

**PROBLEEMOPLOSSEN EN ONTWERPEN, DEEL 3**

***CWB2***

*Schrooten Bernd*

*Schryvers Andreas*

*Sels Shoera*

*Switsers Sander*

*Van den Berghe Pieter*

*Van Laere Nathan*

**Quantified Bike**

Eindverslag

Co-titularis

Professor Duval E.

Begeleider(s)

Charleer Sven

De Croon Robin

Klerkx Joris

Santon Jose Luis

ACADEMIEJAAR 2014-2015

Inhoudsopgave

[1. Groepsleden 2](#_Toc404958780)

[2. Brainstorm 4](#_Toc404958781)

[3. Productbeschrijving 4](#_Toc404958782)

[User story 1 5](#_Toc404958783)

[User story 2 5](#_Toc404958784)

[User story 3 5](#_Toc404958785)

[4. Use cases 5](#_Toc404958786)

[5. Architectuur 6](#_Toc404958787)

[Fysieke architectuur 6](#_Toc404958788)

[Softwarearchitectuur 7](#_Toc404958789)

[6. Klassendiagram 7](#_Toc404958790)

[7. Technologie: 7](#_Toc404958791)

[Raspberry Pi 7](#_Toc404958792)

[Arduino 8](#_Toc404958793)

[JavaScript 8](#_Toc404958794)

[jQuery 8](#_Toc404958795)

[JSON 9](#_Toc404958796)

[Visualisatie Bibliotheken 9](#_Toc404958797)

[8. Vakintegratie 9](#_Toc404958798)

[9. Besluit 9](#_Toc404958799)

[10. Appendixen 10](#_Toc404958800)

[Appendix: geleverde werk 10](#_Toc404958801)

[Appendix: Planning 10](#_Toc404958802)

[Appendix: Gedocumenteerde klasse 10](#_Toc404958803)

[Bronvermelding 10](#_Toc404958804)

# 1. 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

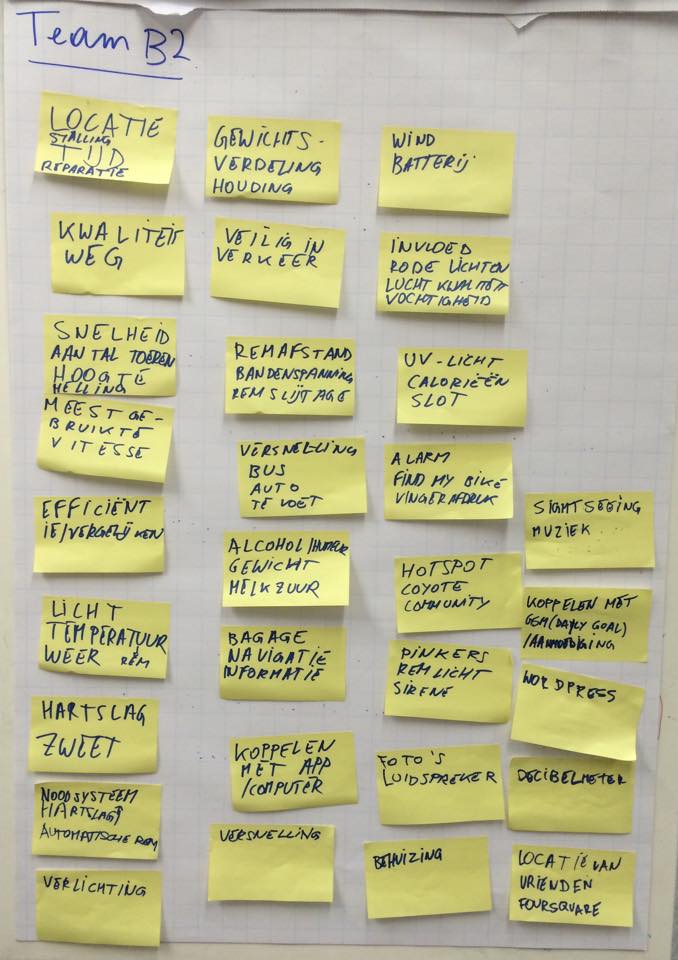
Van den Berghe Pieter 2 Elektrotechniek - Computerwetenschappen

Van Laere Nathan 2 Computerwetenschappen - Elektrotechniek

# 2. 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 lijst 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, temperatuur en foto’s van de omgeving. 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.

# 3. Productbeschrijving

Met onze applicatie is de gebruiker in staat om allerlei informatie over zijn fietstochten bij te houden en te raadplegen op een webapplicatie. Wat de applicatie onderscheidt van andere, is dat een fotologboek van de fietstochten wordt bijgehouden. Zo kan de gebruiken met een time-lapse elke fietstocht herbeleven en tonen aan vrienden en familie. In deze time-lapse krijgt de gebruiker ook extra informatie over zijn trip, zoals bv: de temperatuur, de snelheid en de hoogte gemeten op dezelfde momenten als de foto’s worden getrokken. Tenslotte krijgt de gebruiker tijdens de fietstocht feedback over zijn snelheid tegenover zijn gemiddelde snelheid. Wanneer hij trager fietst dan zijn gemiddelde snelheid brand er een rood lampje, wanneer hij sneller rijdt een groen lampje.

Om een duidelijk beeld te geven van de functionaliteiten van de applicatie worden er nu drie user stories beschreven.

## 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.

# 4. Use cases

* (In lijstvorm) Lijst elke use case, inclusief korte samenvatting in max. 2 zinnen, op. Deze lijst is een ideale en complete weergave van de functionaliteit van je applicatie; d.w.z. dat ook alle use cases die wegens tijdbeperkingen niet geimplementeerd werden hierin worden opgelijst (incl. korte samenvatting).
* (In lijstvorm) Benoem de drie belangrijkste use cases van je applicatie en **motiveer** grondig.
* Werk deze drie belangrijkste use case volledig uit. **De begeleiders zullen deze use case-beschrijvingen gebruiken om de implementatie van je applicatie te evalueren.**
* Minstens 1.5blz **per use case**.
* Maximaal 10blz.

# 5. Architectuur

In deze sectie geven we een overzicht van de fysieke en de softwarearchitectuur van de applicatie. We beginnen met het bespreken van de fysieke architectuur, dit is een overzicht van welke componenten op welke manier met elkaar verbonden interageren om de applicatie te laten werken.

## Fysieke architectuur

Voor dit project hebben we twee single-board computers ter beschikking gekregen, een Arduino Nano en een raspberry Pi. Voor een beschrijving van deze computers, zie sectie 7. De Arduino wordt aangesloten op de Raspberry Pi via een USB-kabel. Op de Arduino zelf worden de gps-sensor en de barometer aangesleten. De Arduino leest data van de sensors en stuurt deze door naar de Raspberry Pi. (PINNNEN NOG AANDUIDEN). Op de raspberry Pi zitten een druksensor, twee driekleurige LED’jes en de cameramodule. Deze laatste staat op de CSI-poort van de Raspberry Pi. De LED’jes zijn verbonden via drie digitale pinnen, die tevens voor stroomtoevoer zorgen. Elke pin vertegenwoordigt een ander kleurtje van de LED’jes. De drie mogelijke kleuren zijn blauw, rood en groen, maar de blauwe component van beide LED’jes zijn in deze opstelling niet in gebruik. De drukknop krijgt rechtstreeks van de Raspberry Pi de nodige stroom en is aangesloten aan de grond. De drukknop stuurt een signaal naar de Raspberry Pi door een verbinding via een digitale pin. De Raspberry Pi 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 en de verkregen data van de barometer en de gps-sensor naar de server, waarna deze kunnen weergegeven worden op de site. Het tweede LED’je werkt op een analoge manier als beschreven voor het andere LED’je. De Raspberry Pi vergelijkt de huidige snelheid met een gemiddelde snelheid, die wordt bijgehouden en na elke trip wordt geupdatet. Wanneer de huidige snelheid hoger ligt dan de gemiddelde wordt de LED groen, ander wordt hij rood.

Afbeelding X geeft een overzicht van de verbindingen tussen Raspberry Pi , de Arduino en de componenten.

AFBEELDING VAN VERBINDINGEN

## Softwarearchitectuur

Bespreek, afzonderlijk, de fysieke en softwarearchitectuur van je applicatie. Voor inspiratie zie [Domeinmodellen en klassendiagrammen](https://docs.google.com/viewer?a=v&pid=explorer&chrome=true&srcid=0B5LaWbG7FgXVNTY4NTI0ZjItMzY3My00Y2E5LTlkY2EtNGE4ODE1YzY2NzRm&hl=en_GB) slide 7 en 8.

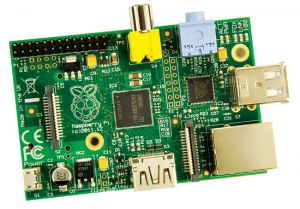
* Verklaar hoe je de gelaagde softwarearchitectuur concreet hebt gerealiseerd.
* Wat zijn de voordelen van de voorgestelde softwarearchitectuur?
* Maximaal 1blz.

# 6. Klassendiagram

* (In appendix) Overzichtelijke en leesbare grafische weergave van het klassendiagram (het volgen van de UML standaard wordt aanbevolen), waarbij de klassen in vier duidelijk te onderscheiden lagen worden opgedeeld. Deze lagen zijn: Interfacelaag, Controllerlaag, Modellaag en Persistentielaag.
* (In lijstvorm) Beschrijf alle klassen, hun verantwoordelijkheden en onderlinge relaties.
* Maximaal 5blz (zonder appendix).

# 7. Technologie:

## Raspberry Pi

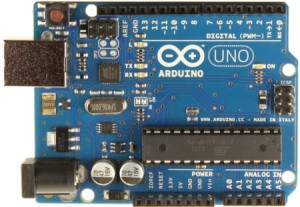


Figuur 1: Raspberry Pi

De Raspberry Pi is een goedkope singleboardcomputer. Het heeft de grootte van een kredietkaart, maar kan enorm makkelijk tot een computer omgebouwd worden, gewoon een scherm, muis en toetsenbord inpluggen. Binnen ons project is dit het voornaamste voordeel. Zijn grootte maakt het mogelijk om een computer te laten draaien tijdens het fietsen, die de fietser niet stoort. De Raspberry Pi heeft vele aansluitmogelijkheden: niet alleen de “basis” aansluitingen zoals USB, HDMI en Ethernet zijn aanwezig, men kan ook een camera en enorm veel andere sensors 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 Raspberr y Pi ook enkele nadelen: het gebruik vraagt wat kennis van Linux, de rekenkracht is niet enorm en er is geen ingebouwde Wi-Fi.

Tijdens het project hebben we toch verschillende keren problemen gehad met de Pi. Doordat we zoveel foto’s trekken, en dus opslaan op de Raspberry Pi tijden het fietsen, is er enorm veel geheugen nodig. De Raspberry Pi heeft hier nier genoeg geheugen voor. Om dit probleem op te lossen comprimeren we de foto’s direct wanneer ze getrokken worden.

## Arduino

Arduino is een opensource-computerplatform. 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. .

Figuur 2: Arduino

De problemen die wij hebben ondervonden tijdens het werken met de Arduino bestond er vooral in dat er interferentie bestaat tussen de data van de verschillende sensors. Eerst was onze drukknop aangesloten op de Arduino, samen met de gps-sensor. Dit hebben we veranderd, omdat de data dan niet meer op een efficiënt manier geïnterpreteerd kan worden. Tussen de barometer en de gps-sensor bestaat er ook interferentie, maar we zijn erin geslaagd

## 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.

PROBLEMEN DIE TEGENGEKOMEN ZIJN …

## 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.

PROBLEMEN DIE TEGENGEKOMEN ZIJN

## 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.

PROBLEMEN DIE TEGENGEKOMEN ZIJN

## Visualisatie Bibliotheken

Als visualisatie bibliotheek gebruiken we Google 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 makkelijke te bewerken zijn. 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.

UITGEBREIDER, PROBLEMEN DIE TEGENGEKOMEN ZIJN

MOET NOG HALF PAPIER MINDER ZIJN

# 8. Vakintegratie

In eerste instantie sluit het project 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.

# 9. Besluit

* Licht minstens de volgende items toe:
  + Wat heb je geleerd tijdens dit project?
  + Welke ontwerpfasen vond je onmisbaar?
  + Welke technieken ga je later nog aanwenden?
  + Wat vond je van de indeling van het project in ontwerpfasen (technische introductie, brainstorm, user stories, use cases, klassendiagram)?
  + Vind je een tussentijdse demo nuttig?
  + Was er te veel of te weinig focus op ontwerpen, documenteren, of implementeren?
  + Wat vond je van de opdracht? Was de keuze te ruim of te gefocussed?
  + Wat vond je van het gebruik van de pi?
  + Indien je zelf het P&O3-project zou mogen inrichten, wat zou je veranderen?
* Minstens 1blz.

# 10. Appendixen

## Appendix: geleverde werk

* (In tabelvorm) Per teamlid (kolom) een nauwkeurige indicatie hoe lang (in uren) er aan elke fase (rij) (technische introductie, brainstorm, user stories, use cases, klassendiagram, implementatie) werd besteed.
* Geef ook de totalen aan per teamlid (extra kolom) en per fase (extra rij).

## Appendix: Planning

* Licht de taakverdeling toe door middel van een gedetailleerde Gantt-chart.
* Geef duidelijk de twee iteraties aan.

## Appendix: Gedocumenteerde klasse

* Een volledig gedocumenteerde klasse bestaande uit minstens 3 niet-triviale methodes. Gebruik (bijvoorbeeld) Eclipse om deze klasse af te drukken. Als printer kies je een PDF-documentprinter, dewelke jou code **met syntax highlighting** exporteert naar een PDF-bestand.

## Bronvermelding

<http://codeduino.com/tutorials/arduino-vs-raspberry-pi/>

<https://sites.google.com/site/bus237raspberrypi/customize>