

Project WPoSaD

In dit individuele project ontwikkelen jullie gedurende de laatste 7 weken van semester 1 een webtoepassing, op basis van de kennis en vaardigheden, die jullie gedurende de eerste 7 weken hebben verworven. Daarnaast zullen jullie ook een aantal dingen zelf moeten uitzoeken bv. praktische configuratie Arduino, samenstellen elektronische schakeling en/of zelf uitwerken van een multi-container environment in Docker.

Er zijn 2 projectvarianten waartussen jullie kunnen kiezen.

- De INFRA-variant betreft de ontwikkeling van een webtoepassing in een Docker-omgeving, waarbij de opdracht ook enkele uitdagingen bevat m.b.t. het samenstellen en configureren van systemen.
- De ELO-variant betreft de ontwikkeling van een webtoepassing met Arduino

Opgelet: naast deze insteek, blijft **webprogrammatie** één van de belangrijke doelstellingen van het project. Je zal dus zelf code (PHP en/of Javascript en/of node.js - zie verder) moeten schrijven en je zal daar ook op worden beoordeeld.

De keuze is **volledig vrij**; en staat dus los van jullie persoonlijke achtergrond(kennis).

Variant 1: webtoepassing in Docker

In deze variant ontwikkelen jullie een proof-of-concept van een leuke webapplicatie. Deze applicatie zal

1. iets te maken hebben met adressen en coördinaten, want jullie zullen gebruik maken van een nominatim-server.
2. e-mails verzenden naar eindgebruikers, al zullen die in jullie systeem onderschept worden door een mailhog-server.

Nominatim

Een nominatim-server is een voorgemaakte software-component die zeer snel adressen en plaatsnamen kan vertalen naar geo-coördinaten (= geocoding) en omgekeerd (= reverse geocoding). Nominatim biedt deze diensten aan in de vorm van een Web API (coördinaten/adressen in de URL, en het antwoord in JSON).

Nominatim kan maar zo snel zijn omdat de coördinaten en adressen op voorhand uitgebreid in een databank werden geïmporteerd en geïndexeerd, gedurende een importfase. Pas daarna wordt de Web API gestart.

De import en configuratie staat o.a. op <https://hub.docker.com/r/mediagis/nominatim/> beschreven en heeft als input een landkaart van een land, regio, ... in de vorm van een OpenStreetMap-bestand.

OpenStreetMap is als het ware *de Wikipedia van de landkaarten*, m.a.w. een grote webapplicatie

waar iedereen de wereldkaart kan bewerken en bekijken. Nominatim werkt met .osm.pbf-bestanden (protobuf-formaat).

De laatste versies van deze bestanden worden gratis ter beschikking gesteld door <https://download.geofabrik.de/> waar je ze op land- en regioniveau kan downloaden. Je kan het download-adres ook gewoon doorgeven aan een nominatim-container. Je test best met zo klein mogelijke regio's (!).

Op <https://nominatim.org/release-docs/latest/api/Overview/> vind je een beschrijving hoe je de uiteindelijk Web API gebruikt. Je mag kiezen of je deze Web API aanspreekt vanuit je Javascript-omgeving (fetch) of vanuit je webserver (bv. met PHP cURL).

Mailhog

Mailhog is een e-mail-testing-tool voor ontwikkelaars en bestaat uit een fake SMTP-server en een webinterface. E-mails die vanuit de PHP-webapplicatie naar de SMTP-server van Mailhog worden verzonden, worden opgevangen en getoond in de webinterface.

Concreet stellen jullie een multi-container environment samen in Docker (door zelf de docker-compose.yml en eventuele Dockerfiles samen te stellen) bestaande uit volgende containers:

- mediagis/nominatim
- mailhog
- nginx (webserver)
- php-fpm
- mysql
- ev. phpmyadmin

Het is de bedoeling dat je aan het einde van de rit een Docker-omgeving hebt, die bij het aanroepen van “docker-compose up” automatisch de nominatim-import-data gebruikt, de DB-dump importeert, enz. zodanig dat de webapplicatie automatisch werkt.

We verwachten een proof-of-concept van een leuke webapplicatie

- waarbij vanuit PHP data wordt gelezen uit en geschreven naar een MySQL-databank
- waar gebruikers kunnen in- en uitloggen (PHP)
- waar voorzien wordt dat gebruikers e-mails ontvangen, die worden onderschept door mailhog
- waar er ofwel vanuit PHP, ofwel vanuit de JS-omgeving adressen of coördinaten worden opgevraagd bij de nominatim-server
- waar er gebruik wordt gemaakt van client-side JavaScript-code, hetzij eigen code om de DOM te manipuleren, hetzij code waarbij je gebruik maakt van een library om een interactieve landkaart te tonen bv. Leaflet

Voorbeelden van een webapplicatie zijn: een website waar gebruikers toeristische plaatsen/leuke adresjes privé kunnen opslaan en delen met andere gebruikers, een website om het vervoer van pakketjes aan te vragen en uit te voeren, een restaurant-review-website, een spelwebsite waarbij

gebruikers binnen een bepaalde tijd een plaatsbeschrijving zo dicht op de kaart dienen aan te klikken ...

We verwachten dat de details proper zijn uitgewerkt:

- Mooi en bruikbaar webdesign (de CSS mag je van ergens overnemen)
- De HTML die je genereert is steeds W3C-valid
- De Javascript genereert geen fouten noch output in de Console
- Tabel(len) in databank hebben correcte structuur
- Je volgt de algemene coderingsinstructies (PHP, javascript, ...) die je in de workshop gezien hebt gedurende de eerste 7 weken

Variant 2: webtoepassing met Arduino Uno

In deze variant ontwikkelen jullie een proof-of-concept van een leuke webapplicatie die interface met de Arduino Uno, die op zich uitgerust is met één of meerdere interessante componenten bv. een hele GRID van LEDs, barometer/temperatuurmeter, ventilator, split-flap display, 7-segment display, bodemvochtigheidssensor, IR led voor TV, ... Je kan de Arduino ook uitrusten met een extra shield bv GPS shield, MP3 player shield, MIDI shield, ...

Concreet maken jullie gebruik van 1 van volgende configuraties.

	Configuratie	Welke sketch wordt uitgevoerd op Arduino?	Welke code/componenten verwachten we van jou?
2A	Op je laptop draait een node.js server die via de Johnny-Five library communiceert met de Arduino.	StandardFirmataPlus	<ul style="list-style-type: none"> - node.js/express - databank SQLite/MySQL - client-side JavaScript
2B	Op je laptop draait een webserver met PHP die communiceert via een library zoals lepiaf/serialport met de Arduino over de seriële poort	jouw eigen sketch die binnenkomende vragen via seriële communicatie verwerkt	<ul style="list-style-type: none"> - PHP-code - databank MySQL - C-sketch die seriële comm. verwerkt - client-side JavaScript
2C	De webserver draait op een Arduino <u>uitgerust met Ethernet Shield</u> . Jij ontwikkelt custom low-level webserver-code voor de Arduino. Je configureert het Mac- en IP-adres en verbindt de Arduino in het lokale netwerk.	jouw webserver-code	<ul style="list-style-type: none"> - C-sketch die HTTP-responses craft op basis van de inkomende HTTP-requests - client-side JavaScript

Je webapplicatie bestaat minstens uit:

- **3 dynamische webpagina's** ontwikkeld in server-side code: node/express (2A), PHP (2B) of C (2C).
- **2A/2B: een databank**: de server-side code schrijft naar en leest uit een MySQL- of SQLite-systeem
- een stukje **client-side JavaScript¹ voor DOM-manipulatie** dat jullie zelf geschreven hebben en dat de gebruikerservaring van de webapplicatie verbetert (bv. polling en sensoren en dynamisch aanpassen, bv bij elkaar klikken van een patroon zonder dat de webpagina moet herladen)

¹ de JavaScript-code is deel van de webapplicatie en staat in een apart bestand. Variant 2C: je kan overwegen om deze script op een andere server te plaatsen en toelating te geven a.d.h.v CORS

We verwachten geen groot project, maar verwachten wel dat je project meer kan dan een LED aan en uitzetten. Zo verwachten we wel dat je webapplicatie een aantal zaken kan 'onthouden':

- als je vooral met sensoren werkt, zal dat eerder de historiek van de sensordata zijn
- als je vooral componenten aanstuurt, kunnen dat terugkerende patronen in de aansturing zijn bv. te volgen pad door wagen, oplichtingspatroon leds, config tv-afstandsbediening

We verwachten dat de details proper zijn uitgewerkt:

- Mooi en bruikbaar webdesign (de CSS mag je van ergens overnemen)
- Opvangen van foutsituaties (in parameters van webpagina maar bv. ook in sensoren)
- De HTML die je genereert is steeds W3C-valid
- De Javascript genereert geen fouten noch output in de Console
- Eventuele tabel(len) in databank hebben correcte structuur
- Je volgt de algemene coderingsinstructies (PHP, javascript, ...) die je in de workshop gezien hebt gedurende de eerste 7 weken

Materiaal

- We gaan ervan uit dat jullie reeds beschikken over een Arduino Uno (Programmeren 1)
- Componenten/shields zijn zeer specifiek en dienen jullie zelf aan te schaffen
- Variant 2C: een Arduino met Ethernet Shield kan worden uitgeleend bij ons op campus

Werkmethoden en deadlines

- Jullie worden meermaals verwacht tijdens de co-workingsessies (di 11u15-13u15) voor een korte statusupdate (aanwezigheid verplicht LW9, LW10, LW12, LW14). Op andere momenten kunnen jullie er terecht om te werken en vragen te stellen.
- Op MS Teams zal voor elke projectvariant een kanaal worden aangemaakt waar jullie praktische tips kunnen uitwisselen met de docenten en met elkaar. De eerste technische aanknopingspunten zullen de docenten ook daar posten. Daarnaast kunnen jullie ook de docenten contacteren als jullie echt vastzitten.
- Het opleveren van de statusupdates en opvolgen van de timeline tellen mee voor 10% van het projecttotaal

Agenda

- LW8: voorstelling projectopgave & begeleiding projectkeuze
- LW9: statusupdate: zorg ervoor dat jullie
 - een projectvariant hebben gekozen
 - een goed beeld kunnen geven van de webapplicatie die jullie zullen uitwerken: wat doet ze juist, waarvoor dient ze, uit welke pagina's/formulieren bestaat ze, databankschema
 - variant 2x: een bestellijst van de nodige componenten hebben klaarliggen
 - een eerste test hebben gedaan van jullie omgeving
 - variant 1: een test uitgevoerd met nominatim-import en -API en met mailhog
 - variant 2: een testscript in de specifieke configuratie die de Arduino kan aansturen
- LW10: bespreken timeline (wat wil ik tegen wanneer af hebben?)
- LW11: uitwerking project
- LW12: statusupdate (a.d.h.v. de timeline)
- LW13: uitwerking project (laatste kans om problemen op te lossen, wacht niet tot dan!)
- FINALE DEADLINE
 - zaterdag 14 januari 2023 om 20u: inzending alle code
 - dinsdag 17 januari 2023: mond. bespreking en demonstratie

Tips. Probeer een realistisch project voor te stellen dat toch voldoende vlees aan het been heeft om te bewijzen dat je de geziene leerstof onder de knie hebt. Denk daarom goed na over de timeline die je moet voorleggen. Werk naar een MVP (minimal viable product) en zorg dat dat zeker haalbaar is. Eens dat bereikt is kan je nadenken over uitbreidingen.