UNIVERSITEIT GENT

FACULTEIT INGENIEURSWETENSCHAPPEN

Vakgroep **Informatietechnologie**

SoftwareOntwikkeling
Project: Glass

Systeem- en Gedetailleerd Ontwerp

Academiejaar 2012-2013
Groepnummer 16

Groepsleden:

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

We kozen ervoor om de **Augmented Reality** applicatie verder uit te werken.

I Systeemontwerp

1 Subsystemen

Analyse

Relevante klassen	Verantwoordelijkheden
FotoAnalyser	Neemt foto's en stuurt deze door naar de servercommunicatie. De informatie die hij terug krijgt wordt door gegeven aan de applicatie.

Camera

Relevante klassen	Verantwoordelijkheden
Camera	Het maken van de foto's.
Multimedia	Een foto/video en zijn eigenschappen.

Servercommunicatie

Relevante klassen	Verantwoordelijkheden
Connection	De connectie met de backend. Bevat methodes om data te ontvangen/te versturen.
Network	Maakt een nieuwe connectie aan. Bevat methodes voor het analyseren van foto's (wat op de server gebeurt) en het aanpassen/toevoegen van info in de databank.
ConnectionListener	Luistert naar binnenkomende Connections.
OverlayDataObject	Algemeen data-object dat kan weergegeven worden op de bril. Dit kan vergeleken worden met een HTML-pagina, die video's, foto's en tekst kan bevatten.

Leermodus

Relevante klassen	Verantwoordelijkheden
InformationEditor	Communicatieklasse met servercommunicatie, bevat de informatie die moet toegevoegd/aangepast worden.

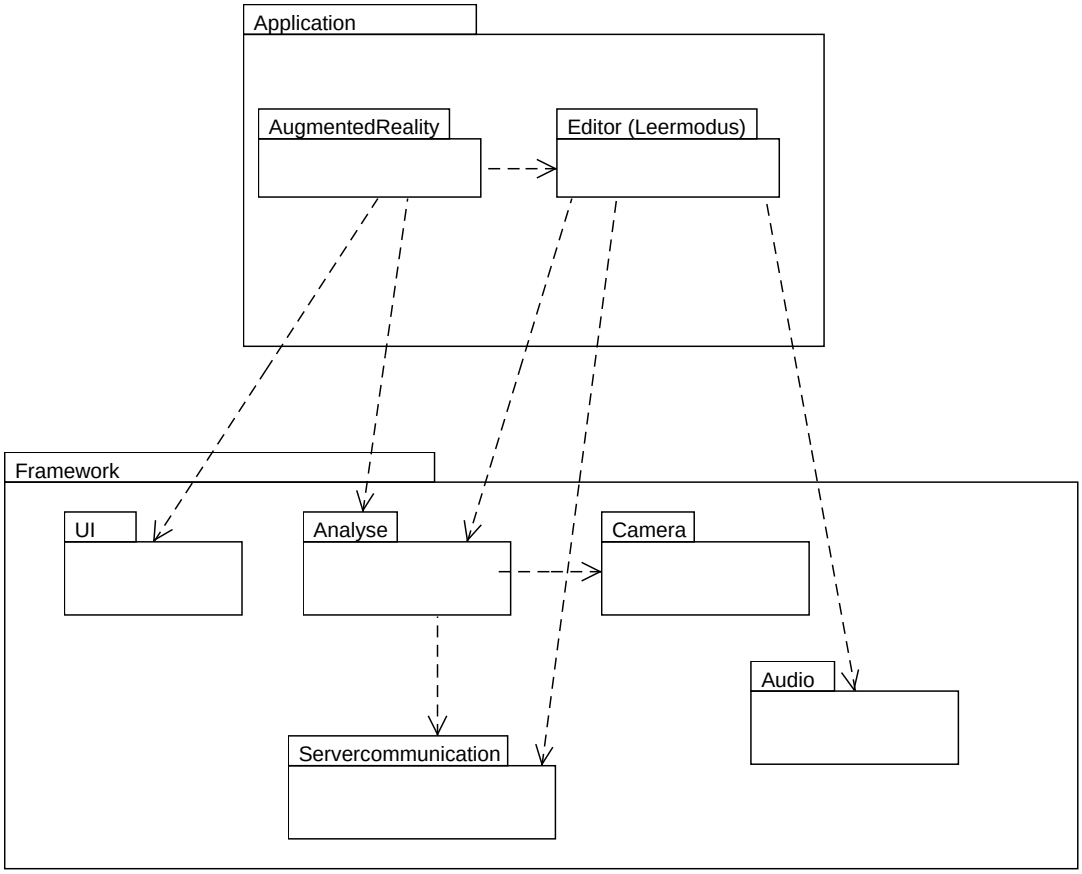
Audio

Relevante klassen	Verantwoordelijkheden
AudioInput	Het commando van de gebruiker.
VoiceCommandRecognizer	Zorgt voor de interpretatie van het commando.

AugmentedReality

Relevante klassen	Verantwoordelijkheden
ARApplication	Centrale communicatieklasse. Van hieruit kan de gebruiker kiezen tussen de overlay en de leermodus. Deze klasse is tevens een communicatieklasse met de userinterface van de Google Glass.

2 Package diagram



Figuur 1: Package diagram

We kozen ervoor om de fotoanalyse en de leermodus in aparte packages te steken. Dit kan handig zijn voor usermanagement: zo kan een verschil gemaakt worden tussen gebruikers die mogen analyseren en editeren, en gebruikers die enkel mogen analyseren. We willen de mogelijkheid voorzien dat sommige gebruikers enkel kunnen analyseren en niet kunnen editeren.

3 Contracten tussen subsystemen

Contract 1

Interface	Type	Collaborators	Classes
AnalysisInterface	client-server	Analyse en AugmentedReality	FotoAnalyser en ARApplication

Contract 2

Interface	Type	Collaborators	Classes
EditorInterface	client-server	Editor en AugmentedReality	InformationEditor en ARApplication

Contract 3

Interface	Type	Collaborators	Classes
ServerEditInterface	client-server	Network en Editor	Network en InformationEditor

Contract 4

Interface	Type	Collaborators	Classes
ServerAnalysisInterface	client-server	Network en Analyse	Network en FotoAnalyser

Interface specificatie voor contract 1

```
interface AnalysisInterface {
  public OverlayDataObject analyse();
}
```

Interface specificatie voor contract 2

```
interface EditorInterface {
  public void edit();
}
```

Interface specificatie voor contract 3

```
interface ServerEditInterface {
  public void updateObject(OverlayDataObject object);
}
```

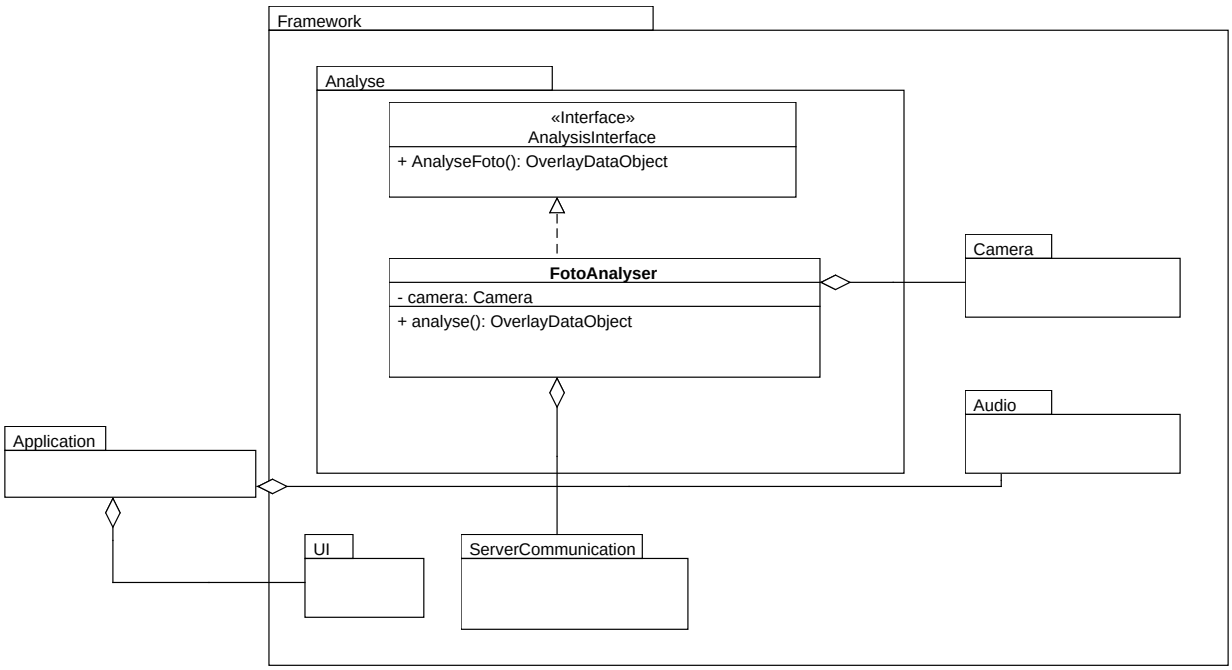
Interface specificatie voor contract 4

```
interface ServerAnalysisInterface {
  public OverlayDataObject analyseFoto(Foto foto);
}
```

II Gedetailleerd ontwerp

1 Analyse

UML klassendiagram:



Figuur 2: Analyse klassendiagram

Algoritmen en datastructuren:

Datastructuren:
De fotoanalyse gebeurt op de server vanwege de performantie. De communicatie-klasse tussen de bril en de server zit in het framework zodat andere applicaties hier ook gebruik van kunnen maken.

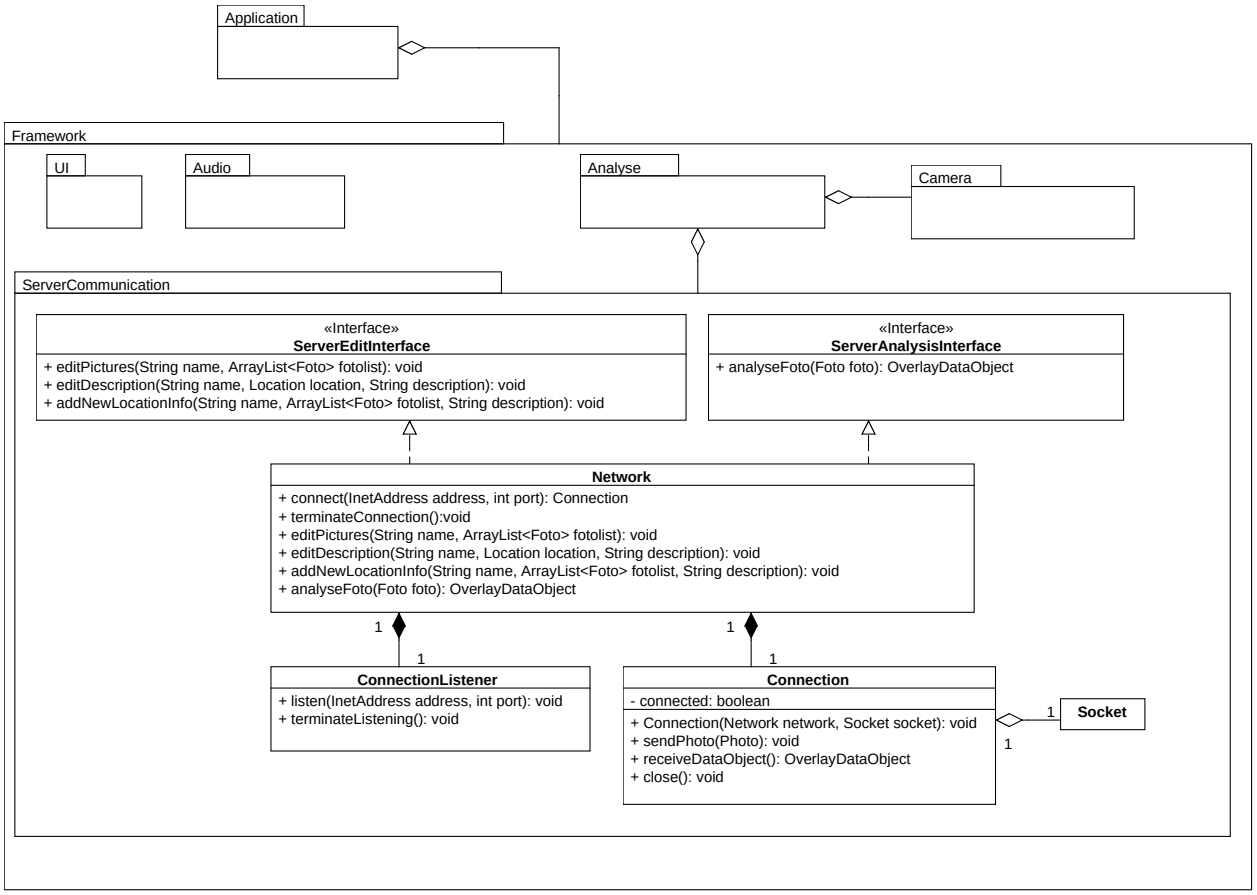
- `<FotoAnalyser>`: We nemen een foto en sturen die door naar de server. Voor de fotoherkenning wordt gebruik gemaakt van de scale-invariant feature transform (SIFT) methode. Indien een match gevonden wordt krijgen we een OverlayDataObject terug. Dit object bevat alle informatie over een bepaald object en kan onmiddellijk doorgestuurd worden naar de userinterface.

Verwachte problemen:

- Het algoritme vindt geen match. Dit kan bijvoorbeeld zijn door de kwaliteit van de foto (bvb. slechte weersomstandigheden, bewegen tijdens het nemen van de foto,...). Hier wordt dan een algoritme op los gelaten dat de kwaliteit van de foto onderzoekt. Is de kwaliteit goed genoeg, kan men ervan uitgaan dat de locatie nog niet in de databank zit en kan de gebruiker worden gevraagd om een nieuwe plaats toe te voegen. Is de foto van een lage kwaliteit, dan zal het algoritme de foto proberen verscherpen en opnieuw proberen te matchen. Wanneer dit faalt, wordt er gemeld aan de gebruiker dat de foto niet geschikt was, en kan deze desgewenst opnieuw proberen.

2 Servercommunicatie

UML klassendiagram:



Figuur 3: Servercommunicatie klassendiagram

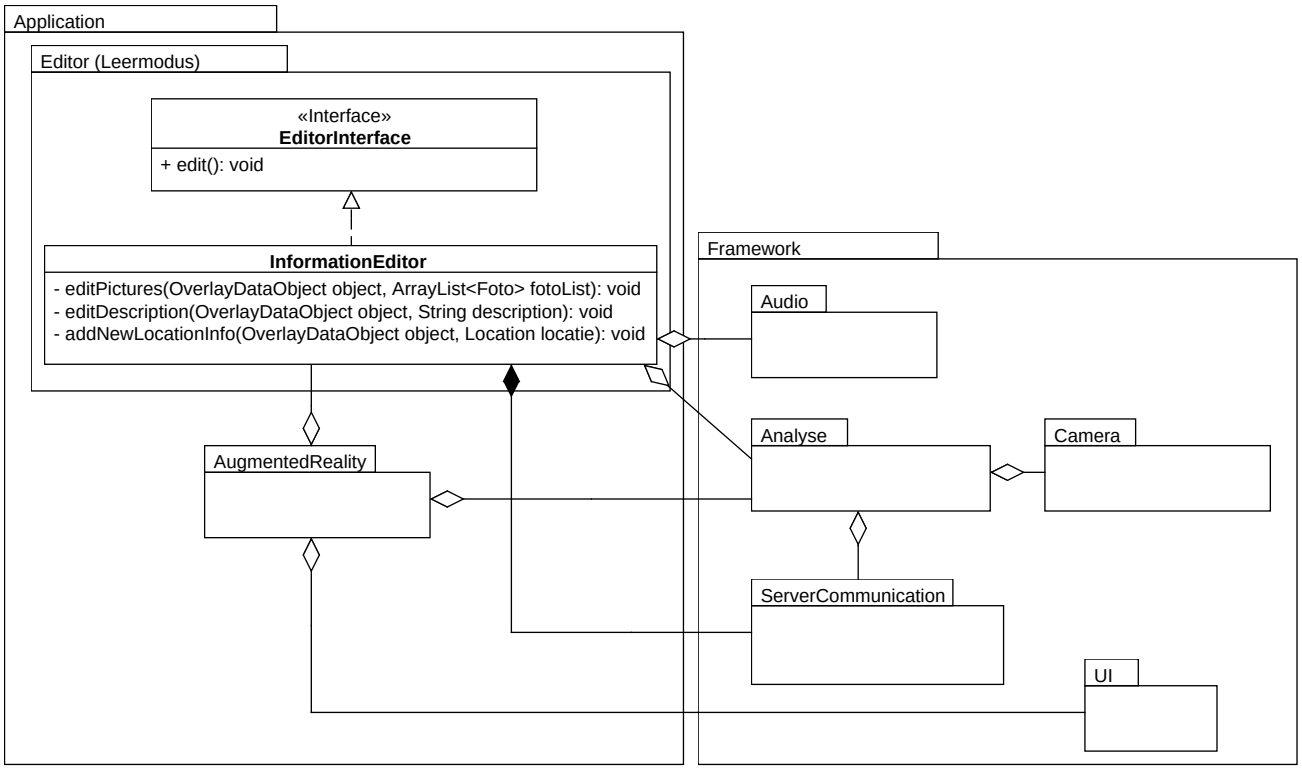
Algoritmen en datastructuren:

De methoden in de 2 interfaces worden geïmplementeerd door Network. Zo kunnen de FotoAnalyse en de InformationEditor hier gebruik van maken. Foto's kunnen aangepast/toegevoegd worden met de methode `editPictures`. In een foto zit ook telkens een locatie als veld. Een beschrijving kan toegevoegd/aangepast worden zonder een nieuwe foto mee te geven. Ten slotte kan men ook een nieuw informatiepunt toevoegen in de database.
Verwachte problemen:

- Connectie met de server faalt of valt weg.

3 Leermodus

UML klassendiagram:



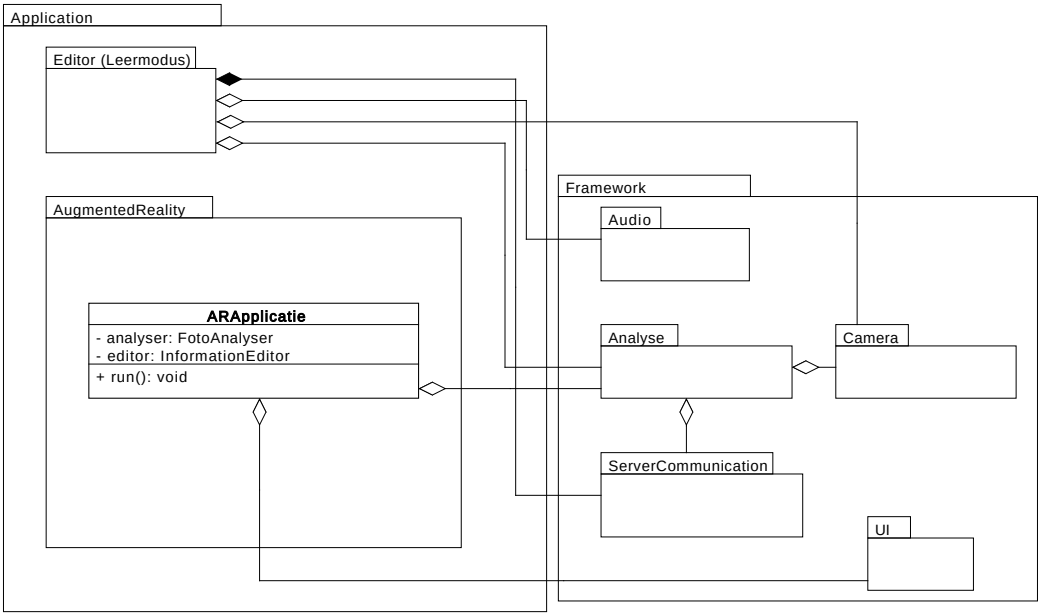
Figuur 4: Leermodus klassendiagram

Algoritmen en datastructuren:

In de leermodus kan de gebruiker info aanpassen/toevoegen met behulp van stem-commando's.

4 AugmentedReality

UML klassendiagram:



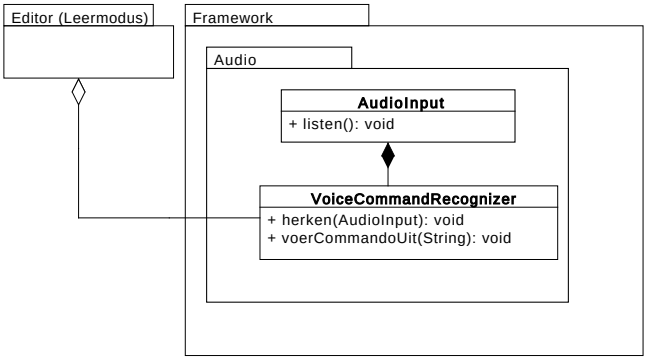
Figuur 5: AugmentedReality klassendiagram

Algoritmen en datastructuren:

De ARApplicatie wordt opgestart als de gebruiker hiervoor opdracht geeft in de UI.

5 Audio

UML klassendiagram:



Figuur 6: AugmentedReality klassendiagram

Algoritmen en datastructuren:

De ARApplicatie wordt opgestart als de gebruiker hiervoor opdracht geeft in de UI.