

SoftwareOntwikkeling Project: Glass

Behoeftenanalyse en Architectuur

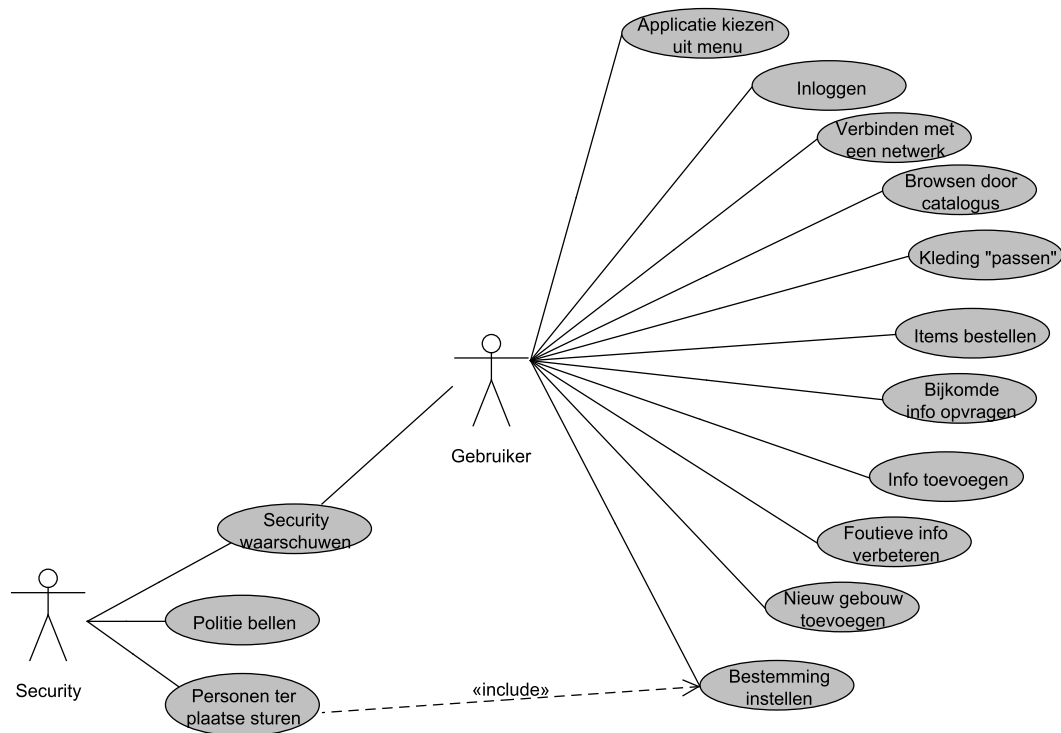
Academiejaar 2012-2013
Groepnummer 16

Groepsleden:

- Caroline De Brouwer
- Eveline Hoogstoel
- Stefaan Vermassen
- Titouan Vervack

Behoeftenanalyse

1 UML-use case diagram



Figuur 1: Use case diagram

2 Use-case beschrijving

A Use-case 1

Use Case: De security waarschuwen

Main Success Scenario: De security wordt op de hoogte gebracht

1. De bril maakt een foto.
2. De bril bepaalt de huidige locatie voor het attribuut locatie in Foto.
3. De bril maakt verbinding met de brillen van de security.
4. De bril verzend een push-melding met de Foto en de Locatie.
5. Scenario beëindigd, postconditie geldt vanaf nu.

Extensions

<MSS STEP 2> [De locatie van de gebruiker is onbekend]

2.1: Zet locatie op onbekend.

2.2: Ga naar MSS stap 3.

<MSS STEP 3> [Er kan geen verbinding worden gemaakt met de security]

3.1: In plaats van met de security maakt de bril verbinding met de politie.

3.2: Ga naar MSS stap 4.

Pre-conditions:

- De camera en de GPS-ontvanger zijn actief.

Post-conditions:

- De bril van de security krijgt een pushnotificatie van het incident.

Triggers:

- De gebruiker meldt een incident met het spraakcommando 'incident melden'.

B Use-case 2**Use Case: Webshop****Main Success Scenario:**

1. De bril ontvangt via bluetooth een afbeelding van een kledingstuk.
2. De camera zoekt naar het silhouet van een persoon.
3. De afbeelding wordt geprojecteerd op de bril.
4. De camera maakt foto's en analyseert de beelden.
5. Aan de hand van de foto's berekent de bril de afmetingen van de gebruiker.
6. Met de spraakcommando's 'next' en 'previous' kan de gebruiker het vorige of het volgende kledingstuk passen uit zijn/haar verlanglijstje.
7. Met het commando 'buy' stuurt de bril een signaal naar de pc om het huidige kledingstuk in het winkelmandje te plaatsen in de juiste maat.

8. Scenario beëindigd, postconditie geldt vanaf nu.

Extensions

<MSS STEP 2> [De camera herkent geen silhouet (na een bepaalde tijd)]

2.1: Er verschijnt een boodschap "Ga voor de spiegel staan".

2.2: Met het commando "search again" begint de camera opnieuw te zoeken.

<MSS STEP 7> [De bluetooth verbinding is verbroken]

7.1: Er verschijnt een foutboodschap.

Pre-conditions:

- Bluetooth is ingeschakeld en er is een connectie met een pc.

Post-conditions:

- De gebruiker heeft een aantal kledingstukken gepast en eventueel besteld.

Triggers:

- De gebruiker stuurt via de pc een commando naar de bril.

C Use-case 3

Use Case: Geavanceerde navigatie (Indoor GPS)

Main Success Scenario: Bestemming instellen

1. De gebruiker stelt een bestemming in.
2. De route wordt door de backend berekend.
3. De kaart met route wordt op de bril geprojecteerd.
4. De gebruiker loopt door het gebouw.
5. De bril stuurt foto's door om de huidige locatie te updaten.
6. De gebruiker loopt tot als hij zijn bestemming bereikt heeft.
7. Scenario beëindigd, postconditie geldt vanaf nu.

Extensions

<MSS STEP 1> [De bestemming bestaat niet of wordt niet herkend]

1.1: Geef een foutmelding en laat de gebruiker opnieuw proberen.

<MSS STEP 2> [Er is geen verbinding met het systeem]

2.1: Toon een foutmelding.

2.2: Probeer opnieuw als de gebruiker aangeeft dat er verbinding met het internet is.

<MSS STEP 4> [De gebruiker loopt verkeerd.]

4.1: Herbereken de route.

4.2: Ga naar MSS stap 3.

Pre-conditions:

- De GPS functie en camera zijn actief.
- De bril heeft verbinding met het backendsysteem.
- De backend weet waar de gebruiker zich bevindt.

Post-conditions:

- De gebruiker heeft zijn bestemming bereikt.

Triggers:

- De gebruiker drukt op 'Bestemming instellen'.

D Use-case 4**Use Case: Augmented Reality overlay****Main Success Scenario: Object krijgt een overlay**

1. De bril scant de omgeving naar objecten.
2. De bril stuurt gevonden objecten naar het backendsysteem.
3. Het object wordt herkend door het backendsysteem.
4. De overlay-informatie wordt ontvangen in de bril.
5. De bril geeft de extra informatie over het object weer in een overlay.
6. De bril geeft een optie om de huidige informatie te verbeteren of nieuwe informatie toe te voegen.
7. Scenario beëindigd, postconditie geldt vanaf nu.

Extensions

<MSS STEP 2,4> [Er is geen verbinding meer met het systeem]

2,4.1: Toon een foutmelding.

2,4.2: Probeer automatisch of op vraag van de gebruiker opnieuw te connecteren met het internet.

<MSS STEP 3> [Het object is niet opgenomen in de database]

3.1: Toon een waarschuwing.

3.2: Geef de gebruiker de mogelijkheid om dit object toe te voegen.

<MSS STEP 5> [Er is geen extra informatie beschikbaar]

5.1: Toon een waarschuwing.

Pre-conditions:

- De camera is actief.
- De bril heeft verbinding met het backendsysteem.

Post-conditions:

- Alle objecten hebben een correcte overlay.

Triggers:

- De AR-applicatie wordt opgestart.

I Architectuur: klassendiagram

1 Projectbeschrijving

Een eerste functie van de bril is het **binnenshuis navigeren**, waardoor de route als overlay op de bril wordt **geprojecteerd**. Omdat er geen GPS-ontvangst mogelijk is, wordt er op regelmatige tijdstippen beelden van de bril doorgestuurd. Door vergelijking van opgeslagen beelden kan de locatie van de gebruiker bepaald worden. De bril kan ook in leermodus geschakeld worden, waarbij de videobeelden van de bril doorgestuurd worden, en de gebruiker kan aangeven waar hij zich bevindt.

Een tweede mogelijkheid van de bril bestaat erin dat de beelden van de bril constant **doorgestuurd** en geanalyseerd worden door een back-end systeem om **voorwerpen**

en personen te herkennen. Hierbij wordt gebruik gemaakt van een gegevensdatabank. De meta-data van herkende voorwerpen of personen wordt opgehaald uit de databank en naar de gebruiker doorgestuurd. De gebruiker kan desgewenst bijkomende informatie opvragen. Ook deze applicatie beschikt over een leermodus, zodat gebruikers meta-informatie kunnen toevoegen of bewerken in de databank.

Een derde mogelijkheid heeft te maken met online shopping. Wanneer de gebruiker doorheen de webwinkel navigeert en een kledingstuk aanklikt, kan hij zichzelf zien met deze kleren aan. Het kledingstuk wordt als overlay **geprojecteerd** op je bril zodat het lijkt alsof je de kleren echt aan hebt. Terwijl de gebruiker voor de spiegel staat meet de bril de maten op en vergelijkt die met de matentabel, zo weet je ook onmiddellijk welke maat je zal nodig hebben.

Tenslotte helpt de bril mee aan de algemene veiligheid. Wanneer de gebruiker zich bevindt op een publieke plaats, wordt er automatisch verbinding gemaakt met de brillen van de security. Dit kan bijvoorbeeld een station, een winkel of een nachtclub zijn. Wanneer de gebruiker iets verdacht opmerkt, dit kan bijvoorbeeld een gevecht zijn, kan de gebruiker het incident rapporteren en wordt de security op de hoogte gebracht. Er wordt een foto getrokken en de locatie wordt bepaald. Dit kan met behulp van use case 1: **binnenshuis navigeren**. Zij worden doorgestuurd naar de security. Op die manier is er ook onmiddellijk bewijs tegen de daders.

De belangrijkste common core functies van onze usecases zijn dus projectie, het doorsturen van beelden en de Indoor GPS.

2 Klassenlijst

Klassenaam	Attributen/Operaties	Opmerkingen
Gebruiker	<ul style="list-style-type: none">• <code>private Account account</code>• <code>public void setAccount(Account account)</code>• <code>public Multimedia maakNieuweVideo()</code>• <code>public void verbindApparaat()</code>• <code>public void startNavigatie()</code>• <code>public void herkenObjecten()</code>	Kan m.b.v. zijn account geïdentificeerd worden.
Account	<ul style="list-style-type: none">• <code>private String naam</code>• <code>private String voornaam</code>• <code>private String email</code>• <code>public String getNaam()</code>• <code>public String getVoornaam()</code>• <code>public String getEmail()</code>• <code>public void setNaam(String naam)</code>• <code>public void setVoornaam(String voornaam)</code>• <code>public void setEmail(String email)</code>	Uniek per gebruiker.
GUI	<ul style="list-style-type: none">• <code>public void projecteer(Multimedia media)</code>	Op deze manier kunnen er allerlei dingen geprojecteerd worden op de bril.

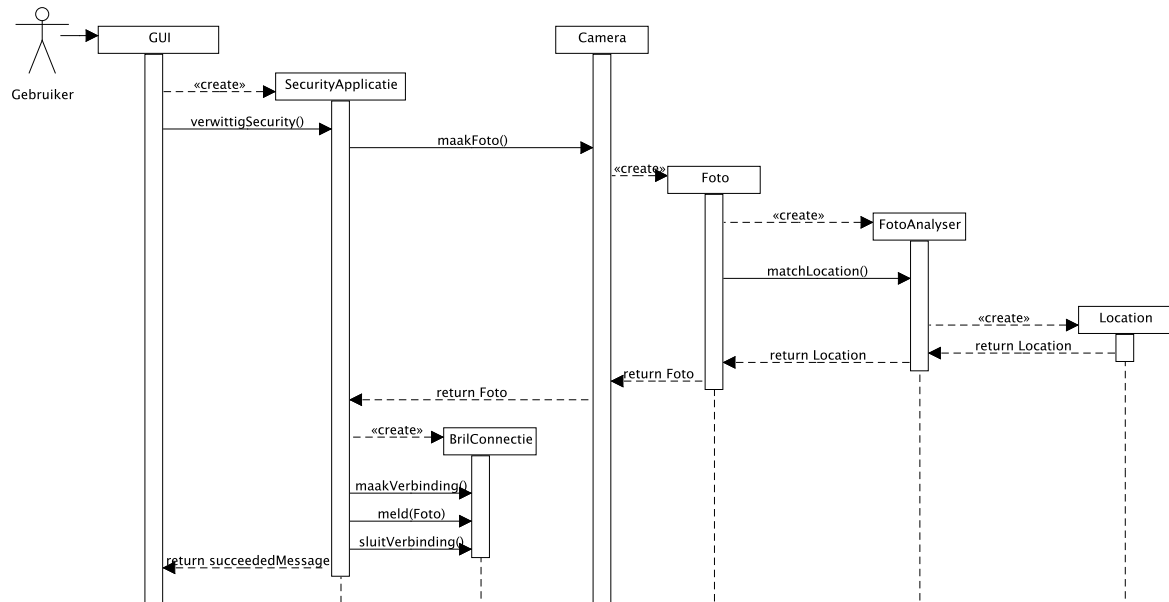
Locatie	<ul style="list-style-type: none"> • private float latitude • private float longitude • private String address • public void setLatitude(float latitude) • public void setLongitude(float longitude) • public void setAddress(String adres) • public float getLatitude() • public float getLongitude() • public float getAddress() 	
Tijdstip	<ul style="list-style-type: none"> • private int dag, maand, jaar • private int uur, minuten, seconden • public void setDatum(int dag, int maand, int jaar) • public void setUur(int uur, int minuten, int seconden) • public int[] getDatum() • public int[] getUur() 	
Camera	<ul style="list-style-type: none"> • public Foto maakFoto() • public Video maakVideo() 	
Video	<ul style="list-style-type: none"> • private Tijdstip tijdstip • private Locatie locatie 	
Foto	<ul style="list-style-type: none"> • private Tijdstip tijdstip • private Locatie locatie 	

BrilConnectie	<ul style="list-style-type: none">• private Account account1• private Account account2• public void maakVerbinding()	Verbinding tussen 2 brillen.
Apparaatconnectie	<ul style="list-style-type: none">• private String deviceId	Bv een bluetoothconnectie tussen de bril en een computer.
AudioInput	<ul style="list-style-type: none">• public void listen()	Neemt de gesproken boodschap op en geeft ze mee aan de VoiceCommandRecognizer.
VoiceCommandRecognizer	<ul style="list-style-type: none">• public void recognize(AudioInput audio)• public void voerCommandoUit(String commando)	Verwerkt de gesproken commando's van de gebruiker.
GPSApplicatie	<ul style="list-style-type: none">• private Locatie start• private Locatie bestemming• private List<Locatie> tussenpunten• public void herbereken()• public String getVolgendeInstructie()• public void setStart(Locatie locatie)• public void setBestemming(Locatie locatie)()	Wanneer de gebruiker de volgende locatie in de lijst bereikt heeft, wordt getVolgendeInstructie() opgeroepen.

FotoAnalyser	<ul style="list-style-type: none">• private Map<Foto, Locatie> fotos• public Locatie matchLocatie(Foto foto)• public void bewerkFouteLocatie(Locatie locatie)• public String herkenSilhouet(Foto foto)• public String herkenObject(Foto foto)• public Array<Float> berekenMaten()	Deze klasse analyseert en vergelijkt fotos en video-beelden met een database.
SecurityApplicatie	<ul style="list-style-type: none">• public void meldIncident(Foto foto)	Zorgt voor het aanmaken van een nieuwe BrilConnetctie en een Foto.
ARApplicatie	<ul style="list-style-type: none">• private FotoAnalyser analyser• private Foto object• public String herkenObject(Foto foto)	Zorgt voor de herkenning van objecten en het aanmaken van de overlay.
WebshopApplicatie	<ul style="list-style-type: none">• public void run	Zorgt voor de algemene coördinatie van de WebshopApplicatie.

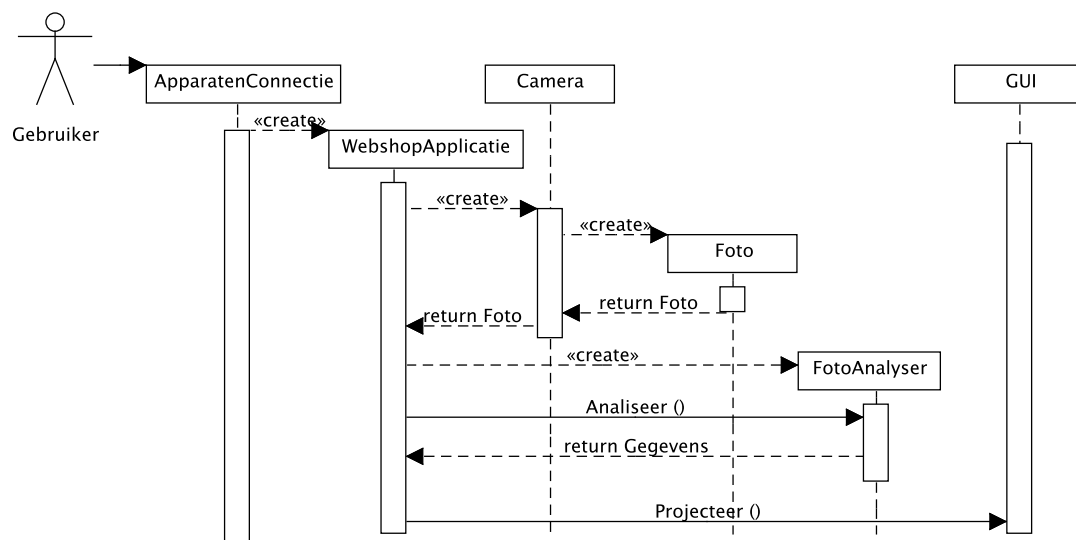
3 Interactiediagrammen

Scenario 1: Security waarschuwen



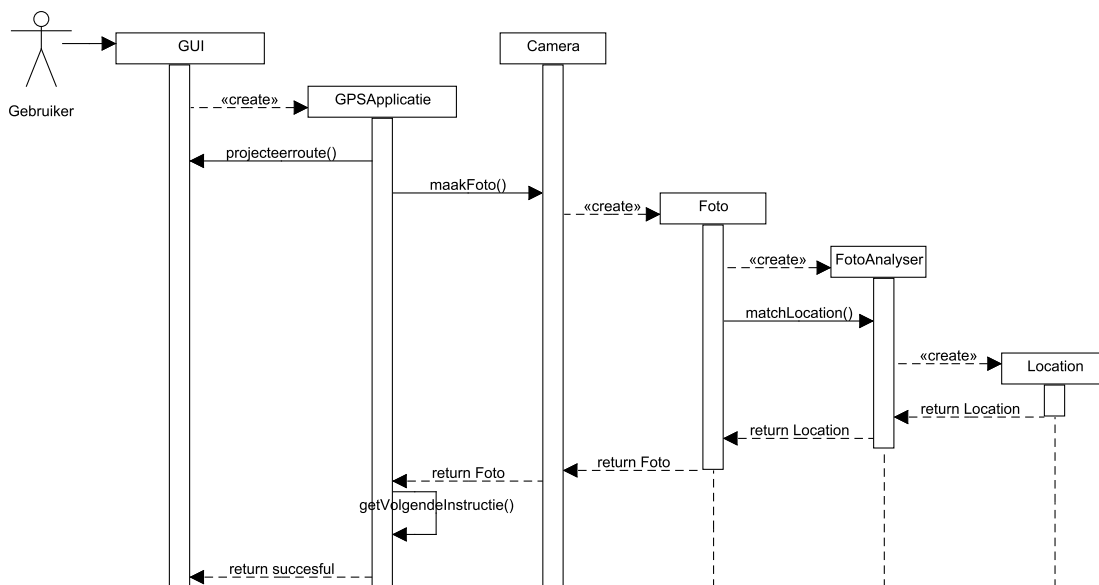
Figuur 2: Interactiediagram veiligheid

Scenario 2: Webshop



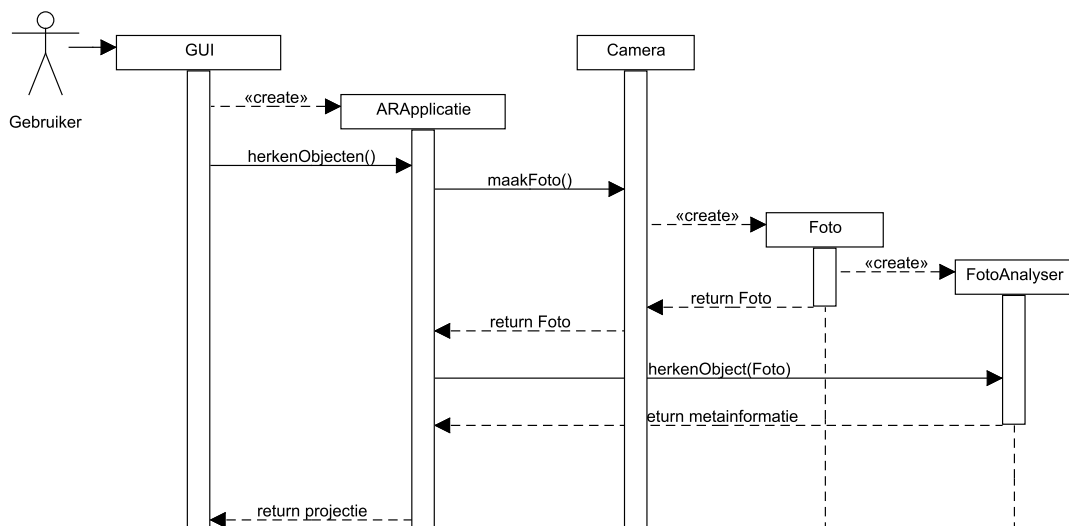
Figuur 3: Interactiediagram webshop

Scenario 3: GPS



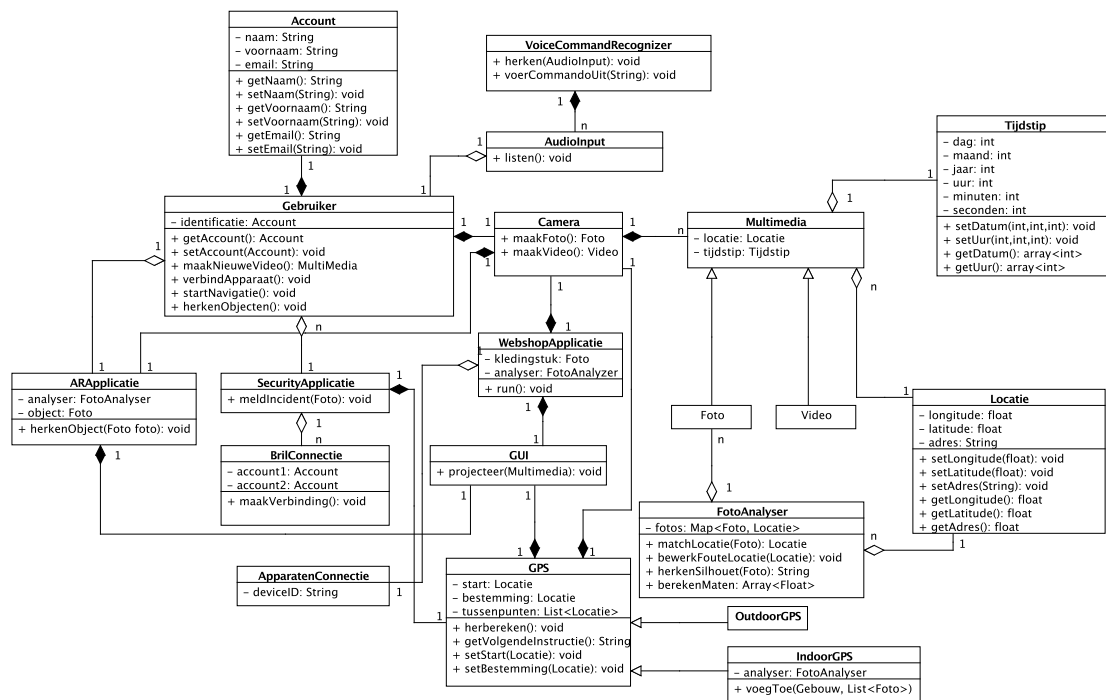
Figuur 4: Interactiediagram gps

Scenario 4: Overlay



Figuur 5: Interactiediagram overlay

4 UML klassendiagram



Figuur 6: UML diagram