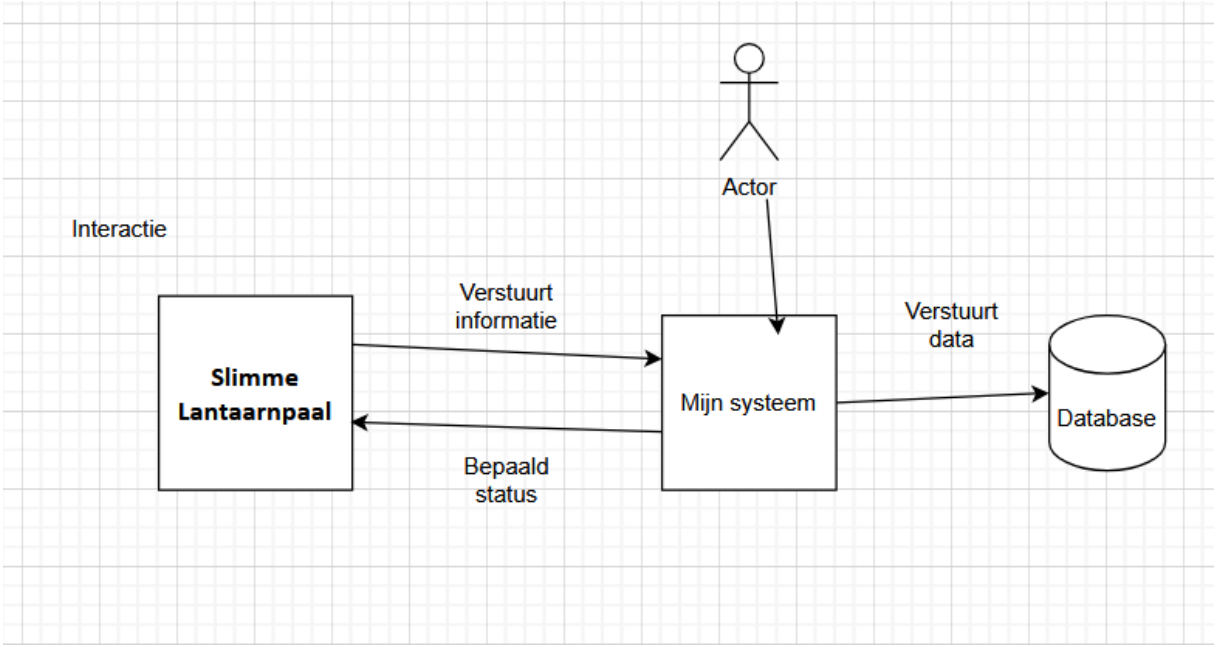
User requirements (MoSCoW)

1. Must-Have
- Het systeem moet lichtsterkte meten en beweging detecteren.
- Op basis van deze gegevens moet het systeem automatisch beslissen of de lantaarnpaal aan of uit gaat.
- De status van de lantaarnpaal (aan/uit) moet worden opgeslagen in een database.
- Er moet een eenvoudige interface zijn om de status van de lantaarnpaal op te vragen.
2. Should-Have
- De interface moet real-time updates tonen.
- Gebruikers moeten historische data kunnen inzien (wanneer het licht aan/uit ging).
3. Could-Have
- Gebruikers moeten de licht- en bewegingsdrempels kunnen aanpassen.
- Visuele weergave van de lantaarnpaalstatus in de interface (bijv. grafiek of kleuren).
4. Won’t-Have
- Ondersteuning voor meerdere lantaarnpalen.
- Mobiele app voor monitoring.

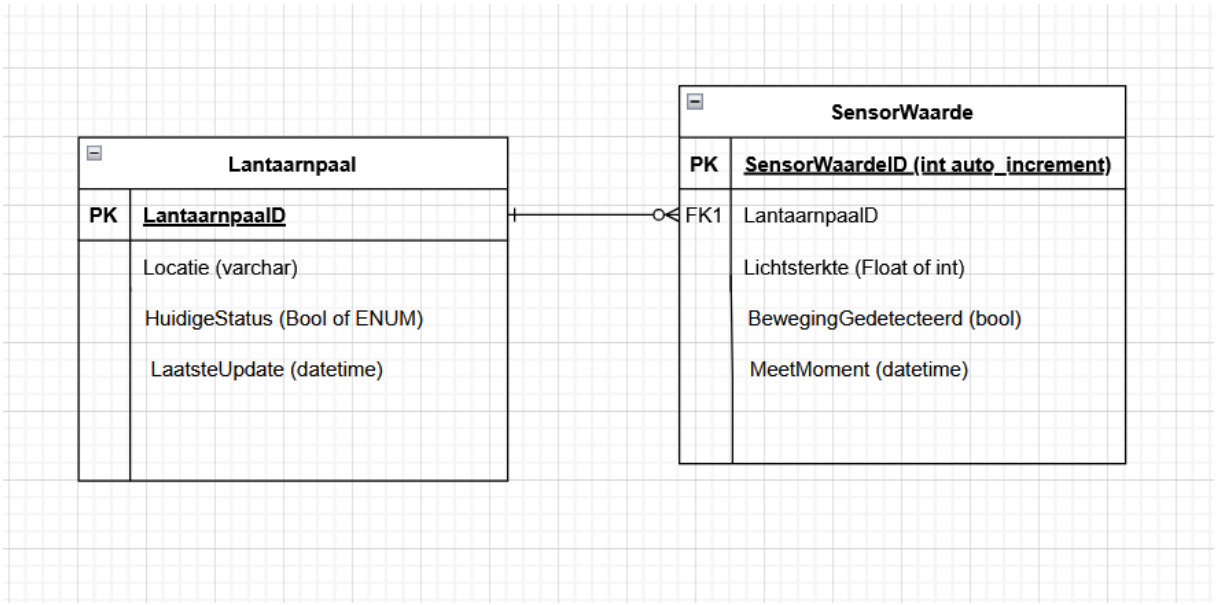
Klassendiagram: functies horen bij klassen.



Concept diagram: Actor bij het systeem.



Database: huidige status en tijdstip wijziging zijn overbodig.



Bibliografie

System context diagram. (z.d.). C4 Model. https://c4model.com/diagrams/system-context