Programming miniproject:

NS Kaartautomaat door: Atal, Achraf, Daniel en Marnick



[**Inleiding**](#_in1hsk5wxrih) **3**

[**Opdracht**](#_ikjjchu5ycce) **4**

[**Taakverdeling en project groep**](#_dgy681tq1da2) **5**

[Atal:](#_j7e274ve3gid) 5

[Create\_Departures\_Frame uitbreiden:](#_8jyse3tvynjl) 5

[Show\_departures:](#_cgme0sw8n6cq) 5

[(Extra)Uitgebreidere GUI](#_dqn2fx2nse6i) 5

[Achraf:](#_d97vv54gla7a) 5

[NS-API & load\_departures\_NS:](#_8ak1i9xzz37d) 5

[Breid Show\_departures uit:](#_lqp18cnmyybw) 6

[(Extra)Uitgebreidere GUI](#_5x1a64r49vkp) 6

[Daniel:](#_eayfy4wsnjrr) 6

[Start Scherm](#_xr6val7znxa) 6

[Show\_application:](#_gsio8o1og3as) 6

[(Extra) Mooiere GUI](#_usrl787gdl4s) 6

[Marnick:](#_suhgnnskzq96) 7

[Eindverslag:](#_k0kluymyiybt) 7

[Aandacht voor documentatie:](#_7lmwbdpbk87m) 7

[Hardware-componenten:](#_jcmnlrdllzw4) 7

[Onderhoud ontwikkelomgeving:](#_6donwtpgwq63) 7

[**Flowchart:**](#_vns4gktcd0mw) **8**

# Inleiding

Voor het vak programming moest een keuze project gemaakt worden. Hierbij was er keuze uit een aantal opgestelde projecten. Wij hebben voor het project “NS Kaartautomaat” gekozen. Dit leek ons de meest geschikte opdracht voor onze groep. Al hadden we een strikte deadline en een korte tijd om het product te realiseren zijn wij snel en nauwkeurig aan de slag gegaan. Hierbij introduceren wij, de NS Kaartautomaat om het volgende toe te passen:

“Om de Servicemedewerkers op de stations te ontlasten bestaat de wens om de reiziger op de NS Kaartverkoopautomaat de mogelijkheid te geven de actuele vertrekinformatie op te vragen. Dezelfde functie bestaat reeds op de NS website en in de Reisplanner app. Een grote groep reizigers benadert op de stations nog steeds de Servicemedewerkers met vragen over vertrektijden, vertragingen en vertreksporen. Door dezelfde functie ook onder te brengen in de NS Kaartautomaat is het de verwachting dat de druk op Servicemedewerkers afneemt.”

# Opdracht

Het is de wens van de NS om op het startscherm van de automaat een “ingang” te creëren naar de nieuwe functie waarmee de klant actuele reisinformatie kan opvragen. Hierbij moet in ieder geval de reisinformatie getoond kunnen worden van het station waarop de kaartverkoopautomaat zich bevindt. Als uitbreiding hierop moet het ook mogelijk zijn om voor een ander Nederlands station de informatie op te vragen. De gebruiker moet hiervoor een stations**naam** (geen afkorting) kunnen invoeren of kiezen.

Voor het ophalen van de lijst met stations en het ophalen van de actuele vertrekinformatie is een NS API beschikbaar. Meer informatie over deze API is te lezen op de NS website ([https://apiportal.ns.nl/ (Koppelingen naar een externe site.)](https://apiportal.ns.nl/)).

In figuur 3 is de layout van het het huidige startscherm van de automaat weergegeven. Bepaal zelf welke informatie die de API biedt nuttig is voor de reiziger en hoe dit het beste (in een Grafische UserInterface (GUI) gepresenteerd kan worden op het scherm. Houd hierbij in gedachten dat alle typen reizigers hiermee overweg moeten kunnen. Leesbaarheid (lettergrootte, font, kleurstelling) is hierbij erg belangrijk.

# Taakverdeling en project groep

Wij hebben de opdracht geanalyseerd en daar de volgende verdeling en taken van gemaakt:

## Atal:

### Create\_Departures\_Frame uitbreiden:

Voeg een Entry toe (voor invoeren van stationscodes)

Voeg een button toe (om vertrek tijden te laden (hoeft nog niet te werken))

Voeg een text-widget toe om vertrektijden te tonen (en zet hier dummy tekst in)

Plaats een button om terug naar het hoofdscherm te gaan. Laat deze werken (kijk naar de gegeven code command-parameter)

### Show\_departures:

txt\_output: de widget waar de resultaten in getoond moeten worden.  
departure: het station waarvoor tijden geladen moeten worden.  
Zorg dat de button 'vertrektijden laden' zorgt voor een aanroep van de functie show\_departures, waarbij je de benodigde informatie doorgeeft (text-widget, en de tekst in de Entry). Gebruik een lambda-functie

### (Extra)Uitgebreidere GUI

Verander input van stationscode naar stations naam.

## Achraf:

### NS-API & load\_departures\_NS:

Roep de NS-API aan om de actuele vertrektijden van een station op te halen. Gebruik parameter 'station' als stationscode die je met het request meestuurt

Zorg dat je functie de departures (list) returned. De functie zelf hoeft dus niets te printen

### 

### Breid Show\_departures uit:

Roep de functie load\_departures\_NS aan, en sla het resultaat op in een variable! Geef parameter departure door, want die bevat de tekst die de gebruiker heeft ingevoerd!  
Door 'loop' de resultaten, toon de vertrektijden in de Text-widget (parameter txt\_output),  
Het programma mag niet crashen (verkeerde invoer)  
Wis oude vertrektijden uit de text-widget

### (Extra)Uitgebreidere GUI

Verander input van stationscode naar stations naam.

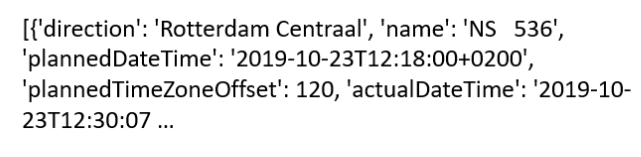
## Daniel:

### Start Scherm

Het inrichten en ontwikkelen van het startscherm van de applicatie.

### Show\_application:

Plaats print(load\_departures\_NS("Ut")) voor Show\_application()

Voorbeeld van de uitvoer: 

### (Extra) Mooiere GUI

Het uitbreiden van de GUI door NS-layout toe te passen en verbeteren van eventueel andere styling

## Marnick:

### Eindverslag:

Het opstellen en uitwerken van het eindverslag. Hier in komt een samenvatting van de opracht, taakverdeling, en flowchart te staan van de werk van de applicatie.

### Aandacht voor documentatie:

Het uitwerken en uitleggen van de technische documentatie van de code. Hierdoor is voor het huidige en eventueel toekomstige ontwikkelteam duidelijk hoe de code werkt en wat er overal gebeurd.

### Hardware-componenten:

Voor deze (optionele) deelopdracht moet minimaal een LED-lampje gekoppeld worden aan de opgeleverde software. Dit lampje kan gaan branden als...

1. ... de zuil de meest recente vertrektijden aan het downloaden is.
2. ... de gebruiker ongeldige invoer geeft.
3. ... er over 1 minuut een trein gaat vertrekken (je kunt in de zuil eventueel melden welke trein).

Kies één van de bovenstaande opties, of bedenk zelf een betere en bespreek dit met je docent.

*De opgeleverde software mag als één geheel programma op de Raspberry Pi uitgevoerd worden. Het kan echter handig zijn om de software in twee delen op te leveren. Bijvoorbeeld 1 applicatie op de Raspberry Pi, die via een API-URL aangeroepen kan worden om het LED-lampje (eenmalig) te laten knipperen. Een tweede applicatie op je laptop kan dan de gebruikersinterface bevatten, en het lampje laten knipperen door de URL aan te roepen.*

### Onderhoud ontwikkelomgeving:

Onderhouden van Trello (ticket board) en de GIT repo. Dit betekend ook het oplossen van merge conflicts en andere zaken.

# Flowchart:

# Werking van het programma:

Start het programma: dit kan door in het bestand rechtermuisknop te gebruiken en op run code te drukken. Vervolgens druk je op de knop “toon vertrektijden” en voer je in het bovenste veld de stationscode in. Daarna druk je op vertrektijden laden en worden alle actuele vertrektijden geladen. Ook worden van vorige zoekopdrachten de vertrektijden weggehaald zodat er geen verwarring onstaat.