

**PROBLEEMOPLOSSEN EN ONTWERPEN, DEEL 3**

***CWB2***

*Schrooten Bernd*

*Schryvers Andreas*

*Sels Shoera*

*Switsers Sander*

*Van den Berghe Pieter*

*Van Laere Nathan*

**Quantified Bike**

TUSSENTIJDS VERSLAG

Co-titularis

Professor Duval E.

Begeleider(s)

Charleer Sven

De Croon Robin

Klerkx Joris

Santon Jose Luis

ACADEMIEJAAR 2014-2015

# INHOUDSTAFEL

[INHOUDSTAFEL 2](#_Toc401935194)

[1. Groepsleden 3](#_Toc401935195)

[2. Brainstorm 3](#_Toc401935196)

[3. User stories 4](#_Toc401935197)

[User story 1 4](#_Toc401935198)

[User story 2 4](#_Toc401935199)

[User story 3 4](#_Toc401935200)

[Productbeschrijving 5](#_Toc401935201)

[4. Architectuur 5](#_Toc401935202)

[5. Technologie 6](#_Toc401935203)

[Raspberry Pi 6](#_Toc401935204)

[Arduino 6](#_Toc401935205)

[JavaScript 7](#_Toc401935206)

[jQuery 7](#_Toc401935207)

[JSON 7](#_Toc401935208)

[Visualisatie bibliotheken 7](#_Toc401935209)

[6. Vakintegratie 8](#_Toc401935210)

[7. Besluit 8](#_Toc401935211)

[8. Appendix 1: geleverde werk 9](#_Toc401935212)

[9. Appendix 2: Planning 9](#_Toc401935213)

[10. Appendix 3: oplijsting brainstormsessie 10](#_Toc401935214)

[11. Bibliografie 11](#_Toc401935215)

# Groepsleden

Alle groepsleden volgen de bachelor in de ingenieurswetenschappen. Volgende lijst vat de nodige gegevens samen.

**Naam Jaar Richting (major – minor)**

Schrooten Bernd 2 Computerwetenschappen - Elektrotechniek

Schryvers Andreas 2 Computerwetenschappen - Elektrotechniek

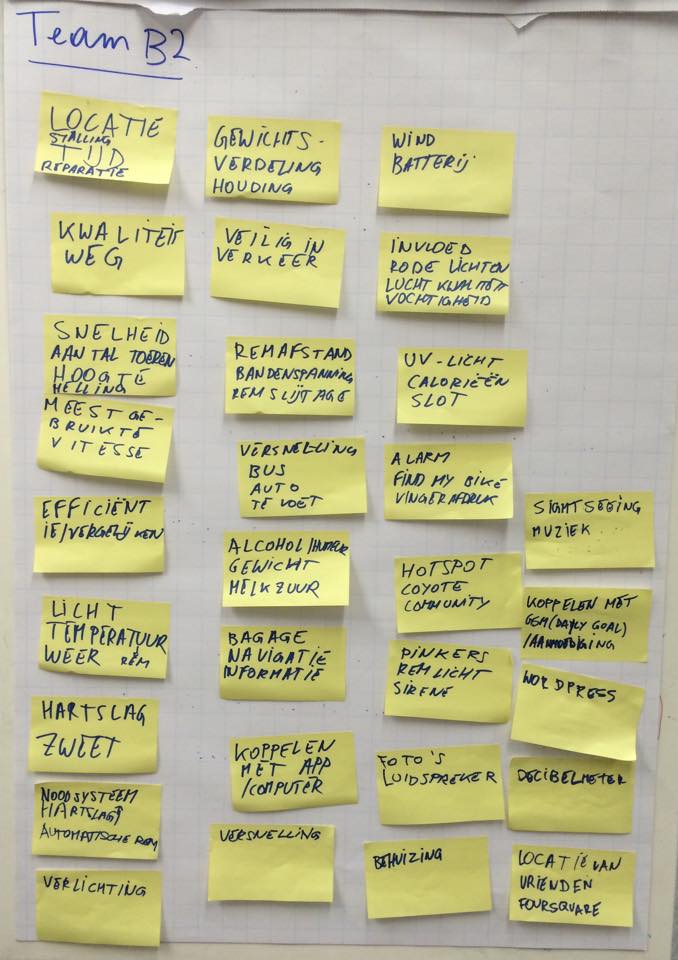
Sels Shoera 2/3 Elektrotechniek - Computerwetenschappen

Switsers Sander 2 Computerwetenschappen - Elektrotechniek

Van den Berghe Pieter 2 Elektrotechniek - Computerwetenschappen

Van Laere Nathan 2 Computerwetenschappen - Elektrotechniek

# Brainstorm

Afbeelding 1 toont het resultaat van de eerste brainstormsessie, waar het de bedoeling was om in eerste instantie na te gaan wat allemaal gemeten zou kunnen worden en hoe feedback aan de gebruiker gegeven zou kunnen worden. Appendix 3 bevat een oplijsting van alle elementen uit deze brainstorm met bijhorende beschrijving (indien nodig), opgedeeld in enkele categorieën.

Afbeelding 1: Brainstormsessie

Bij onze eerste brainstormsessie lag de nadruk te hard op het creëren van (navigatie)diensten voor de gebruiker, en te weinig op de eigenlijke opdracht: het vergaren en in beeld brengen van informatie. Hierdoor vielen al een heleboel ideeën weg, namelijk alles onder de categorie diensten *(zie Appendix 3).* Bij de categorie gezondheid was er weinig wat we echt zouden kunnen meten, behalve de hartslag. Bovendien zou het quasi onhaalbaar zijn om alles onder de categorie locatie en omstandigheden te meten. Om die reden werd besloten om (momenteel) enkel de voor ons relevante gegevens te meten: locatie, snelheid, hoogte en foto’s van de omgeving. Indien de tijd het toelaat komt daar nog temperatuur bij. De insteek van de toepassing, die de naam Bike-Lapse krijgt, is veranderd: tijdens de eerste brainstormsessie werd er over een dienst/augemented reality toepassing nagedacht. Na feedback van de assistenten veranderde dit: momenteel is het de bedoeling om een fotologboek van fietstochten bij te houden. Aan de gebruiker wordt ook feedback over zijn snelheid gegeven. Hierover meer in het volgend deel.

# User stories

## User story 1

Jans grootmoeder komt op bezoek. Net als hij heeft ze een passie voor fietsen. Jan vertelt haar uitgebreid over zijn laatste fietstocht. Hij was de weg kwijt geraakt in het bos, afgeleid door de prachtige natuur om hem heen. Gelukkig had hij zijn fietst uitgerust met de laatste technologie. Een klein apparaat op zijn fiets neemt nu en dan een foto en houdt bovendien zijn locatie en andere informatie bij. Hierdoor was Jan in staat om de weg terug te vinden. Bovendien kan hij nu die tocht aan z’n grootmoeder laten zien, zowel op een kaart als met behulp van een time-lapse. Zij snapt niet hoe dit allemaal werkt, maar ook zij is verbaasd!

## User story 2

Piet studeert burgerlijk ingenieur in de computerwetenschappen aan de KU Leuven en gaat elke dag met de fiets naar de les. Hij heeft de slechte gewoonte om te laat in de les aan te komen en professor Dutré kan daar niet meer mee lachen. Laatst kreeg hij zelfs een krijtje naar zijn hoofd geslingerd. Om dit probleem op te lossen maakt hij gebruik van Bike-Lapse. Deze toepassing houdt zijn gemiddelde snelheid bij en geeft, met behulp van lichtjes op zijn stuur, aan wanneer hij sneller of trager rijdt dan gemiddeld. De lichtjes worden rood wanneer hij trager rijdt en groen wanneer hij sneller rijdt. Op die manier wordt hij aangemoedigd om steeds sneller te rijden dan zijn gemiddelde snelheid. Bovendien kan Piet via een webapplicatie verschillende fietstochten vergelijken om zo de voor hem efficiëntste weg te vinden. Sinds hij Bike-Lapse gebruikt, hoeft hij niet meer te vrezen voor de krijtjes van professor Dutré.

## User story 3

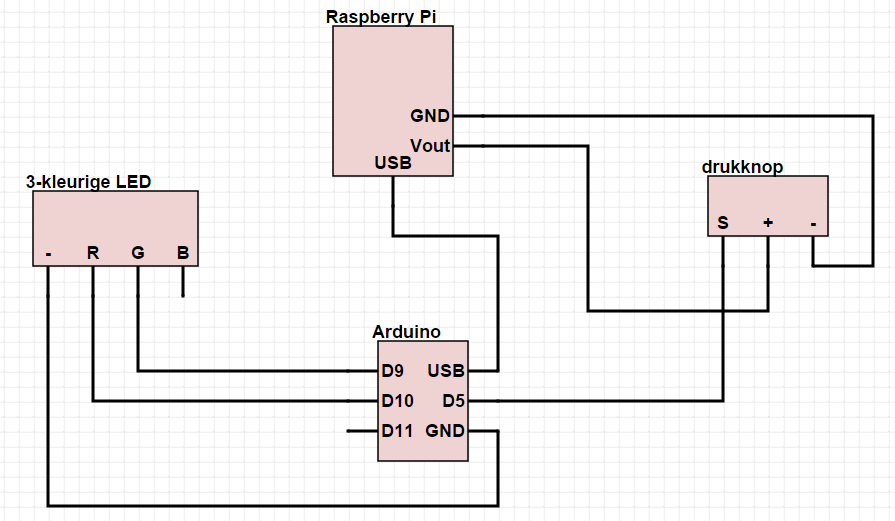
Joris is enorm gefocust op het bijhouden van allerhande informatie. Zo wil hij graag weten waar hij is geweest, wanneer dat was, hoe warm het was etc. In de goede oude tijd had hij altijd een notitieboekje bij, waarin hij zulke informatie noteerde. Hij was hier zo op gefocust dat hij altijd andere zaken vergat en op zijn werk consequent dossiers te laat indiende. Wanneer zijn baas dreigt met ontslag, beslist hij om iets te ondernemen. Op aanraden van een vriend installeert hij Bike-Lapse. Joris’ informatie wordt automatisch bijgehouden tijdens het fietsen en op een mooie manier gepresenteerd op de webtoepassing. Vanaf nu kan hij op een gestructureerde manier data van voorbije trips opvragen, zonder dat hij uren verliest aan zelf alles bij te houden.

## Productbeschrijving

Met onze applicatie bent u in staat om allerlei informatie over uw fietstochten bij te houden en te raadplegen op een webapplicatie. Wat onze applicatie onderscheidt van andere, is dat u een fotologboek van uw fietstochten kan bijhouden. Zo kan u met een time-lapse elke fietstocht herbeleven en tonen aan vrienden en familie. Bovendien krijgt u tijdens uw fietstocht feedback over uw snelheid op een bepaalde locatie: zo weet u of u trager of sneller rijdt dan gewoonlijk.

# Architectuur

De Arduino Nano is aangesloten op de Raspberry Pi via een USB-kabel. Een cameramodule staat op de CSI-poort van de Raspberry Pi. Op de Arduino zelf zijn een drukknop en een driekleurige LED aangesloten. Deze laatste is verbonden via drie digitale pinnen, die tevens voor stroomtoevoer zorgen. Elke pin vertegenwoordigt een ander kleurtje van de LED. De drie mogelijke kleuren zijn blauw, rood en groen, maar de blauwe component is in deze opstelling niet in gebruik. De drukknop krijgt rechtstreeks van de Arduino de nodige stroom en is aangesloten aan de grond. De drukknop stuurt een signaal naar de Arduino door een verbinding via een digitale pin. De Arduino is zo geprogrammeerd dat wanneer de drukknop wordt ingedrukt, de LED op groen springt. Zolang de LED groen is, worden er om de vijf seconden foto’s getrokken door de camera. Wanneer de drukknop opnieuw wordt ingedrukt, wordt de LED rood en stopt de camera met foto’s te trekken. De Raspberry Pi stuurt de getrokken foto’s naar de server, waarna deze kunnen weergegeven worden op de site.

Afbeelding 2 geeft een overzicht van de verbindingen tussen Raspberry Pi en Arduino en tussen de componenten op de Arduino en de Arduino zelf.

Afbeelding 2: schema met verbindingen tussen Raspberry Pi en Arduino en tussen de componenten op de Arduino en de Arduino zelf

# Technologie

## Raspberry Pi

De Raspberry Pi is een goedkope, op ARM-processors gebaseerde singleboardcomputer. Het heeft de grootte van een kredietkaart en is ontwikkeld door de Raspberry Pi Foundation met de bedoeling om computerwetenschappen op school te stimuleren. [1] De Raspberry Pi heeft vele aansluitmogelijkheden: niet alleen de “basis” aansluitingen zoals USB, HDMI en Ethernet zijn aanwezig, men kan ook een camera, een accelerometer, en temperatuurmeter en nog veel meer aansluiten, meestal via de digitale pinnen op het board. Ook qua software zijn de mogelijkheden legio: meer dan tien verschillende besturingssystemen zijn te gebruiken. Naast deze vele voordelen, heeft de Raspberry Pi ook enkele nadelen: het gebruik vraagt wat kennis van Linux, de rekenkracht is niet enorm en er is geen ingebouwde Wi-Fi.

Wij gebruiken de Raspberry Pi in onze toepassing in combinatie met een Arduino *(zie verder).* De Raspberry Pi verzamelt gegevens via (eigen) sensoren (camera, GPS), verwerkt deze en stuurt ze ten slotte door naar onze webapplicatie.

## Arduino

Arduino is een opensource-computerplatform dat is opgebouwd rond de ATmega168-microcontroller van Atmel en het softwareontwikkelplatform Processing (een uitbreiding van Java). Het Arduinoplatform kent allerlei uitvoeringen van de hardware rond de Atmega168 of Atmega328. Met Arduino is het mogelijk apparaten en objecten te creëren die reageren op hun omgeving door middel van digitale en analoge inputsignalen. Input kan bijvoorbeeld worden gegenereerd door schakelaars, lichtsensoren, bewegingsmeters, afstandsmeters, temperatuursensoren etc. Outputsignalen kunnen bijvoorbeeld motoren, lampjes, pompjes en beeldschermen aansturen, maar ook input genereren voor een andere Arduinomodule. [2] Die beschikbaarheid van enorm veel aansluitmogelijkheden is een van de grote voordelen van de Arduino. Qua software is het ook toegankelijker dan een Pi: de Arduino kan bestuurd worden via Windows, OS X en Linux. De Arduino is geen computer zoals de Pi: er is geen beeldscherm en internetaansluiting (doch beschikbaar via shields) en er draait niet echt een besturingssysteem op. De Arduino wordt namelijk bestuurd en geprogrammeerd vanaf een andere computer.

Wij gebruiken in onze toepassing de Arduino Nano. Met een drukknop op de Arduino wordt de start en het einde van onze fietstochten geregeld. De Arduino wordt ook gebruikt om met behulp van een driekleurige LED feedback te geven aan de gebruiker.

## JavaScript

JavaScript wordt gebruikt in combinatie met HTML en CSS om websites te bouwen. JavaScript maakt webpagina’s interactief en dynamisch. Het grote voordeel van JavaScript is dat elke browser het ondersteunt en iedereen het gebruikt, aangezien er weinig alternatieven zijn. De alternatieven die bestaan werken via een compiler in JavaScript. Op internet vindt men vele tutorials en voorbeelden waardoor het relatief gemakkelijk is om met JavaScript te leren werken. Een nadeel is dat een functionaliteit vrij veel code vergt.

Wij gebruiken JavaScript om onze website interactief te maken: grafieken, knoppen en kaarten zijn er mee toegevoegd.

## jQuery

jQuery is een bibliotheek voor JavaScript die browseronafhankelijk is. jQuery zorgt ervoor dat vele commando’s makkelijk opgeroepen kunnen worden, bijvoorbeeld het selecteren van HTML-elementen. Bovendien maakt jQuery code veel compacter en makkelijker te lezen. Een nadeel is dat het niet altijd even goed werkt wanneer er frameworks zoals Bootstrap worden gebruikt. Voor de niet-werkende commando’s is vaak wel een alternatief voorzien binnen het framework.

Wij gebruiken JQuery om de code van onze website overzichtelijker en korter te maken.

## JSON

JSON is een gestructureerd formaat voor het versturen en het opvragen van data via een server. Het voordeel van JSON is dat het goed leesbaar is, beter dan XML. Het is mogelijk om data van anderen op te vragen en te gebruiken wanneer iedereen dezelfde structuur gebruikt. Zo is men niet beperkt tot het gebruik van enkel onze eigen gegevens. Een ander voordeel van JSON is dat het onafhankelijk is van programmeertalen: het kan met andere woorden over verschillende talen heen gebruikt worden.

Wij gebruiken JSON om ervoor te zorgen dat onze data gestructureerd naar de server wordt verstuurd.

## Visualisatie bibliotheken

Google werd gebruikt als visualisatiebibliotheek tezamen met een bibliotheek van JQuery: JQRangeSlider. Google biedt veel gemakkelijk toegankelijke en aanpasbare templates aan waar veel gebruik van is gemaakt. Een voorbeeld hiervan is Google Maps: dit wordt gebruikt om de coördinaten afkomstig van de GPS om te zetten in een route op een kaart. Een tweede voorbeeld is Google Charts: dit bevat een groot aantal grafieken die zich makkelijk laten tekenen. Er moet enkel gelet worden op de structuur van de data die wordt gegeven. Met een beperkte hoeveelheid code is het mogelijk om uitgebreide grafieken weer te geven en alle data goed te visualiseren. Google biedt ook een mogelijkheid aan om in een grafiek te scrollen maar dit werkt niet op een touchscreen. JQuery kan hier een oplossing voor bieden in de vorm van een slider die wel op een touchscreen werkt: JQRangeSlider. Hierdoor kan de site ook op een smartphone gebruikt worden.

Wij gebruiken bovengenoemde zaken voor de toepassingen die genoemd werden.

# Vakintegratie

In eerste instantie sluit het project enorm aan bij het vak *Methodiek van de informatica (Informatie)* uit het tweede semester van het eerste jaar, aangezien daar de basisbeginselen van programmeren werden aangeleerd. Bovendien wordt alles met betrekking tot de Raspberry Pi in Python geschreven, de taal die in dat vak werd gebruikt. Verder komen ook enkele belangrijke principes uit de vakken horend bij *Wiskunde* aan bod voor de snelheids- en versnellingsbepaling en voor de statistieken die bij de gegevens horen.

# Besluit

Tijdens dit project leren we o.a. hoe we moeten werken met microcomputers (Raspberry Pi, Arduino Nano), hoe we sensoren op deze microcomputers aansluiten en aansturen, hoe we de data van die sensoren kunnen extraheren en visualiseren op een webapplicatie enzovoort. Ook de beginselen van HTML, CSS, JavaScript en JQuery werden aangeleerd via tutorials en vervolgens gebruikt in onze toepassing. Verder is ook het gebruik GitHub en WebStorm uitgelegd en geïmplementeerd.

De kans is groot dat velen van ons GitHub in de toekomst nog zullen gebruiken. Het blijkt de standaard te zijn voor het delen van en samenwerken aan code. Ook WebStorm heeft zijn nut bewezen. Dit programma zorgt voor heel wat meer overzicht in de code dan bv. Kladblok. Dat sommigen onder ons nu beter overweg kunnen met HTML, CSS of JavaScript zal bovendien zeker nog van toepassing komen in het vervolg van onze studie.

Wat de technische introductiesessies betreft: deze waren zeker handig om onszelf vertrouwbaar te maken met de Raspberry Pi en de Arduino. Tijdens de sessies konden we hulp vragen aan de assistenten, en die hulp was zeker in het begin nog nodig.

Een verandering die we als assistent zouden doorvoeren: het tussentijds verslag zou best naar een iets later tijdstip worden verplaatst. We zijn eigenlijk nog maar net écht bezig met onze applicatie en het zou gemakkelijker zijn als we ons momenteel volledig hierop zouden kunnen concentreren. Een andere mogelijkheid zou zijn is om bv. enkel puntje 1, 2, 3 en 6 van het tussentijds verslag te laten indienen.

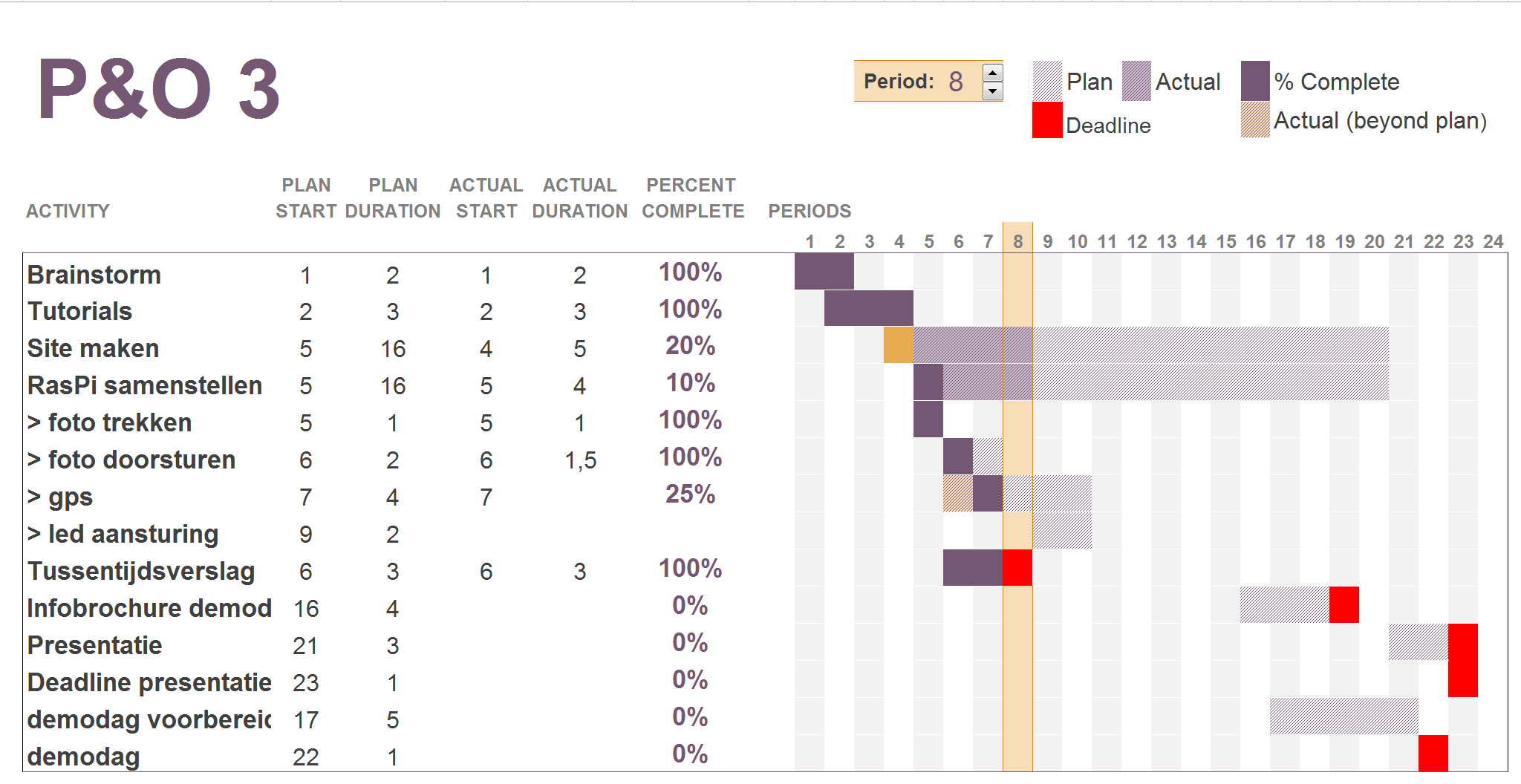
# Appendix 1: geleverde werk

Tabel 1 vat het geleverde werk per teamlid samen.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Andreas | Bernd | Nathan | Pieter | Sander | Shoera | Totaal | |
| introductiesessies + brainstorm | 14 | 14 | 14 | 14 | 14 | 14 | 84 |
| user stories | 0,5 | 0,5 | 0,5 | 0,5 | 0,5 | 0,5 | 3 |
| verslag | 10 | 0,5 |  | 1 | 2 | 2,5 | 16 |
| website |  | 13 |  | 12,5 | 3 |  | 28,5 |
| Raspberry | 3,5 |  | 13,5 |  | 8,5 | 11 | 36,5 |
| GitHub | 2 | 2 | 2 | 2 | 2 | 2 | 12 |
| Totaal | 30 | 30 | 30 | 30 | 30 | 30 | 180 |

Tabel 1: geleverde werk per teamlid

# Appendix 2: Planning

Afbeelding 3 toont de Gantt-chart.

Afbeelding 3: Gantt-chart

# Appendix 3: oplijsting brainstormsessie

**Locatie en omstandigheden**

Hoogte

Helling

Snelheid

Versnelling

Aantal toeren

Meest gebruikte versnelling

Temperatuur

Wind

Luchtvochtigheid

Luchtkwaliteit

Kwaliteit van de weg

UV-licht

Decibelmeter

Bandenstapping

Remafstand

**Foto’s**: *er* *worden foto’s van de omgeving van de gebruiker doorgestuurd*

**Gezondheid**

Hartslag

Zweet

Calorieën

Alcohol

**Melkzuur**: *hoeveelheid melkzuur in de spieren van de gebruiker*

**Gewichtsverdeling**: *de verdeling van het gewicht van de gebruiker op het frame van de fiets*

**Houding**: *informatie over de houding van de gebruiker*

**Navigatie**

**Find my Bike:** *laatste locatie van de fiets weergeven*

**Bus/auto/te voet:** *een vergelijking van de reistijd per fiets met die per bus, per de auto of te voet (mogelijk voor en na een fietstocht)*

**Pinkers:** *mogelijkheid om pinkers te gebruiken met knoppen links en rechts op het stuur*

**Remlicht:** *automatisch remlicht*

**Coyote/community:** *informatie van andere gebruikers over opstopping etc.*

**Diensten**

**Stalling:** *de mogelijkheid om de dichtstbijzijnde fietsenstallingen weer te geven*

**Reparatie:** *de mogelijkheid om de dichtstbijzijnde reparatielocaties weer te geven*

**Noodsysteem hartslag:** *wanneer de hartslag van de gebruiker ongezond hoog wordt, wordt hij hiervan verwittigd. Indien de hartslag van de gebruiker stopt, worden de hulpdiensten hiervan verwittigd, waarbij bovendien de locatie van de gebruiker doorgegeven wordt*

**Kopellen met een app:** *in een app kan de gebruiker allerhande informatie op een overzichtelijke manier raadplegen*

**Foursquare/Locatie van vrienden:** *op een kaart ziet de gebruiker de locatie van vrienden*

**Slot:** *een automatisch slot op de fiets wanneer de gebruiker de fietstocht beëindigt (eventueel inclusief vingerafdrukbeveiliging)*

**Sightseeing:** *de gebruiker krijgt informatie over bezienswaardigheden nabij hem*

**Hotspot:** *de gebruiker wordt verwittigd wanneer er een internet hotspot nabij hem is*

Muziek/luidspreker

**Varia**

**Batterij:** *mogelijkheid om de batterij van de Raspberry Pi op te laden met een dynamo*

**Invloed rode lichten:** nagaan hoe lang bepaalde lichten gemiddeld op rood staan, en aan de hand hiervan de invloed op de reistijd berekenen

# Bibliografie

|  |  |
| --- | --- |
| [1] | „Wikipedia,” [Online]. Available: http://nl.wikipedia.org/wiki/Raspberry\_Pi. [Geopend 21 Oktober 2014]. |
| [2] | „Arduino (Computerplatform),” [Online]. Available: http://nl.wikipedia.org/wiki/Arduino\_(computerplatform). [Geopend 21 Oktober 2014]. |