**PLAN VAN AANPAK**  Quinten Peels   
1667740

INHOUDSOPAVE

[INLEIDING 3](#_Toc450312618)

[DOELSTELLING EN OPDRACHT 4](#_Toc450312619)

[PROJECTORGANISATIE 9](#_Toc450312620)

[PROJECTSTRUCTUUR 9](#_Toc450312621)

[AANPAK 10](#_Toc450312622)

[PROGRAMMEER/MARKUP TALEN 10](#_Toc450312623)

[SOFTWARE 11](#_Toc450312624)

[FASERING 11](#_Toc450312625)

[PLANNING 12](#_Toc450312626)

[RISICO’S 13](#_Toc450312627)

# INLEIDING

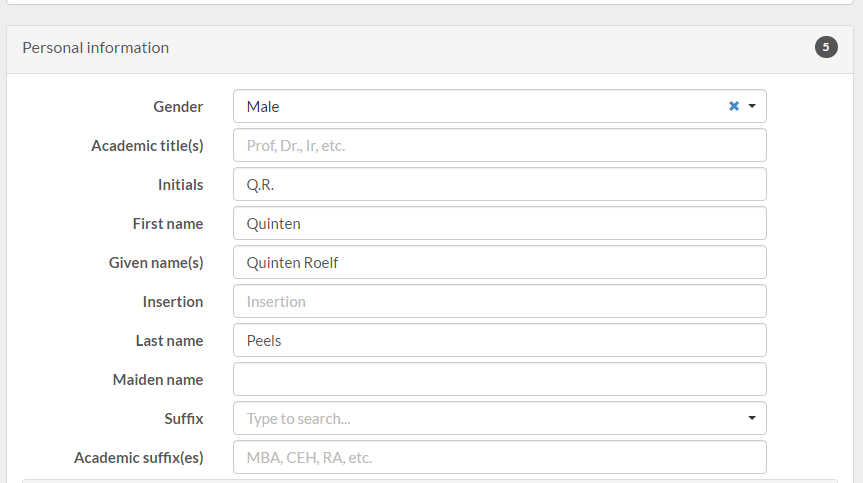
Er is een groot gegevens probleem in de wereld.

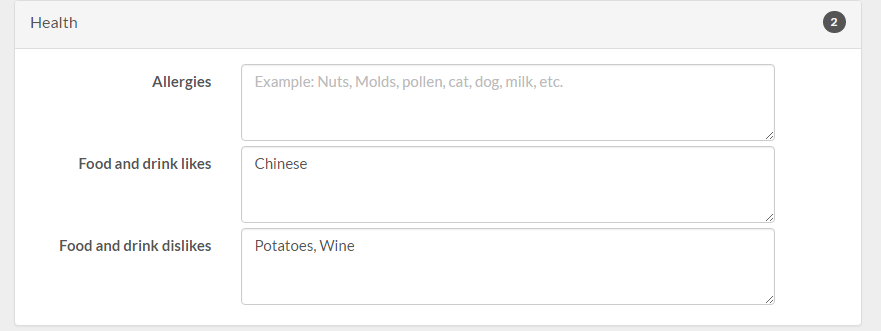
Eigenaren van gegevens hebben geen controle over waar hun gegeven blijven en wat er mee gebeurd en het delen van gegevens is een ware hel. Portals, formulieren, mutaties doorgeven via sms, whatsapp aan je vrienden etc.

Aan de andere willen gegevens gebruikers (consument) het liefst altijd up2date gegevens hebben. Niet alleen, up2date, maar je wilt ook dat aangeleverde gegevens ‘waar zijn’. Is de persoon wel wie die zegt dat hij is. Woont hij wel waar hij woont? Je wilt het liefst alle gegevens van al je familieleden, vrienden en kennissen zonder dat je daar iets voor hoeft te doen, aangeleverd krijgen.

# DOELSTELLING EN OPDRACHT

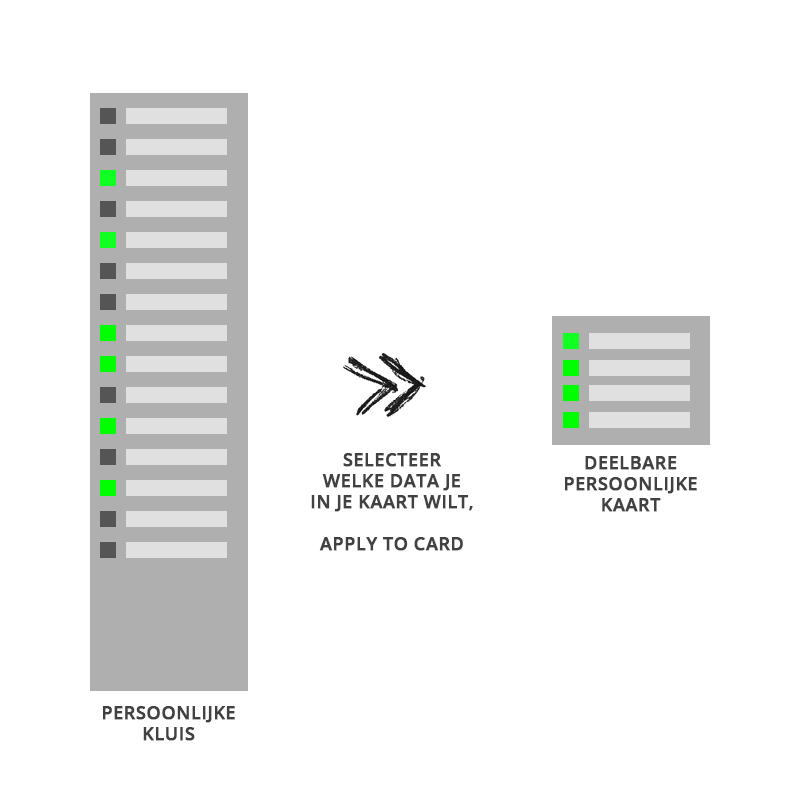
Mensen zullen gegeven kunnen opslaan en onderhouden in mijn webapplicatie. Dit gebeurt in een persoonlijke digitale kluis waar alleen de eigenaar toegang tot heeft. In deze kluis bevinden zich allemaal soorten ‘datasets’.

  
*(Hier een voorbeeld van mijn kluis onder het onderwerp ‘personal information’, hierin bevinden zich allemaal soorten datasets wat betrekking heft tot de naam.) Bron foto: OnlyOnce.com*



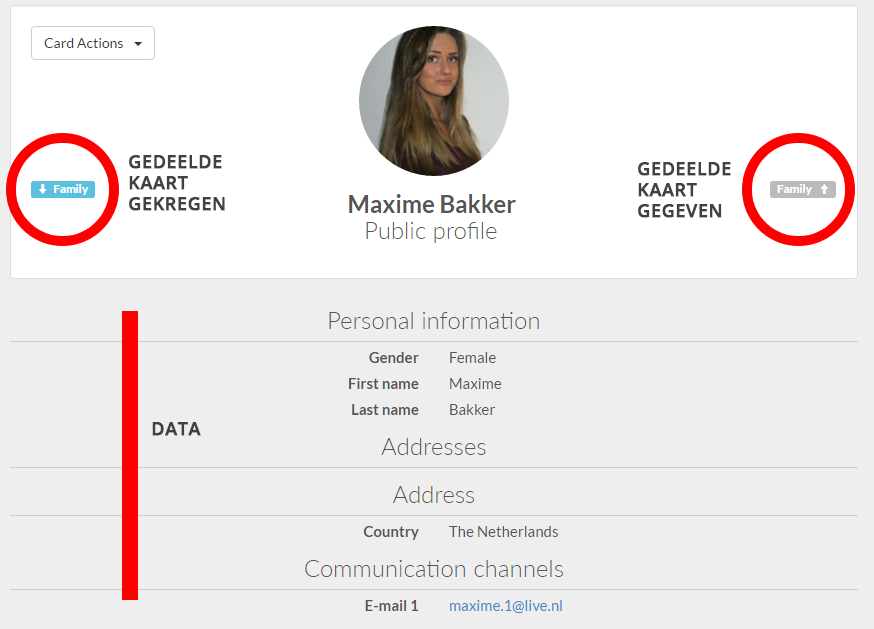
*Bron foto: OnlyOnce.com*

Je vult deze datasets in, maakt een kaartje (benaming), selecteer welke ‘datasets’ je op dit kaartje wilt en vervolgens kan je dit kaartje delen.



Een voorbeeld hiervan is het uitwisselen van een digital visitekaartje.In dit geval heet mijn deelbare kaart, ‘visitekaartje’. Op mijn visitekaartje wil ik mijn naam, nummer, beroep, adres. Ik selecteer deze datasets van mijn kluis en apply deze naar een deelbare kaart.

Wanneer een gebruiker zijn persoonlijke gegevens aanpast in de digitale kluis wordt deze data/gegevens automatisch ge-update op de kaartjes waar de data op staat.



*(Hier een voorbeeld van zoon gekregen kaart van Maxime Bakker. Het gaat hier om een gekregen kaart ‘Family’ met de volgende data inhoud) Bron foto: OnlyOnce.com*

Het idee van de applicatie is dat je maar 1x je gegevens invult, en daarna blijft onderhouden. Dit word teen centrale plaats voor je data. Nooit meer handmatig overal je data te hoeven aanpassen of handmatig dingen te blijven invullen. Denk hierbij aan het aanmelden bij een sportschool, ze willen je voornaam, achternaam, woonplaats, etc. Dit kan voortaan allemaal via de applicatie omdat je toegang kan geven tot bepaalde informatie.



*Context diagram*



*Use case diagram*

**Samenvatting:**Met de applicatie kan je eenvoudig data uitwisselen/ontvangen en onderhouden. Nooit meer keer op keer zelf dezelfde informatie invullen. Verander het 1x in de applicatie en iedereen (Waarmee jij de data hebt gedeeld) heeft gelijk jouw nieuwe data binnen.

# PROJECTORGANISATIE

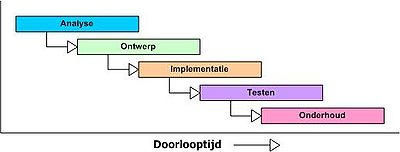
## PROJECTSTRUCTUUR

|  |  |
| --- | --- |
| PERSOON | PROBLEEM |
| Quinten (Gebruiker) | Wilt niet eindeloos lang formulieren blijven invullen. Wilt gemakkelijk, snel en veilig data kunnen uitwisselen Daarnaast wil ik een overzicht van alle mensen zie inzicht hebben over mijn data. |
| Remco (Bedrijf) | Wilt accuraat en up2date informatie van klanten. Daarnaast wilt remco natuurlijk ook aan klanten informatie over het bedrijf geven (Naam, adres, telefoonnr). |

# AANPAK

## ONTWIKKELMETHODIEK

**GEKOZEN ONTWIKKELMETHODIEK:** WATERVAL



Bij de waterval methodiek wordt het process van boven naar beneden doorlopen. Deze methodiek werkt uitstekend bij mij aangezien ik al ervaring heb in webdevelopment. Je analyseert, maakt een scope en gaat in die scope vervolgens verder.

Het voordeel van de waterval ontwikkelmethodiek is dat het een simpel en ‘straight forward’ model is. Voor elk process heb je een deadline, anders kan je niet met de volgende beginnen. Dit was voor mij ook een goed argument om deze ontwikkel methodiek toe te passen voor mijn ipas. Daarnaast is de waterval methodiek erg gepast voor kleinere projecten.

## PROGRAMMEER/MARKUP TALEN

|  |  |
| --- | --- |
| TAAL | FUNCTIE |
| Java | Backend taal, gaat gebruikt worden voor bijvoorbeeld het inloggen, registreren van de gebruiker alsmede het delen van de kaartjes. Java zal primair gebruikt worden met het uitvoeren van SQL queries en het opvangen van dit soort data. |
| SQL | Om data uit de database te halen en op te slaan |
| HTML | Front-end markup (Input forms, buttons, etc) |
| CSS | Styling van de webpagina |
| J2EE | Servlets en JSP’s |

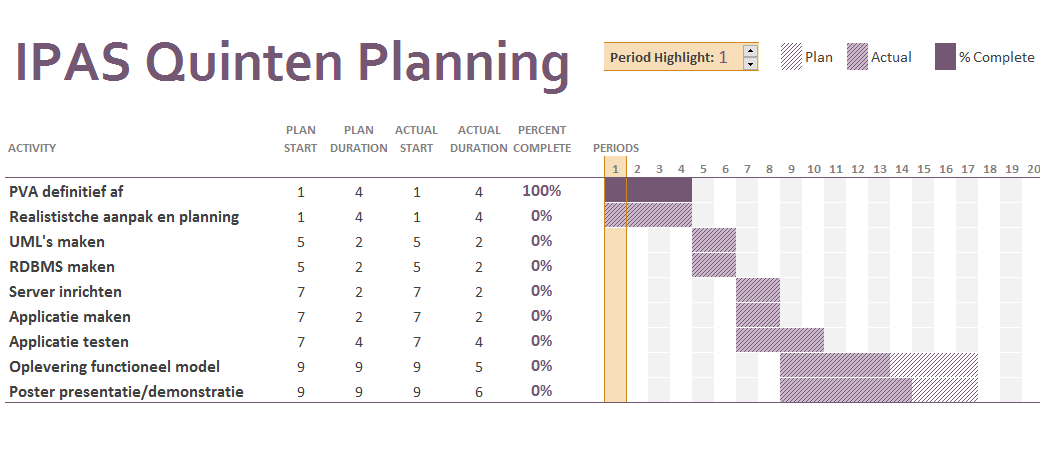
## SOFTWARE

|  |  |
| --- | --- |
| SOFTWARE | FUNCTIE |
| nginx | Webserver (http) software voor op de server. |
| MySQL | Server database software |
| IntelliJ | Java IDE |
| Draw.io | Voor het uitwerken van UML’s/RDBM’s |
| HeidiSQL | SQL viewer/development |

## FASERING

|  |  |
| --- | --- |
| TE DOEN | DEADLINE |
| Defnitieve PVA maken | 30 april |
| Realistische aanpak en planning af | 9 mei |
| UML maken | 12 mei |
| Relationele database ontwerpen | 12/13 mei |
| Applicatie maken | 14 mei |
| Bugs testen | 14 mei ~ 1 juni |
| Oplevering functioneel ontwerp | 1 juni |
| Poster presentative/demonstratie | 16/17 juni |

# PLANNING



# RISICO’S

|  |  |
| --- | --- |
| RISICO | OPLOSSING |
| Ziekte | Op tijd beginnen met punten en hiervoor ook deadlines voor opgeven |
| Webserver software setup/config problemen  **(Te weinig Linux kennis)** | Forums zo als LowEndTalk kunnen mij hierbij helpen, ook kan ik hulp krijgen in IRC. |
| Met het kaartje niet automatisch formulieren kunnen invullen  **(Te weinig Java kennis)** | De kaartjes kunnen exporten naar CSV of ander soort type en hierop weer een koppeling kunnen bouwen.  Dus niet “kaartje -> formulier” maar “kaartje -> CSV -> formulier” |
| Het kaartje technisch niet aan account kunnen koppelen **(Te weinig Java kennis)** | Het kaartje een dynamische pagina kunnen geven en vervolgens op de pagina access geven ipv aan het kaartje zelf. |
| Het niet voor elkaar krijgen om allemaal verschillende soorten data op de kaart te kunnen opslaan (Persoonlijke gegevens, medische gegevens, school gegevens)  **(Te weinig Java/SQL kennis)** | Ik wil er dan voor kiezen om alleen voor persoonlijke informatie te gaan, denk hierbij aan NAW gegevens. |