Ontwerpkeuzes

Inhoudsopgave

[Inleiding 1](#_Toc156224570)

[Points of interest 2](#_Toc156224571)

[Kleurenpallet 3](#_Toc156224572)

[Metingen 4](#_Toc156224573)

[Wijkkaart 5](#_Toc156224574)

[Heatmap 7](#_Toc156224575)

[Tijdsbestek metingen 8](#_Toc156224576)

[Omgang met foutieve metingen 9](#_Toc156224577)

[Correctie van locatiegegevens 9](#_Toc156224578)

[Filteren van onrealistische data 9](#_Toc156224579)

[Timelapse 11](#_Toc156224580)

# Inleiding

Tijdens de ontwikkeling van Ontdekstation013 zijn er diverse ontwerpkeuzes gemaakt. Hier wordt voor elke van deze keuzes beschreven hoe zij tot stand zijn gekomen.

# Points of interest

De temperatuurkaart laat goed de contouren van de stad zien, maar een Tilburger zal zich misschien toch moeilijk kunnen vinden in de kaart. Om de gebruikers een beter richtingsgevoel te geven, is overwogen bepaalde Points Of Interest (POIs) expliciet aangegeven op de kaart.

Er is overlegd met de stakeholders en zelf onderzoek gedaan naar welke features dit kunnen zijn, hier is de volgende lijst uit voortgekomen:

**Meest herkenbaar**

* Spoorlijn
* Willemina Kanaal
* Bekenden gebouwen (zoals westpoint, rabobank, willem II stadion)
* Ziekenhuizen
* Stations
* Universiteiten en MBO’s

**Herkenbaar**

* Winkelcentra
* Bekende rotondes
* Kerken en kloosters
* Musea

**Enigszins herkenbaar**

* Bouwmarkten
* Meren
* Formaat parken (Wilhelmina Park, tivoli park, park oude dijk)

**README**

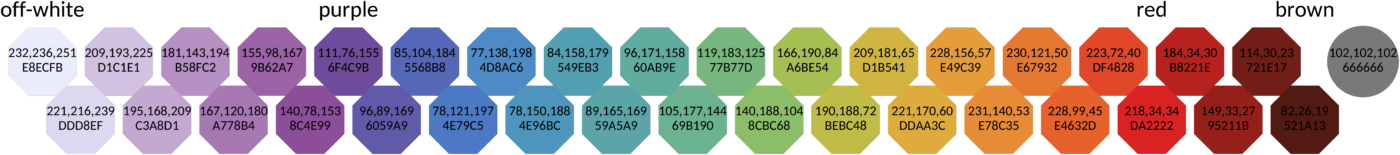
Na een niet gerelateerde update van de heatmap is gebleken dat de stadscontouren weer goed zichtbaar waren. Dit maakt het toevoegen van POIs op de kaart overbodig. Deze optie wordt voor nu geschrapt.

# Kleurenpallet

De heatmaps werken met kleuren om de temperatuurverschillen aan te geven. Echter werkt dit alleen als de gebruiker goed onderscheid kan maken tussen de kleuren, wat een probleem is bij kleurenblindheid.

Vanuit de opdrachtgever is de wens om rekening te houden met deze groep mensen door het gebruik van een speciaal kleurenpallet. Het voorstel om een speciale modus te introduceren voor kleurenblinden is afgewezen, omdat dit kan leiden tot een overvloed aan weergave-opties. Het doel is dan ook om 1 weergave te maken die voor bijna iedereen redelijk te lezen is.

In de besluitvorming is sterk gekeken naar het rapport van Paul Tol (Tol, 2021), hier beschrijft hij uitgebreid welke kleurenblind-vriendelijke pallets goed toepasbaar zijn voor verschillende situaties. Voor onze toepassing kijken we naar sequentiële data. Voor Ontdekstation 013 zijn meerdere custom kleurenpallets samengesteld die gebaseerd zijn op een gedeelte van het [Smooth rainbow](https://personal.sron.nl/~pault/#fig:scheme_rainbow_smooth) pallet van Poll, zie onderstaande figuren. Uiteindelijk is gekozen voor Figuur 4, de pallet met een zachte focus op rood.



Figuur 1: Smooth rainbow pallet van Toll, afkomstig van <https://personal.sron.nl/~pault/>

#5568B8 #4D8AC6 #69B190 #D1B541 #DDAA3C #E78C35 #E4632D #DF4828 #DA2222 #B8221E

Figuur 2: Custom kleurenpallet, met een harde focus op rood. Hier is er te weinig verschil in de roodtinten

#5568B8 #4D8AC6 #549EB3 #60AB9E #77B77D #A6BE54 #D1B541 #E49C39 #E67932 #DF4828 #B8221E

Figuur 3: Custom kleurenpallet, met een gelijke distributie tussen blauw en rood. Hier is blauwgroen te dominant.

#5568B8 #4E96BC #60AB9E #77B77D #A6BE54 #D1B541 #E49C39 #E67932 #DF4828 #B8221E

Figuur 4: Custom kleurenpallet, met zachte focus op rood. De eindkeuze.

# Metingen

Data

De beschikbare gegevens zijn temperatuur- en luchtvochtigheid metingen, afkomstig van meetstations, die opgehaald kunnen worden via de [openbare API](https://meetjestad.net/data/) van MeetJeStad. De metingen worden deels aangepast, zie Omgang met foutieve metingen, en bevatten onder andere de volgende relevante gegevens:

* MeetstationId
* Lengtegraad
* Breedtegraad
* Timestamp
* Temperatuur
* Luchtvochtigheid

In de toekomst kan de meetdata uitgebreid worden met fijnstofgehalte of andere data. De software dient geschreven te worden om rekening te houden met dit soort uitbreiding.

Scope

Ontdekstation wordt gesubsidieerd vanuit de gemeente Tilburg, daarom worden metingen binnen de gemeente Tilburg in beschouwing genomen. Metingen buiten de gemeente moeten weg gefilterd worden, dit wordt momenteel gedaan met een [geofencing algoritme](https://community.appinventor.mit.edu/t/geofence-check-if-a-point-is-inside-a-polygon-javascript-map/57091). Als andere gemeente laten besluiten mee te doen met het project, moet het mogelijk zijn deze toe te voegen in de scope.

Zolang alleen gemeente Tilburg deelneemt aan het project, dienen alle heatmaps en kaarten gefocust te zijn op Tilburg.

# Wijkkaart

Vanuit de opdrachtgever is de wens om een heatmap te realiseren die werkt op wijkniveau. Deze heatmap moet een overzicht vormen van de temperatuurverschillen tussen wijken. Hier worden wijken relatief aan elkaar vergeleken.

Scope

Er worden alleen wijken van de gemeente Tilburg weergeven op de wijkkaart. Wijken in andere gemeenten worden niet gerealiseerd. Mocht een andere gemeente later besluiten mee te doen met het project kan besloten worden hun wijken alsnog toe te voegen, de software dient hier rekening mee houden.

Wijken

Selectie

Afhankelijk van de bron, wisselt het aantal wijken in Tilburg. In dit project hebben wij als basis gekozen voor de lijst zoals deze bestaat op [Wikipedia](https://nl.wikipedia.org/wiki/Wijken_en_buurten_in_Tilburg) deze is weer gebaseerd op de lijst van het CBS. Vanwege moeilijkheden in het onderscheiden van wijk-dimensies zijn een aantal wijken op lijst samengevoegd, deze zijn:

* Verspreide huizen Udenhout is geïntegreerd met **Udenhout**
* Verspreide huizen ten westen van Berkel-Enschot is geïntegreerd met **Enschot**
* Verspreide huizen ten noorden van Berkel-Enschot is geïntegreerd met **Berkel**

*Als de selectie van wijken veranderd moeten eventuele berekeningen die gemaakt zijn met de wijken opnieuw gedaan worden.*

Naamgeving

Een aantal van de wijken zoals deze staat op Wikipedia heeft namen die onbekend zijn voor de inwoners hiervan. Hierom zijn een aantal wijknamen aangepast naar hun beter bekende naam, deze wijken zijn:

* Hoevenseweg is veranderd naar **Fatima**
* Hagelkruis is veranderd naar **Noordhoek**
* Gasthuisstraat is veranderd naar **Bouwmeester**
* Mariaziekenhuis-Vredeburcht is veranderd naar **Vredeburcht**

Dimensies

Om de wijken van gemeente Tilburg te weergeven op een heatmap, moeten de afmetingen hiervan bepaald worden. Er is geprobeerd deze afmetingen te fetchen van verschillende GIS-overheidssystemen, maar dit is niet gelukt. Daarom is het besluit genomen om de hoogte en breedtegraad van elke coördinaat van elke wijk handmatig te copy-pasten van Google Maps naar een database. Dit proces is niet perfect omdat:

* Er lichte overlap kan zijn tussen de wijken in Google Maps
* Niet elke wijk wordt aangegeven op Google Maps
* Het handmatig invoeren kleine afwijkingen kan hebben
* Google Maps niet per se de ‘officiële’ wijk-dimensies gebruikt

*Mochten ooit de wijken van Tilburg veranderen, zal dit handmatig aangepast moeten worden, en moeten eventuele berekeningen die gemaakt zijn met de wijk-dimensies opnieuw gedaan worden.*

Sorting algoritme

Om te bepalen welke metingen bij welke wijk horen wordt er een [geofencing algoritme](https://community.appinventor.mit.edu/t/geofence-check-if-a-point-is-inside-a-polygon-javascript-map/57091) toegepast. Dat simpel weg bepaalt of een bepaald punt binnen de vorm van de wijk staat.

Geschiedenis grafiek

In de grafiek die de geschiedenis van temperaturen in een wijk weergeeft, is gekozen om voor elke dag één datapunt te gebruiken. Dit datapunt bevat een minimum, maximum en gemiddelde temperatuur van alle metingen van die dag in die wijk.

# Heatmap

Vanuit de opdrachtgever is de wens om een heatmap te realiseren, deze moet een overzicht vormen van de temperatuurverschillen in de gemeente Tilburg.

Soorten

Er bestaan meerdere soorten heatmaps, deze zijn weighted, proximity en cell based. MB Ontdekt heeft alleen baat bij een weighted heatmap, waar een punt op de map aan de hand van een gewogen waarde zwaarder meetelt. Met deze soort is het mogelijk om de meetpunten vorm te geven afhankelijk van hun toegewezen waarde, oftewel een meetpunt met een temperatuur van 20 graden kan een ander visueel effect hebben dan meetpunt met een temperatuur van 15 graden.

Plugins

Er is uitvoerig gekeken naar wat de mogelijkheden zijn voor het weergeven van de meetstation data als weighted heatmap. Hier zijn verschillende plugins voor bekeken, waarna de conclusie is getrokken dat de meeste beschikbare plugins niet toereikend zijn, of niet binnen de scope van het project vallen.

De vorige plugin die werd gebruikt was ‘react-leaflet-heatmap-layer’, welke een React component aanlevert die met een config object ingesteld kan worden. De heatmap die vervolgens gegenereerd werd was een proximity heatmap, met geen enkele optie om hiervan een weighted heatmap te maken.

Een enkele plugin daarentegen bood wel mogelijkheid voor het weergeven van de meetstation data als weighted heatmap, namelijk: ‘*heatmap.js*’. De documentatie over de werking en het gebruik van de plugin is grotendeels te vinden op <https://www.patrick-wied.at/static/heatmapjs/>.

Echter is dit niet compleet voor hoe het in het project wordt toegepast. Zo is de plain JavaScript code verwerkt in het React component ‘HeatmapLayer.js’ en gecombineerd met de plugin ‘react-leaflet’ in plaats van het reguliere ‘leaflet’. Hoe er met React Leaflet gewerkt wordt is terug te vinden op <https://react-leaflet.js.org/docs/api-map/>.

Vervolg

Zoals eerder benoemd is de huidige implementatie van de heatmap niet toereikend voor de wensen van de stakeholders. Dit komt door de beperkingen van de gebruikte plugins en daarbij komt de noodzaak voor het coderen van een custom interpolatie systeem. Dit wordt uitgebreid behandeld in het document ‘Opdracht Heatmap interpolatie’.

# Tijdsbestek metingen

De meetstations van MeetJeStad sturen elke ± 16 minuten een meting door. Vanwege atmosferische omstandigheden, een langs zwaaiende tak, of voor wat voor reden dan ook, worden deze metingen niet altijd geregistreerd. Dit leidt ertoe dat er soms ± 32, ± 48 minuten of langer geen metingen binnenkomen.

Dit tijdsbestek moet genomen worden in de context van een heatmap die gegevens van dat moment moet weergeven. Als de heatmap alleen maar data weergeeft van 1 seconde oud, zal de kaart er vrij leeg uit zien. Hieruit kun je dus concluderen dat er een bepaalde ‘tijdsmarge’ moet zijn voor de metingen die weergeven worden.

Stel je wilt de heatmap van temperatuur zien op 18 december 2023 om 15:00 uur, is een meting van 14:55 uur op diezelfde dag acceptabel om weer te geven op deze kaart? En wat denk je van om 14:45 uur, of 14:00 uur? Wat is de grens tussen relevante en irrelevante metingen?

Die grens hebben wij gezet op **35 minuten, waarbij ALLEEN gekeken wordt in het verleden**. De redenering voor dit getal heeft te maken met de frequentie van metingen (± 16 min), met 35 minuten is er namelijk altijd data aanwezig voor wanneer een meetstation één meting mist. Dit houdt in dat voor ons voorbeeld op 18 dec. 23 15:00 uur alle metingen tussen 14:25 en 15:00 meegenomen mogen worden als data van 15:00 uur.

Hierbij komt het volgende probleem. Wat als een meetstation niet 1 maar 3 metingen in dit tijdsbestek heeft? In dit geval zullen de metingen gefilterd worden om alleen de meting met de dichtstbijzijnde timestamp te behouden, zo wordt dus alleen maar de meest relevante meting gebruikt. Om terug te komen op ons voorbeeld, houdt dat dus in dat van een meetstation met metingen op 14:26, 14:42, en 14:57 uur alleen de meting van 14:57 uur gebruikt wordt op de heatmap van 15:00 uur.

De parameter voor tijdsmarge wordt in de code ‘minuteLimit’ genoemd en is makkelijk aan te passen, mocht deze ooit veranderd moeten worden.

# Omgang met foutieve metingen

Tijdens de ontwikkeling is naar voren gekomen dat de metingen afkomstig van MeetJeStad af en toe onmogelijke of onvolledige data bevatten. Deze afwijkingen zijn met de stakeholders besproken en er is geconcludeerd dat deze foutieve data deels aangevuld en deels weg gefilterd moet worden.

## Correctie van locatiegegevens

Het komt regelmatig voor dat meetstations voor een willekeurige periode hun metingen doorsturen zonder locatiegegevens. Het vermoeden is dat dit veroorzaakt wordt door tijdelijke verstoringen, aangezien dit meestal maar enkele uren duurt. Vanuit de opdrachtgevers is de wens om deze incomplete data aan te vullen met de laatst bekende locatie van meetstations.

Er is gezocht voor een oplossing via directe calls naar de MeetJeStad API, maar dit bleek alleen mogelijk op een zeer inefficiënte manier. Er zou dan voor elke incomplete foutieve meting een nieuwe request moeten worden gestuurd naar de MeetJeStadAPI die de geschiedenis van dat station opvraagt. Als er dan nog steeds geen meting met locatie is moet er opnieuw een langere geschiedenis opgevraagd worden, etc.

Met een data caching feature zou dit echter wel goed mogelijk zijn, omdat de laatst bekende locatie dan makkelijk op te halen is uit eigen database en omdat dit maar 1 keer hoeft te gebeuren voor elke foutieve meting.

Dit is besproken met de stakeholders en het is besloten dat dit probleem verholpen zal worden in een toekomstige feature, zie het bestand ‘Data caching’.

## Filteren van onrealistische data

Een gedeelte van de data die binnenkomt is onrealistisch, deze foutieve data wordt in de meeste features eruit gefilterd.

**Moment van filteren**

Momenteel worden de foutieve metingen weg gefilterd uit de heatmap en wijkkaart, maar nog niet uit de geschiedenis grafieken.

Het filter moet uiteindelijk wel toegepast worden op de geschiedenis grafiek van wijken, maar specifiek niet op die van meetstations zodat men zelf kan spotten dat hun meetstation foutieve metingen geeft.

Temperatuur

Sommige meetstations geven, vergeleken met de rest, zulke hoge of lage temperatuurwaarden door, dat deze logischerwijs alleen maar als meetfouten beschouwt kunnen worden. Dit kan komen doordat een meetstation niet volgens de richtlijnen is opgehangen, bijvoorbeeld binnenshuis of naast een kachel. Deze metingen mogen niet gebruikt worden.

Als methodiek is gekozen om een bandpass filter toe te passen. Dit is een snelle, betrouwbare en makkelijk instelbare manier om uitschieters uit een dataset te filteren. De grens waarbij metingen worden gezien als meetfout is makkelijk in te stellen via parameters in de filter. Verder houdt de filter goed rekening met negatieve waardes, waardes dicht bij nul en natuurlijk hoge waardes.

Luchtvochtigheid

De sensoren voor luchtvochtigheid in de meetstations lijken voor een groot deel niet goed te werken. Zo geven sommige meetstations waardes door onder de 0% of boven de 100% luchtvochtigheid. Deze onmogelijke waardes worden in Ontdekstation013 vlakuit weg gefilterd. Er wordt momenteel niet verder gekeken naar de metingen die binnen de range 0-100% vallen, omdat deze zelf ook erg uiteenlopend zijn.

# Timelapse

Een gebruiker kan op een bepaald moment de temperatuur op de kaart zien. Om verschillen in temperatuur per periode te zien, kan de gebruiker zelf de dag en het tijdstip instellen. Een oplossing om dit gebruiksvriendelijker te maken, is een timelapse.

De ontwikkeling van de temperatuur over een periode zou dan duidelijker en sneller te zien zijn. Hierbij kan het ontstaan van bijvoorbeeld warme of koude temperatuureilanden gevisualiseerd worden.

De timelapse is op dit moment niet geïmplementeerd, maar het is mogelijk binnen de heatmap package zelf. Overweeg om van de timelapse een React-component te maken om te voldoen aan best practices voor React. Dit is een opzet, verder onderzoek voor deze feature is nog vereist.

Een voorbeeld en uitleg over de implementatie zijn te vinden in de onderstaande link: <https://www.patrick-wied.at/static/heatmapjs/example-heatmap-animation.html>

