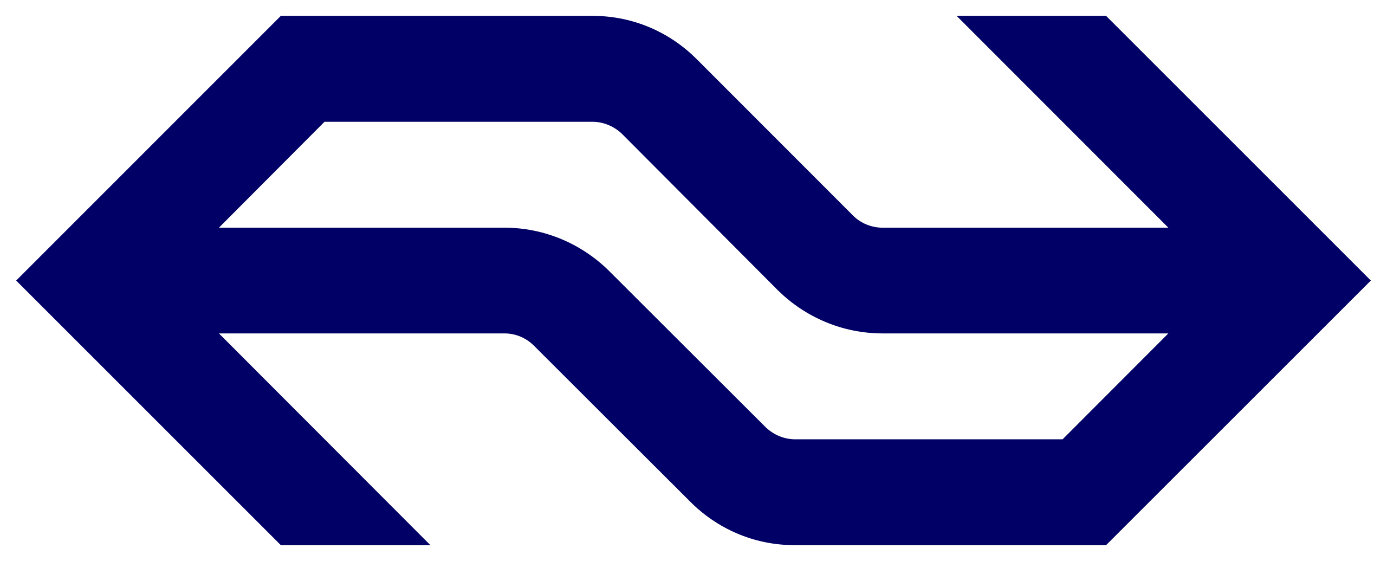
Project A - Stationszuil



Gemaakt en geschreven door: Sueño (Swen) Keijer

HBO-ICT HU

2023

Inhoud

[Inleiding 2](#_Toc125323622)

[Doelstelling 2](#_Toc125323623)

[Ontwerp 4](#_Toc125323624)

[Resultaat 6](#_Toc125323625)

[Referenties 7](#_Toc125323626)

# Inleiding

Dit project was een opdracht van de Hogeschool Utrecht. Het zorgde ervoor dat alle 3 de vakken die tijdens het eerste blok gegeven werden uiteindelijk gecombineerd werden in 1 project waar de student meer werkte en leerde.

Het algemene idee was om 3 programma’s te schrijven, ieder met een eigen doel.

De eerste schreef reviews (van een random station),

de tweede controleerde de reviews,

en de derde liet vervolgens een scherm zien met de reviews van het gekozen station.

Het ontwerpen hiervan heb ik vrij simpel gehouden. Zolang het werkt, werkt het, en ik wordt hier toch niet voor betaald. Zolang ik trots was met mijn eigen resultaat, was het goed genoeg. Ik wist van mezelf al dat ik anders te veel tijd zou gaan spenderen aan styling, en minder aan bugs fixen en nuttige code schrijven.

Er zit in deze map ook een `station\_service.txt`. Hierin staat een grote SQL code die in de server gerund moet worden. Ik had helaas geen manier gevonden om dat in mijn code makkelijk te zetten.

# Doelstelling

Het doel was simpel: maak elk programma werkend, en duidelijk voor de gebruiker. Maak het niet al te moeilijk, en gebruik de console indien mogelijk

1. review.py

De review programma zou om een input vragen van de gebruiker, een bericht niet langer dan 140 tekens. Klinkt kort, maar dat was de verplichting van het project.

Eenmaal als dat bericht gevraagd is, wordt er gevraagd of de gebruiker zijn/haar/hun naam wilt geven. Indien dat niet wordt gedaan (en het veld leeg blijft), wordt er “Anoniem” opgeslagen.

Het bericht en de naam worden vervolgens in een .csv bestand opgeslagen, met de datum en een willekeurig station erbij. Het station wordt gekozen uit een tekst bestand genaamd `stations.txt`.

1. mod.py

Het moderator programma is gemaakt om een moderator de reviews te laten lezen en controleren.

Eerst wordt om het emailadres gevraagd van de moderator. Indien die niet in de database gevonden is, wordt er een nieuw account voor diegene gemaakt. Hierbij wordt ook een naam aangevraagd.

Eenmaal als de moderator is ingelogd, worden de nieuwe reviews geüpload naar de database (indien die er zijn). Vervolgens komen uit de database alle reviews die nog niet gekeurd zijn.

Één voor één wordt de review geschreven in de console, en kan hij door middel van “1” of “0” in de console voeren, de review goedkeuren of afkeuren. Deze status wordt in een aparte tabel in de database toegevoegd.

Eenmaal als alle reviews gekeurd zijn, sluit het programma zichzelf af.

1. scherm.py

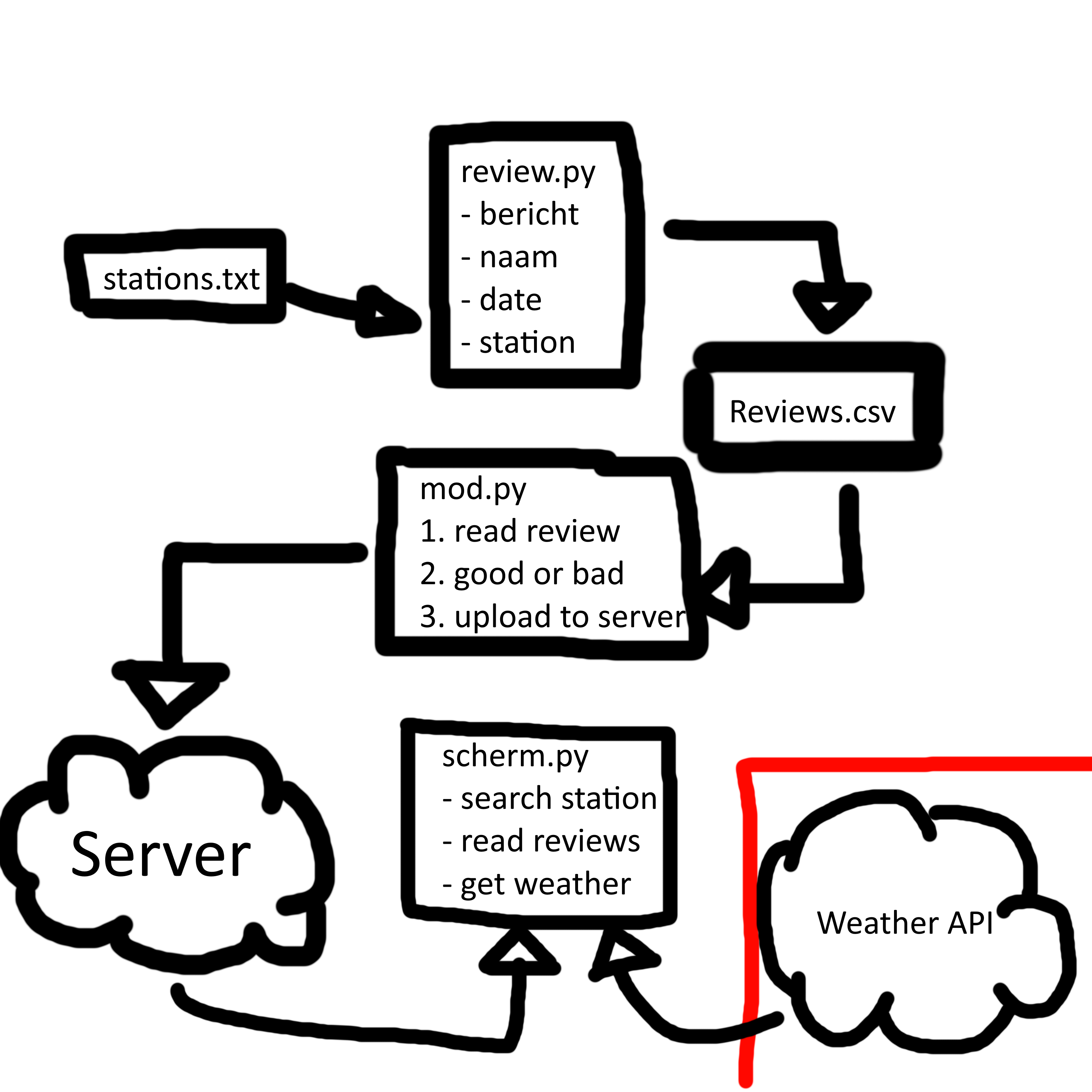
Dit programma bestaat uit 2 schermen:

* het keuze scherm
* Het station scherm

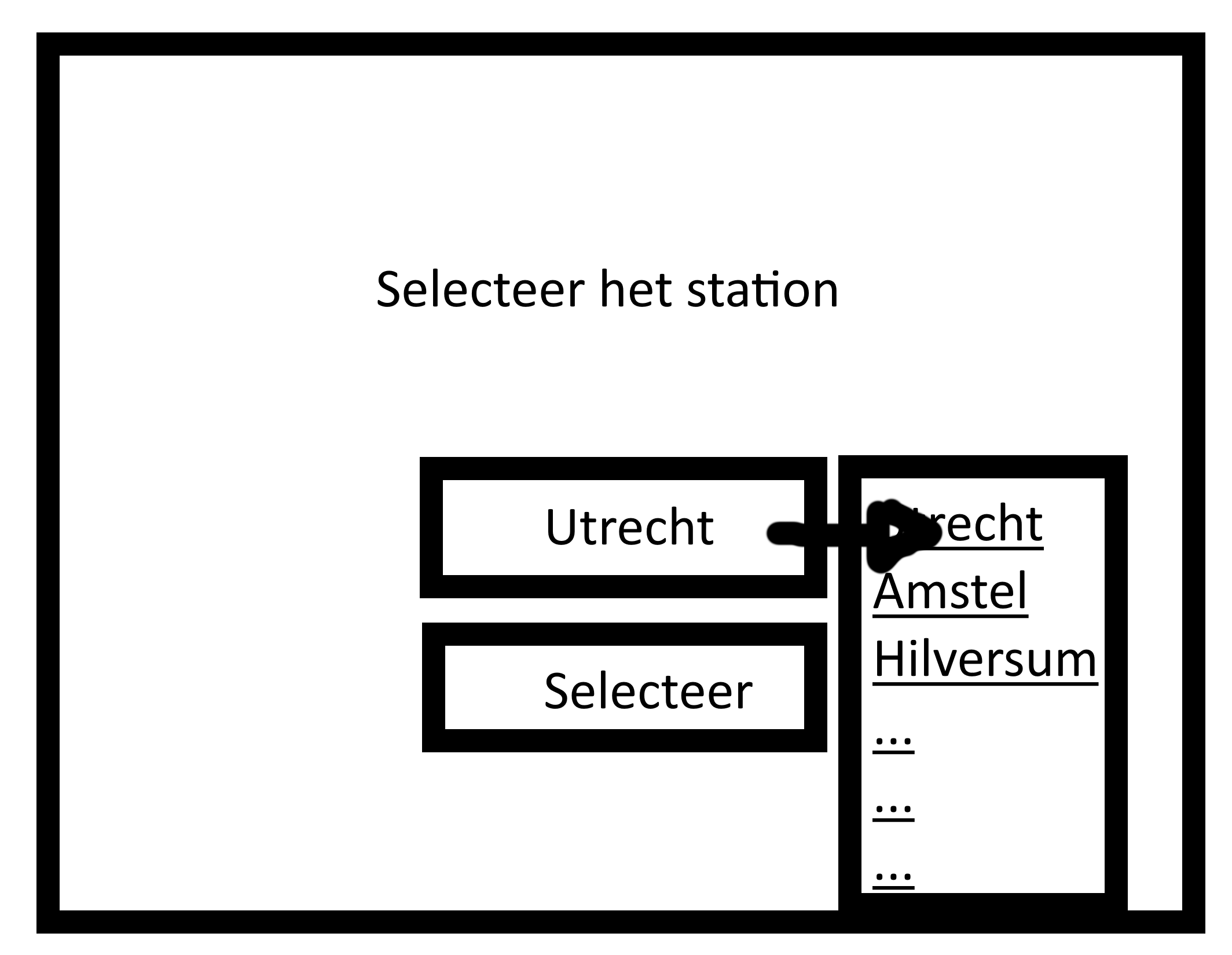
Het keuze scherm laat een knop zien met een al geselecteerde station. Eenmaal als je erop klikt, komt de volledige lijst met stations tevoorschijn waaruit je kan kiezen. Als je de juiste hebt gekozen, klik dan op de knop “selecteer”, en het volgende scherm komt tevoorschijn

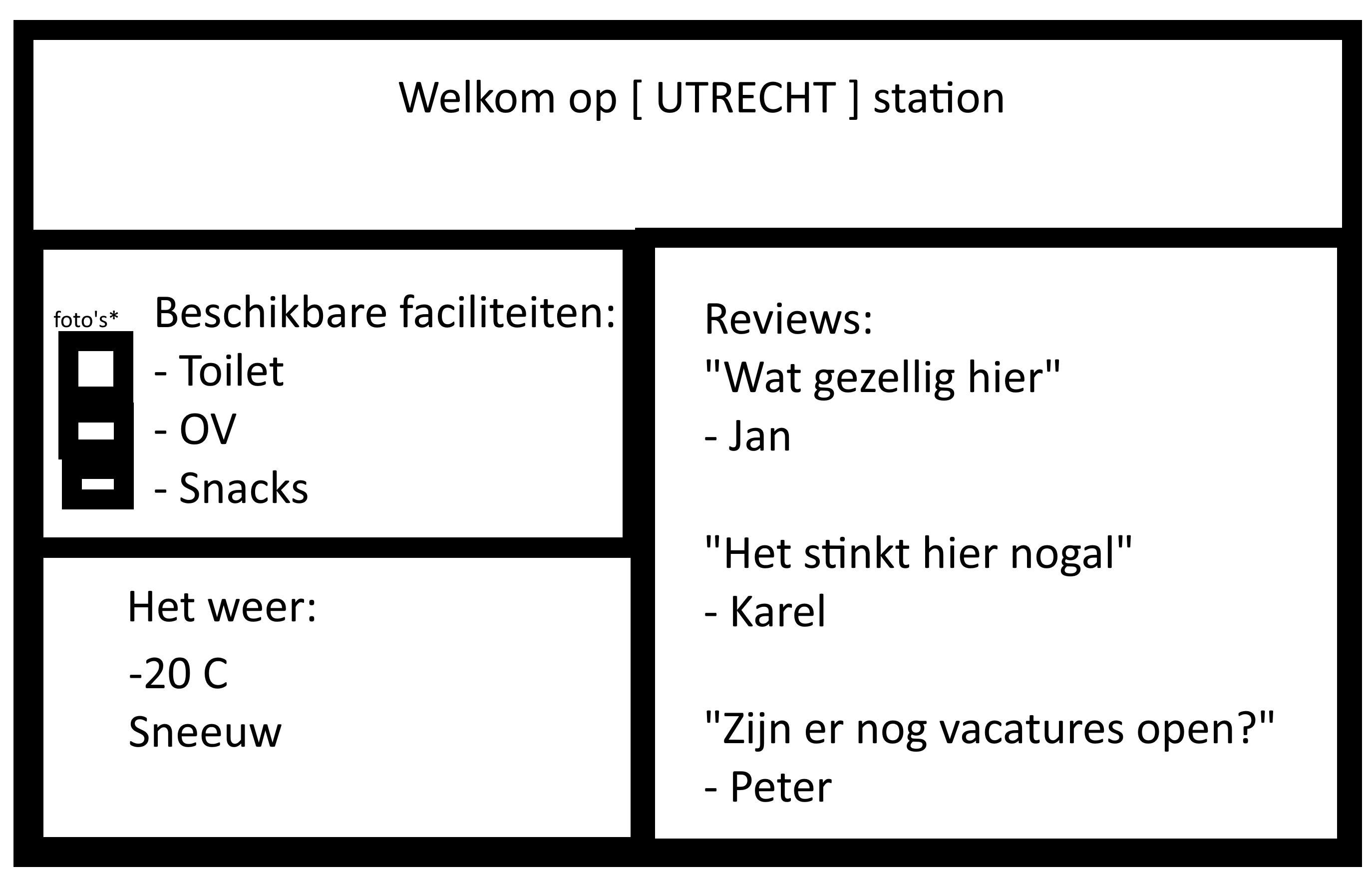
Hierin staat de station naam, beschikbare faciliteiten, de laatste 5 reviews, het weerbericht, en een klok onderin.

# Ontwerp

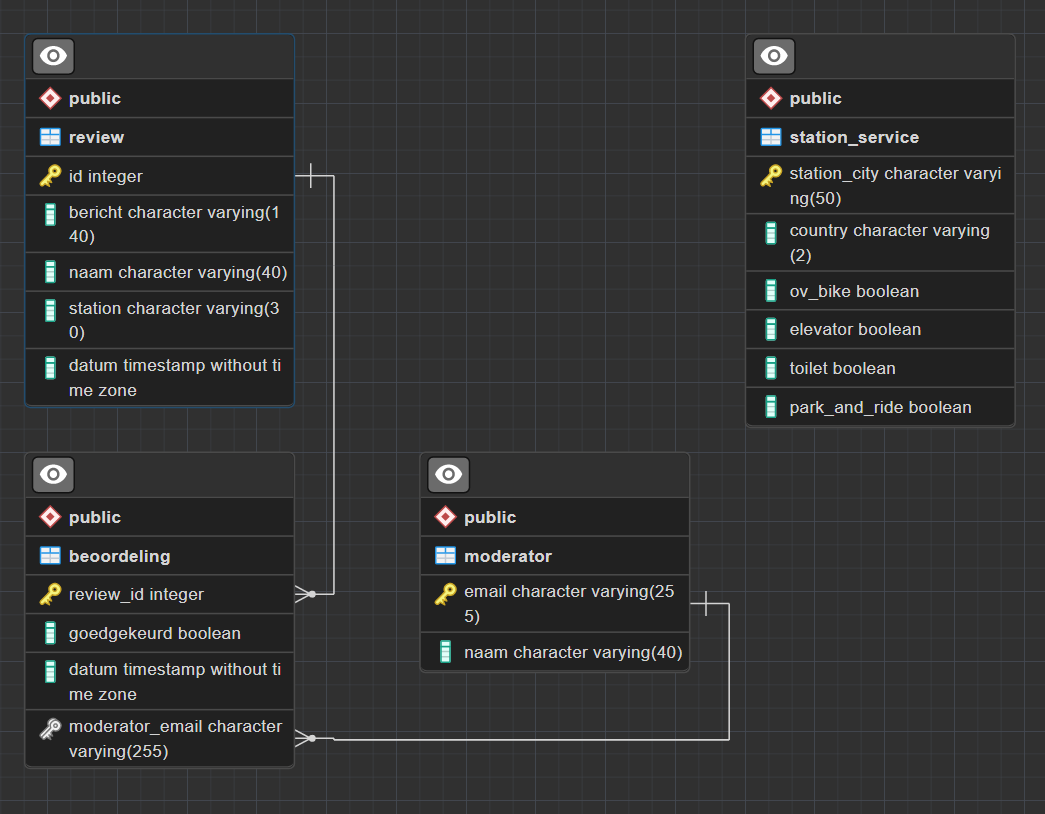


Basis idee van flow



****

Inlog scherm (boven), station scherm (onder)



Database ERD

# Resultaat

Alles werkte! Met een paar problemen op te lossen over de API key van de weersite

Ik had wel iets meer tijd moeten vrij plannen voor betere documentatie en onderzoek. Ik zat veel te lang met ander schoolprojecten in mijn hoofd, en ben daardoor bijna de herkansing hiervan vergeten. Hierdoor veel dingen last minute in elkaar gezet, maar wel voldoende genoeg om het werkend te krijgen.

Ik ben er niet bepaald trots op, maar het was een leuk project om te maken.

# Referenties

Een aantal websites die ik nodig had, of waarmee ik zat te strugglen. Lees ze goed door, je hebt het hard nodig.

Weather API:

<https://openweathermap.org>

How to use fetchall():

<https://pynative.com/python-cursor-fetchall-fetchmany-fetchone-to-read-rows-from-table/>

CSV lib docs:

<https://docs.python.org/3/library/csv.html>

howto pgAdmin:

<http://127.0.0.1:50674/help/help/connecting.html>

Psycopg2 docs:

<https://pypi.org/project/psycopg2/>

<https://www.psycopg.org/docs/>