**FUNCTIONEEL ONTWERP**  Quinten Peels   
1667740

INHOUDSOPAVE

[Hoofdstuk 1: Inleiding 3](#_Toc452370749)

[Hoofdstuk 2: Overzicht 4](#_Toc452370750)

[Context diagram 7](#_Toc452370751)

[Hoofdstuk 3: Uses cases 8](#_Toc452370752)

[Use case diagram 8](#_Toc452370753)

[Actor templates 9](#_Toc452370754)

[Actor templates 12](#_Toc452370755)

[Hoofdstuk 4: Datamodel 13](#_Toc452370756)

# Hoofdstuk 1: Inleiding

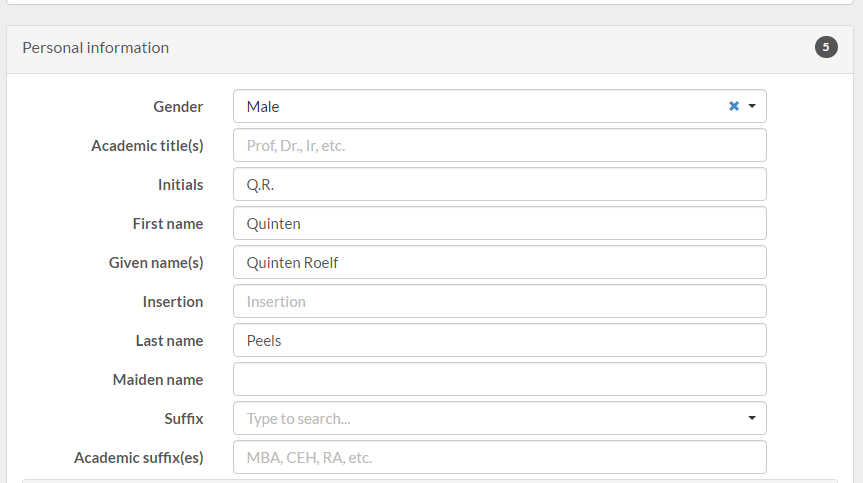
Er is een groot gegevens probleem in de wereld.

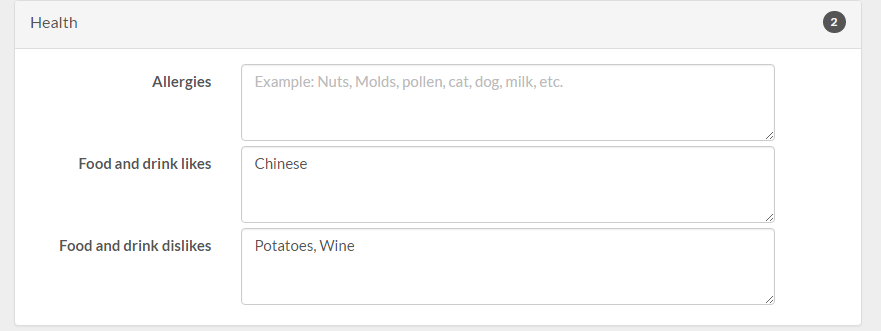
Eigenaren van gegevens hebben geen controle over waar hun gegeven blijven en wat er mee gebeurd en het delen van gegevens is een ware hel. Portals, formulieren, mutaties doorgeven via sms, whatsapp aan je vrienden etc.

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.

# Hoofdstuk 2: Overzicht

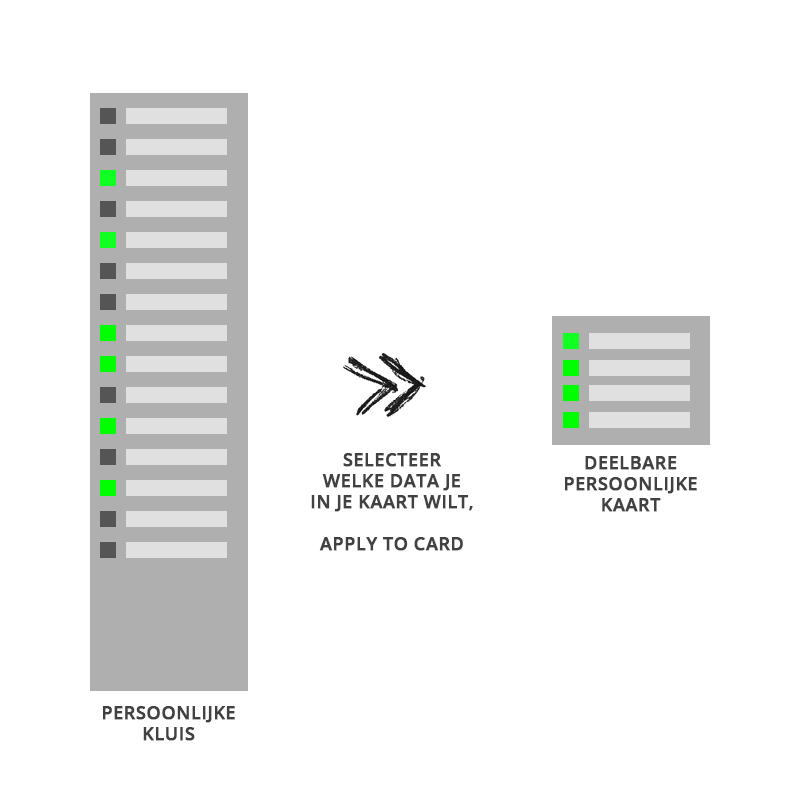
Mensen zullen gegeven kunnen opslaan en onderhouden in mijn webapplicatie. Dit gebeurt in een persoonlijke digitale kluis waar alleen de eigenaar toegang tot heeft. In deze kluis bevinden zich allemaal soorten ‘datasets’.

  
*(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*



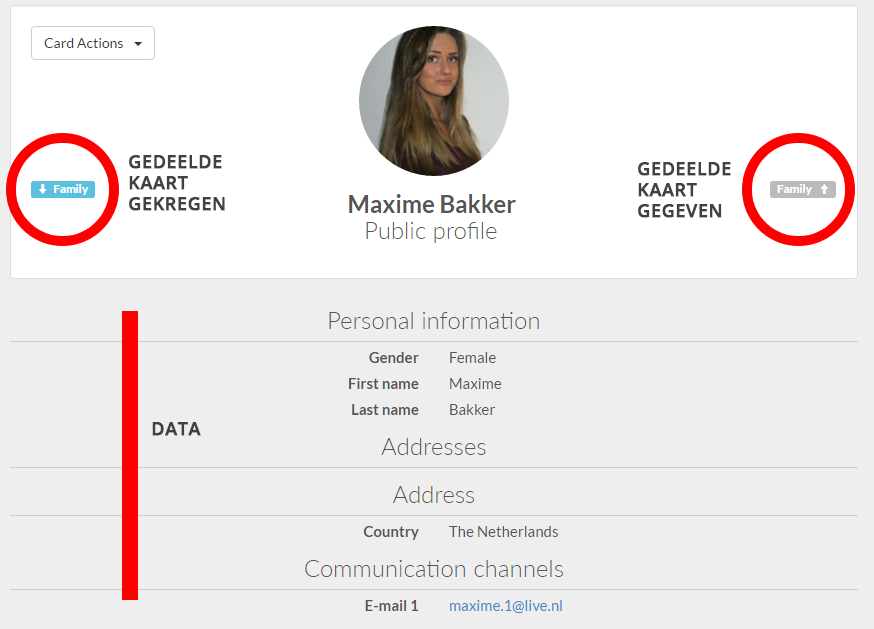
*Bron foto: OnlyOnce.com*

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



# Hoofdstuk 3: Uses cases

## Use case diagram



**Toelichting:**Met de applicatie kan je eenvoudig data uitwisselen/ontvangen en onderhouden. Bedrijven kunnen formulieren verzenden waarop de gebruiker makkelijk de aangevraagde data kan beantwoorden met wat in zijn of haar kluis staat.

## Actor templates

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_1 |
| Use Case naam | Updaten van de card |
| Actor | Gebruiker |
| Samenvatting | Actor kan informatie toepassen op een card indien zijn/haar kluis gevuld is met informatie. |
| Pre Conditie | Gebruiker heeft een kluis met genoeg ingevulde informatie om een card te maken. |
|  | |
| Main scenario |  |
|  | 1. Actor bedenkt welke data hij wilt toepassen op de desbetreffende card. Persoonlijke informatie voor een persoonlijke card, business voor een business card. 2. Systeem geeft een overzicht van alle data die de Actor in de klus heeft staan 3. Actor kiest de fields die hij of zij wilt toepassen op een card en slaat dit op. 4. Systeem voegt een activity log toe in de activity feed van de actor. |
| Post-Conditie(Main Scenario) | 1. Systeem linkt de data aan de card. |
| Alternatief Secnario (A1) | |  |  | | --- | --- | | Trigger | Actor heeft geen data in de klus | | Stappen | 1. Systeem kan geen data verwerken 2. Actor krijgt een lege card terug | |
| Post conditie (A1) | Systeem is ongewijzigd. |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_2 |
| Use Case naam | Bekijken van data op een card |
| Actor | Gebruiker |
| Samenvatting | Actor kan de data van een gekregen card bekijken |
| Pre Conditie | Actor heeft een card ontvangen van een andere gebruiker |
|  | |
| Main scenario |  |
|  | 1. Actor ziet een lijst met gebruikers met daarbij de ontvangen card. 2. Actor kiest welke card hij wilt zien 3. Systeem weergeeft de card |
| Post-Conditie(Main Scenario) | 1. Actor ziet een overzicht van alle data die op de debetreffende card staan |
| Alternatief Secnario (A1) | Nvt. |
| Post conditie (A1) | - |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_3 |
| Use Case naam | Verzenden van een card |
| Actor | Gebruiker |
| Samenvatting | De gebruiker kan een card kiezen en deze delen met een andere gebruiker op het system. |
| Pre Conditie | Er moet al data op de card staan (Zie UC\_1) |
|  | |
| Main scenario |  |
|  | 1. Actor kiest de card die hij of zij wilt delen 2. Actor kiest de gebruiker waar hij of zij de card wilt naar sturen. De actor vult hierbij de ontvangers ID in. 3. Systeem stuurt de card naar de ontvanger 4. Systeem voegt een activity log toe in de activity feed van de actor en ontvanger. |
| Post-Conditie(Main Scenario) | 1. De gebruiker heeft de card gekregen. |
| Alternatief Secnario (A1) | |  |  | | --- | --- | | Trigger | Er is geen member met de opgegeven ID | | Stappen | 1. Actor vult een incorrecte ID in 2. Systeem controleert of er een gebruiker is met het opgegeven ID 3. Systeem weergeeft een error | |
| Post conditie (A1) | Actor kan de card niet delen met de opgegeven gebruiker |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_4 |
| Use Case naam | Invullen/updaten van de datastore |
| Actor | Gebruiker |
| Samenvatting | Een gebruiker kan al zijn of haar informatie opslaan in de datastore (Een kluis met informatie). |
| Pre Conditie | Geen |
|  | |
| Main scenario |  |
|  | 1. Actor vult de data in bij het debetreffende veld. 2. Systeem slaat data op in de datastore 3. Systeem update cards waar de informatie van de datastore op staat. 4. Systeem maakt een activity feed aan dat de Actor zijn of haar datastore is ge-update. |
| Post-Conditie(Main Scenario) | 1. Actor’s informatie is geupdate. |
| Alternatief Secnario (A1) | |  |  | | --- | --- | | Trigger | Gebruiker wilt informatie verwijderen | | Stappen | 1. Systeem laat huidige data zien in de velden 2. Actor verwijderd de data van het veld 3. Actor slaat de data op 4. Systeem overschrijft de oude data met de nieuwe data | |
| Post conditie (A1) | Data is verwijderd uit Actor’s kluis. |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_5 |
| Use Case naam | Registreren |
| Actor | Bezoeker |
| Samenvatting | Een bezoeker kan een account aanmaken door het opgeven van een username en password. |
| Pre Conditie | Geen |
|  | |
| Main scenario |  |
|  | 1. Actor geeft een gebruikers naam en wachtwoord op. 2. Systeem checkt of de gebruikersnaam en password nog niet geregistreerd zijn 3. Systeem registreerd ingevulde gegevens 4. Systeem redirect gebruiker naar de login pagina met een boodschap dat het account is aangemaakt |
| Post-Conditie(Main Scenario) | 1. Actor kan nu inloggen |
| Alternatief Secnario (A1) | |  |  | | --- | --- | | Trigger | Het account bestaat al | | Stappen | 1. Actor geeft gebruikersnaam en wachtwoord op. 2. Systeem doet een check. 3. Systeem ziet dat er al een account bestaat. 4. Systeem weergeeft error aan gebruiker. | |
| Post conditie (A1) | Actor heeft niet succesvol een account aangemaakt. |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_6 |
| Use Case naam | Inloggen |
| Actor | Bezoeker |
| Samenvatting | Een bezoeker kan een inloggen om het platform te kunnen gebruiken |
| Pre Conditie | Geen |
|  | |
| Main scenario |  |
|  | 1. Actor vult een gebruikersnaam en wachtwoord in 2. Systeem controleert of er een match is op de ingevulde data. 3. Systeem geeft een sessie mee aan de Actor die bij het desbetreffende account hoort. |
| Post-Conditie(Main Scenario) | 1. Actor is nu ingelogd |
| Alternatief Secnario (A1) | |  |  | | --- | --- | | Trigger | Het account bestaat niet | | Stappen | 1. Actor vult een gebruikersnaam en wachtwoord in. 2. Systeem controleert of er een match is op de ingevulde data 3. Systeem geeft een error terug dat er een onjuiste gebruikersnaam/wachtwoord combinatie is. | |
| Post conditie (A1) | Actor heeft niet succesvol kunnen inloggen. |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_7 |
| Use Case naam | Shared card intrekken (Unsharen) |
| Actor | Gebruiker |
| Samenvatting | Een gebruiker kan access revoken / Shared card intrekken |
| Pre Conditie | Geen |
|  | |
| Main scenario |  |
|  | 1. Actor kiest de card van welke user hij wilt intrekken. 2. Actor verwijderd de card. 3. Systeem weergeeft een bericht 4. Systeem weergeeft een ge-update lijst |
| Post-Conditie(Main Scenario) | 1. Card is ingetrokken |
| Alternatief Secnario (A1) | N.v.t |
| Post conditie (A1) | N.v.t |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_8 |
| Use Case naam | Updaten van je profile picture / avatar |
| Actor | Gebruiker |
| Samenvatting | Een gebruiker kan nieuwe avatar linken. |
| Pre Conditie | Geen |
|  | |
| Main scenario | Actor heeft een afbeelding geupload op een site. |
|  | 1. Actor vult de link in de profile field in. 2. Systeem verwerkt de link 3. Systeem weergeeft een bericht |
| Post-Conditie(Main Scenario) | 1. Avatar is ge-update. |
| Alternatief Secnario (A1) | N.v.t |
| Post conditie (A1) | N.v.t |

|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_9 |
| Use Case naam | Updaten van de Actor’s wachtwoord |
| Actor | Gebruiker |
| Samenvatting | Een gebruiker kan het wachtwoord veranderen |
| Pre Conditie | Geen |
|  | |
| Main scenario | Geen |
|  | 1. Actor vult huidige en nieuwe wachtwoord in 2. Systeem doet een check of het huidige wachtwoord klopt. 3. Systeem weergeeft een bericht |
| Post-Conditie(Main Scenario) | 1. Wachtwoord is veranderd |
| Alternatief Secnario (A1) | 1. Actor vult verkeerd huidig wachtwoord en nieuwe wachtwoord in 2. Systeem doet een check of het huidige wachtwoord klopt. 3. Systeem weergeeft een bericht |
| Post conditie (A1) | 1. Wachtwoord is niet veranderd |

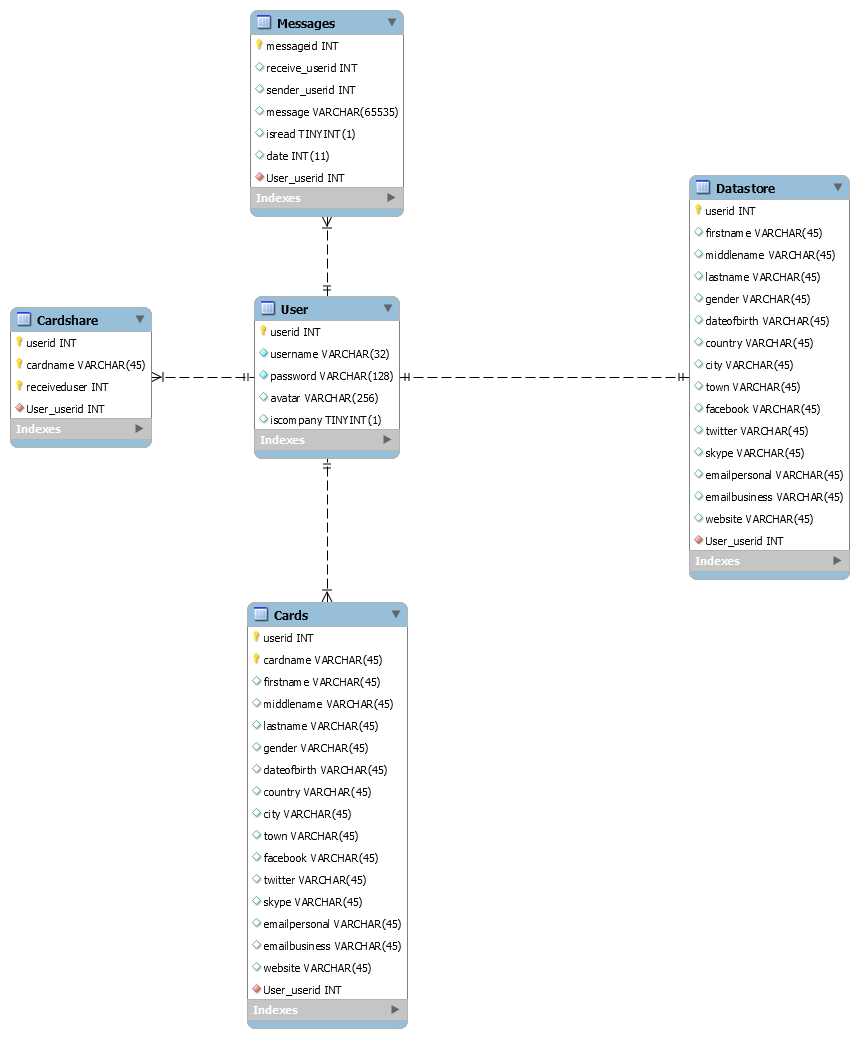
|  |  |
| --- | --- |
| Overview | |
| Use Case nummer | UC\_10 |
| Use Case naam | Updaten van de Actor’s gebruikersnaam |
| Actor | Gebruiker |
| Samenvatting | Een gebruiker kan de gebruikersnaam veranderen |
| Pre Conditie | Geen |
|  | |
| Main scenario | Geen |
|  | 1. Actor vult nieuwe gebruikersnaam in 2. Systeem update gebruikersnaam |
| Post-Conditie(Main Scenario) | 1. Gebruikersnaam veranderd |
| Alternatief Secnario (A1) | N.v.t |
| Post conditie (A1) | N.v.t |

## Actor templates

|  |  |
| --- | --- |
| **Rolnaam** | Gebruiker |
| **Functienaam** | Een gebruiker van het platform |
| **Omschrijving van de rol** | Kan datastore invullen, updaten. Vervolgens hiermee cards sharen en forms invullen. |
| **Uitoefenaar** | Quinten Peels |

|  |  |
| --- | --- |
| **Rolnaam** | Bezoeker |
| **Functienaam** | Een bezoeker / non geregistreerde gebruiker |
| **Omschrijving van de rol** | Kan de front-end kant van de website zien. Geeft kortweer wat de applicatie inhoud. Kan daarna kiezen om in te loggen of te registreren. |
| **Uitoefenaar** | Remco Ruijsenaars |

# Hoofdstuk 4: Datamodel



User(ID int, isCompany int, Username string, Password String)  
Messages(MessageID int, msg\_content string, read int, msg\_to int, date int, msg\_from int)  
Card(UserID int, CardName String, CardID int, Card-data string, CardSharedWith int)  
Datastore(UserID int, DateofBirth date, placeofbirth String, email\_personal string, email\_business string, website\_url string, facebook\_url string, twitter\_url string, Skype\_url string, gender string, lastname string, middlename string, firstname string)