**Opdracht 2**  
Software engineering 3

Datum: 18 december 2018

Team: 2TI2.3  OneHouseStand

Teamleden:

* Fabian Denoodt (Teamleider)
* Glenn van Wezel
* Bram Van den Abbeele (Secretaris)
* Arnaud Wellens (Eindredacteur)
* Kobe Van De sijpe
* Alfred El Yagoubi
* Maarten Miers

# 

# Contents

[Contents 2](bookmark://_Toc532936571#_Toc532936571)

[Deel 1 3](bookmark://_Toc532936572#_Toc532936572)

[Use case, Use case diagram, Use case beschrijving, Activity diagram 3](bookmark://_Toc532936573#_Toc532936573)

[Actoren: 3](bookmark://_Toc532936574#_Toc532936574)

[Communicatie 4](bookmark://_Toc532936575#_Toc532936575)

[Pre conditie en eindconditie van elke use case: 4](bookmark://_Toc532936576#_Toc532936576)

[Screenen 6](bookmark://_Toc532936577#_Toc532936577)

[Pre conditie en eindconditie van elke use case: 6](bookmark://_Toc532936578#_Toc532936578)

[Processen, requirements en user stories: 8](bookmark://_Toc532936579#_Toc532936579)

[Primaire processen: 8](bookmark://_Toc532936580#_Toc532936580)

[Management processen: 8](bookmark://_Toc532936581#_Toc532936581)

[Strategische processen: 8](bookmark://_Toc532936582#_Toc532936582)

[Requirements: 8](bookmark://_Toc532936583#_Toc532936583)

[User stories: 8](bookmark://_Toc532936584#_Toc532936584)

[Activity Diagram 8](bookmark://_Toc532936585#_Toc532936585)

[UI prototype / mockup 10](bookmark://_Toc532936586#_Toc532936586)

[Deel 2 Objectoriëntatie Klassendiagram Domeinmodel 13](bookmark://_Toc532936587#_Toc532936587)

[Kandidaat objecten 13](bookmark://_Toc532936588#_Toc532936588)

[Woordenboek 14](bookmark://_Toc532936589#_Toc532936589)

[Klassendiagram voor feedback 15](bookmark://_Toc532936590#_Toc532936590)

[Klassendiagram na feedback 17](bookmark://_Toc532936591#_Toc532936591)

[Feedback van andere groepen 17](bookmark://_Toc532936592#_Toc532936592)

[Klassendiagram 17](bookmark://_Toc532936593#_Toc532936593)

[Code generatie 20](bookmark://_Toc532936594#_Toc532936594)

[Deel 3 20](bookmark://_Toc532936595#_Toc532936595)

[User stories die gevarieerd gedrag van de objecten vragen 20](bookmark://_Toc532936596#_Toc532936596)

[Acceptatietesten 20](bookmark://_Toc532936597#_Toc532936597)

[Toestandsdiagrammen van de (super)klassen 22](bookmark://_Toc532936598#_Toc532936598)

[Sequentiediagram 23](bookmark://_Toc532936599#_Toc532936599)

[Aanpassingen 9dec: 26](bookmark://_Toc532936600#_Toc532936600)

[Code generatie: 27](bookmark://_Toc532936601#_Toc532936601)

[Unit testen: 34](bookmark://_Toc532936602#_Toc532936602)

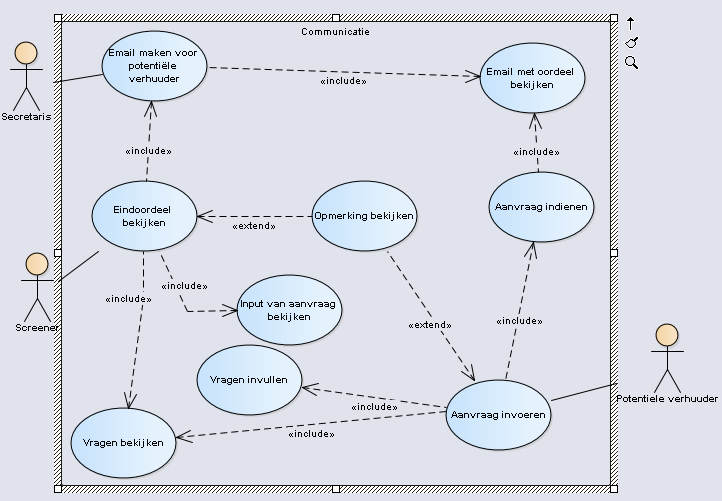
# Deel 1

Use case, Use case diagram, Use case beschrijving, Activity diagram

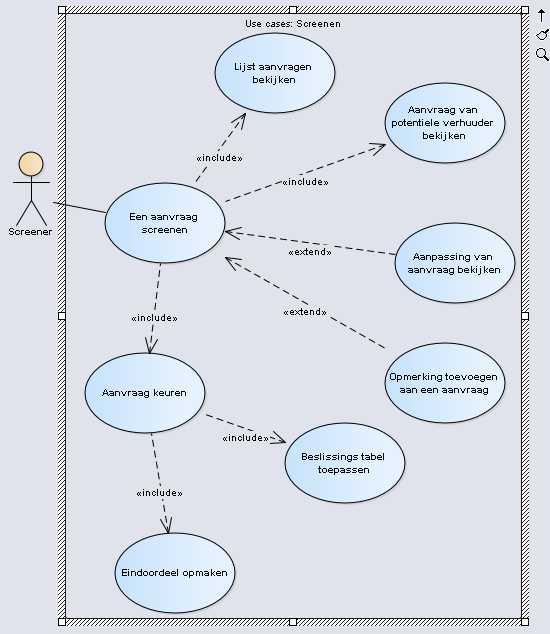
Actoren:

* Secretaresse
* Screener
* Potentiële verhuurder

Omdat het use case diagram te complex zou worden, besloten we om deze in twee te splitsen, namelijk **communicatie** en **screenen**.



Een potentiële verhuurder gaat een aanvraag invoeren en die aanvraag gaat door naar de screener.  
De screener zal een eindoordeel hebben en geeft deze door aan de secretaris zodat de secretaris een e-mail met daarin het oordeel kan sturen naar de potentiële verhuurder.



Een screener zal een aanvraag screenen adhv een beslissingstabel toe te passen en zal dan zo zijn oordeel opmaken.

Communicatie

### Pre conditie en eindconditie van elke use case:

Aanvraag invoeren

* **Preconditie:** De potentiële verhuurder wil zijn pand aanbieden om te verhuren.
* **Eindconditie bij succes:** De potentiële verhuurder zit op het scherm om een aanvraag in te voeren.
* **Eindconditie bij falen:** De potentiële verhuurder kan niet tot op het aanvraag scherm komen.

Vragen invullen

* **Preconditie:** De potentiële verhuurder ziet een scherm met oningevulde vragen
* **Eindconditie bij succes:** Alle vragen zijn ingevuld.
* **Eindconditie bij falen:** Niet alle vragen werden ingevuld en er wordt een foutmelding getoond.

Vragen bekijken

* **Preconditie:** Alle vragen zijn ingevuld.
* **Eindconditie bij succes:** De potentiële verhuurder en screener zien een scherm met ingevulde vragen.
* **Eindconditie bij falen:** De vragen konden niet bekeken worden en er wordt een foutmelding getoond.

Aanvraag indienen

* **Preconditie:** Alle vragen zijn ingevuld.
* **Eindconditie bij succes:** De aanvraag is opgestuurd en de potentiële verhuurder krijgt hier een melding van.
* **Eind conditie bij falen:** De aanvraag kon niet worden verstuurd, en krijgt hier een foutmelding van.

Email met oordeel bekijken:

* **Preconditie:** De secretaris heeft een email opgesteld met het oordeel van de screener.
* **Eindconditie bij succes:** Zowel de secretaris als de potentiële verhuurder kunnen deze email zien
* **Eind conditie bij falen:** De secretaris of potentiële verhuurder kan de email niet bekijken en krijgt een foutmelding te zien.

Opmerking bekijken:

* **Preconditie:** De screener heeft een opmerking toegevoegd op een incorrect veld, of onduidelijke foto.
* **Eindconditie bij succes:** Als de screener op de knop “opmerkingen” klikt krijgt hij een overzicht van alle opmerkingen van een pand, en als de potentiële verhuurder de opmerkingen kan zien van zijn eigen pand.
* **Eind conditie bij falen:** De screener heeft geen overzicht van opermkingen, of de potentiele verhuuder kan de aan hem toebehorende opmerkingen niet zijn. Er verschijnt een foutmelding. (in beide gevallen)

Eindoordeel bekijken

* **Preconditie:** Er is eindoordeel vastgelegd door de screener.
* **Eindconditie bij succes:** Het eindoordeel kan bekeken worden door de secretaris.
* **Eind conditie bij falen:** Het eindoordeel kan niet bekeken worden.

Email maken voor potentiele verhuurder:

* **Preconditie:** De screener heeft een pand beoordeeld en de secretaris kan de status bekijken en zit op het scherm om een e-mail te sturen.
* **Eindconditie bij succes:** de secretaris heeft een mail gestuurd naar de potentiele verhuuder in verband met zijn aanvraag waarin het oordeel vermeld staat.
* **Eind conditie bij falen:** de secretaris kon geen mail sturen naar de potentiele verhuurder en krijgt een foutmelding te zien.

Input van aanvraag bekijken

* **Preconditie:** Een potentiële verhuurder heeft een aanvraag ingediend.
* **Eindconditie bij succes:** De lijst van inputs kan succesvol bekeken worden.
* **Eind conditie bij falen:** De lijst van inputs kan niet succesvol bekeken worden.

## Screenen

Pre conditie en eindconditie van elke use case:

Een aanvraag screenen:

* **Preconditie:** Er is een nieuwe aanvraag ingediend, of een al bestaande aanvraag is bijgewerkt
* **Eindconditie bij succes:** Een aanvraag is succesvol gekeurd.
* **Eind conditie bij falen:** Een aanvraag kon niet gekeurd worden

Lijst aanvragen bekijken

* **Preconditie:** De screener zit op het scherm waar de aanvragen te voorschijn moeten komen.
* **Eindconditie bij succes:** De lijst van aanvragen komen tevoorschijn en de status van elke aanvraag is zichtbaar.
* **Eind conditie bij falen:** De lijst van aanvragen kon niet worden weergegeven en er komt een foutmelding tevoorschijn.

Aanvraag van potentieel verhuurder bekijken

* **Preconditie:** De screener heeft een aanvraag aangeklikt.
* **Eindconditie bij succes:** De screener komt op het scherm van de aanvraag terecht en heeft een overzicht van de ingegevulde gegevens.
* **Eind conditie bij falen:** De aanvraag kon niet worden weergegeven en er verschijnt een foutmelding.

Aanpassing van aanvraag bekijken

* **Preconditie:** De screener zit op het scherm van een bepaalde aanvraag en er zijn aanpassingen gemaakt door de verhuurder op een opmerking van de screener.
* **Eindconditie bij succes:** De screener ziet in rood de aangepaste onderdelen.
* **Eind conditie bij falen:** Er komen geen aanpassingen tevoorschijn en er verschijnt een foutmelding.

Opmerking toevoegen aan een aanvraag:

* **Preconditie:** Een veld is niet correct ingevuld door de potentiele verhuurder, of de ingevoerde afbeeldingen zijn niet duidelijk
* **Eindconditie bij succes:** De screener heeft een opmerking toegevoegd.
* **Eind conditie bij falen:** De screener kan de opmerking niet toevoegen en er verschijnt een error-boodschap.

Aanvraag keuren:

* **Preconditie:** De screener staat op het scherm van een ongekeurde aanvraag.
* **Eindconditie bij succes:** Er is een beslissing gemaakt over het pand.
* **Eind conditie bij falen:** Er kon geen beslissing gemaakt worden.

Beslissingstabel toepassen:

* **Preconditie:** De screener staat op scherm van een ongekeurde aanvraag en is die aan het keuren. De ingevulde gegevens zijn correct ingevuld en de afbeeldingen zijn duidelijk.
* **Eindconditie bij succes:** De screener bekomt een eindscore a.d.h.v. de beslissingstabel.
* **Eind conditie bij falen:** Er kon geen eindscore gemaakt worden, en de screener ziet waar het mis ging.

Eindoordeel opmaken:

* **Preconditie:** Een pand is gekeurd en er is nog geen eindoordeel voor opgemaakt.
* **Eindconditie bij succes:** Er is een eindoordeel gemaakt en de secretaris heeft hier toegang toe.
* **Eind conditie bij falen:** Het eindoordeel is niet beschikbaar.

# Use case beschrijving van een belangrijke use case

|  |  |
| --- | --- |
| USE CASE Titel | **Een aanvraag screenen** |
| Versie | 0.1 |
| Primaire Actor | Screener |
| Secundare actoren | Secretaris |
| Preconditie | Er is een nieuwe aanvraag ingediend door de potentiële verhuurder, en de screener staat op het scherm waar hij de lijst van ongekeurde aanvragen ziet. |
| Eindconditie bij success | De screener heeft het pand beoordeeld adhv beslissingstabel en het is goedgekeurd (>= 23 punten in de beslissingstabel). |
| Eindconditie bij fallen | Pand afgekeurd (minder dan 23 punten in de beslissingstabel), stuur redenen door naar secretaris. |
| **Hoofdscenario** | 1 De screener klikt op een ongekeurde aanvraag. |
|  | 2 Het systeem toont de vragen en ingevoerde antwoorden van het aanvraagformulier. |
|  | 3 De screener klikt op “volgende” om de afbeeldingen te bekijken. |
|  | 4 Het systeem geeft de afbeeldingen weer van de aanvraag |
|  | 5 De screener klikt op “volgende” |
|  | 6. De afbeeldingen blijven op scherm staan, maar het systeem voegt de vragen uit de beslissings tabel toe aan het scherm |
|  | 7. De screener geeft een waarde in bij elke vraag. De waarde komt van de afbeeldingen. |
|  | 8. Het systeem telt de punten bij elkaar op en toont het eindresultaat aan de screener. |
|  | 10. Het systeem toont aan dat pand goedgekeurd is. |
|  | 11. De screener klikt op bevestig. |
|  | 12. Het systeem verandert de status van de aanvraag in “Goedgekeurd”. |
|  |  |
| **Uitbreidingen** | 2.1 De screener merkt op dat de velden incorrect zijn ingevuld, duidt dit aan, en plaatst een opmerking met commentaar in het systeem |
|  | 2.2 Het systeem veranderd de aanvraag van “ongekeurd” naar “wachten op aanpassingen van gebruiker”. |
|  | 4.1 De screener merkt op dat er te weinig afbeeldingen zijn, of of de afbeeldingen onduidelijk zijn en plaatst hier een opmerking voor in het systeem. |
|  | 4.2 Het systeem veranderd de aanvraag van “ongekeurd” naar “wachten op aanpassingen van gebruiker”. |
|  | 8.1 Het systeem toont de eindscore en toont aan dat het pand is afgekeurd vermits die minder dan 23 punten is. |
|  | 8.2 De screener klikt op bevestig. |
|  | 8.3 Het systeem verandert de status van de aanvraag in “Afgekeurd”. |
|  |  |
| Speciale vereiste |  |

## Processen, requirements en user stories:

### Primaire processen:

* Een aanvraag screenen
* Een nieuw pand toevoegen

Managementprocessen:

* Online platform beheren

### Strategische processen:

* Innoveren (Nieuwe features toevoegen aan website/ applicatie die wij gaan ontwikkelen voor hun)
* Ons nieuw verhuurders sollicitatie systeem implementeren

### Requirements:

* Als screener wil ik de secretaris duidelijk maken dat ik een pand ter plekke wil screenen.
* Als screener wil ik afbeeldingen van een aanvraag kunnen bekijken
* Als screener wil ik een aanvraag kunnen goed of af keuren
* Als screener wil ik de secretaris kunnen verwittigen van mijn beslissing
* Als screener wil ik de status van een applicatie kunnen doorgeven (begonnen, in proces, done)

### User stories:

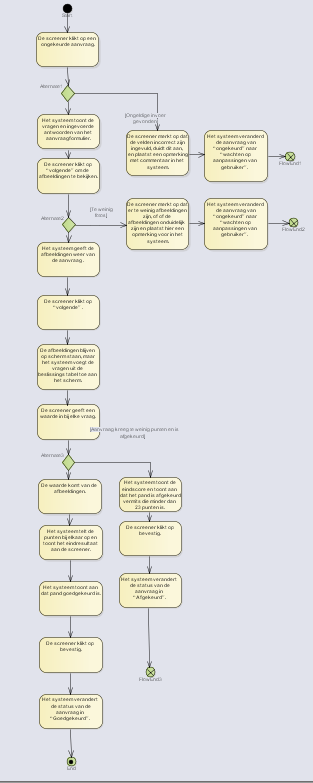
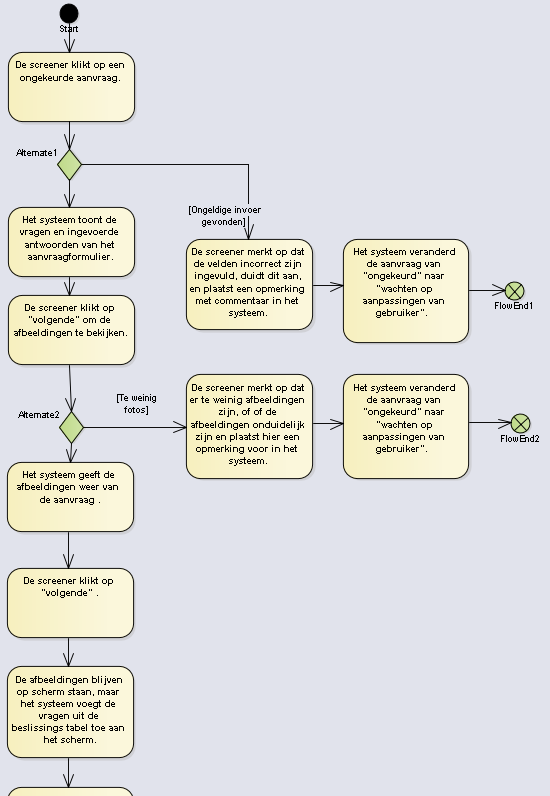
Als screener wil ik de foto’s van de potentiële verhuurder kunnen bekijken en er een rating op zetten, zodat ik de verschillende panden kan vergelijken en zo de beste uitkiezen. Zodat hij bij de voortzetting van zijn werk de meest waardevolle panden kan bezien.

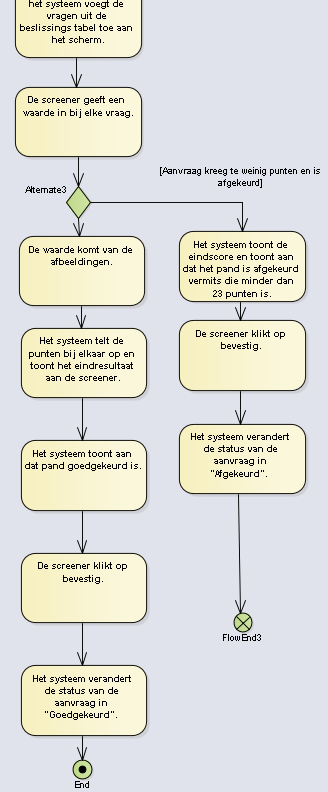
Als screener wil ik de secretaris mijn oordeel over een pand doorsturen zodat zij de potentiële verhuurder op de hoogte kan houden van de status van het pand.

Als screener wil ik een overzicht van de panden hun locatie zodat ik milder kan zijn waar er minder panden

Als screener wil ik de secretaris eenvoudig duidelijk kunnen maken dat ik het pand fysiek wil kunnen screenen zodat zij voor mij een afspraak kan regelen

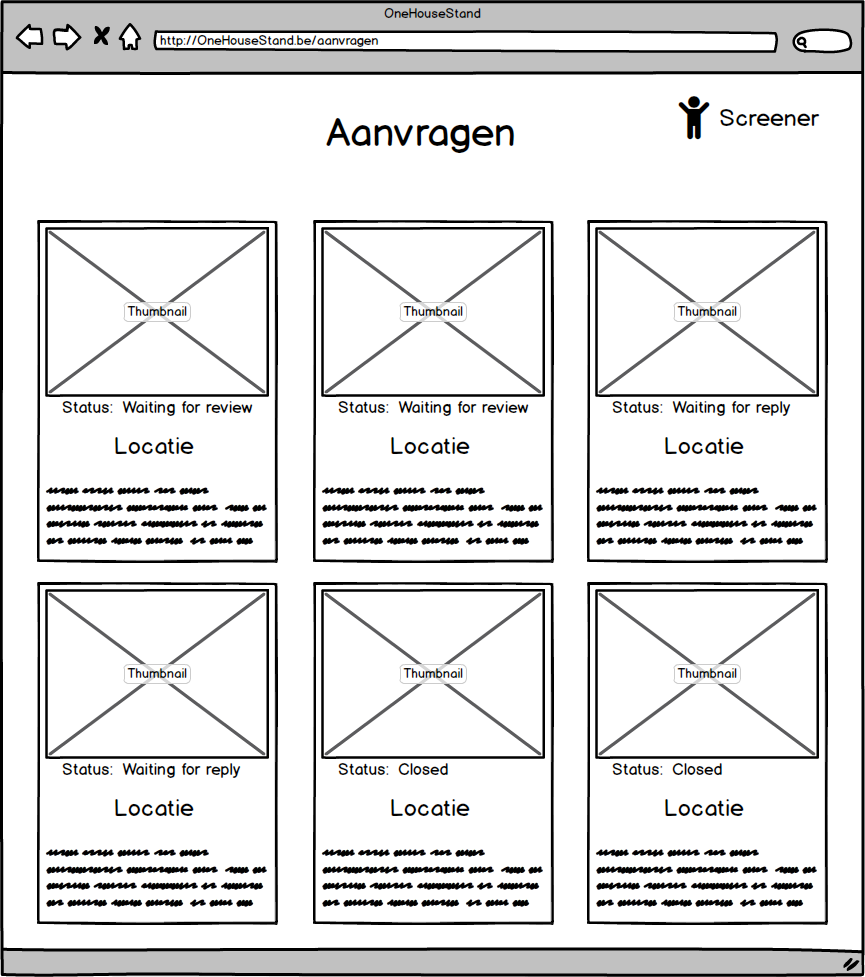
Activity Diagram



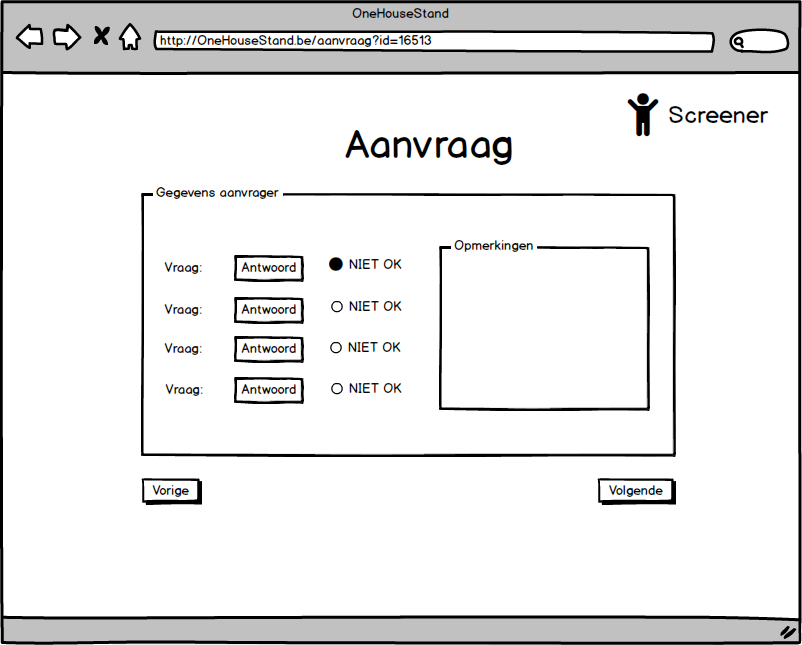


UI prototype / mockup

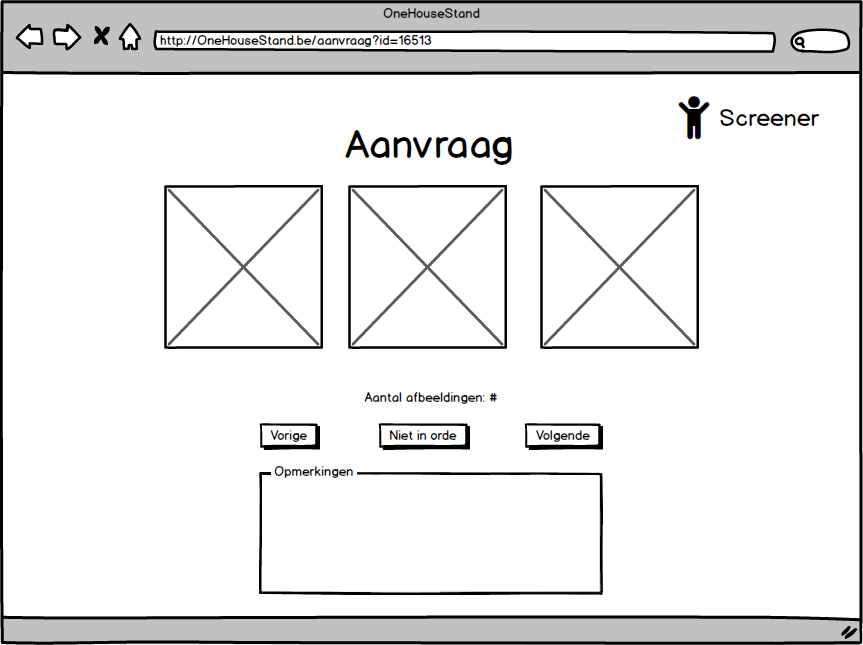
Dit zijn de Mockups/Low fidelity prototypen die we gemaakt hebben aan de hand van Balsamiq© .



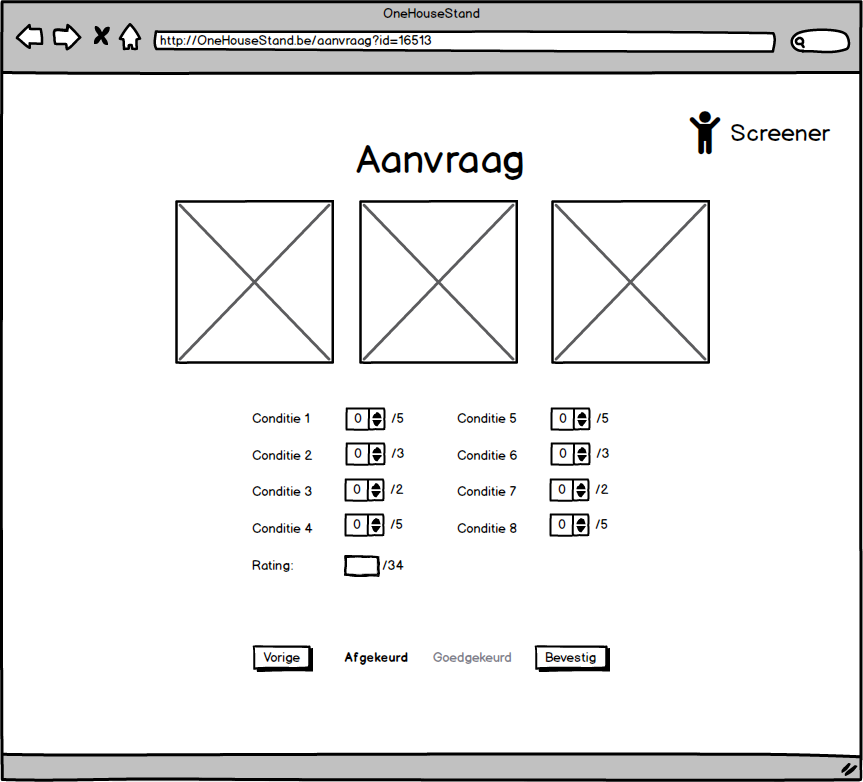
Het overzicht van alle aanvragen die de screener nog moet beoordelen/behandelen



De pagina waarop de screener de aanvraag beoordeelt, hierop ziet hij de aanvraag en kan hij aanduiden of een bepaalde vraag niet goed ingevuld was of niet accepteerbaar was.



Hier kijkt de screener of de foto’s wel van voldoende kwaliteit zijn (niet flu, niet te weinig foto’s, geen foto’s die missen, etc).



Hier gaat de screener effectief een score geven aan de aanvraag, hij ziet de foto’s en geeft punten op basis van decision table criteria.

# Deel 2 Objectoriëntatie Klassendiagram Domeinmodel

Kandidaat objecten

De eerste stap naar het klassendiagram toe is het opzoek gaan naar kandidaat objecten.

Deze worden gevonden door het zoeken van zelfstandige naamwoorden in de volgende onderdelen die reeds afgewerkt zijn: probleemstelling, VPC, scope of work, scope of product, business processes, requirements, user stories en tot slot het use casemodel. We zijn volgend overzicht bekomen:

|  |  |  |  |
| --- | --- | --- | --- |
| **Kandidaat-objecten** | **Relevant voor probleem? Nodig voor de oplossing?** | **Bestaat het op zichzelf? Heeft eigen attributen/operaties?** | **Dubbels schrappen** |
| Screener (10) | Screener (10) | Screener (10) | Screener (10) |
| Foto's (2) | Foto's (2) | / | / |
| Verhuurder (9) | Verhuurder (9) | Verhuurder (9) | / |
| Rating (3) | Rating (3) | / | / |
| Panden (4) | Panden (4) | Panden (4) | / |
| Voortzetting (1) | / | / | / |
| Werk (2) | / | / | / |
| Secretaris (14) | Secretaris (14) | Secretaris (14) | Secretaris (17) |
| Secretaresse (3) | Secretaresse (3) | Secretaresse (3) | / |
| Afspraak (3) | Afspraak (3) | / | / |
| Hoogte (1) | / | / | / |
| Status (4) | Status (4) | / | / |
| Aanvraag (12) | Aanvraag (12) | Aanvraag (12) | Aanvraag (24) |
| Sollicitatie (4) | Sollicitatie (4) | Sollicitatie (4) | / |
| Idee (1) | Idee (1) | / | / |
| Oordeel (4) | Oordeel (4) | / | / |
| Beeld (1) | / | / | / |
| Rangorde (2) | / | / | / |
| Verhuurders (3) | Verhuurders (3) | Verhuurders (3) | / |
| Agenda (3) | Agenda (3) | / | / |
| Overzicht (2) | / | / | / |
| Screeningen (1) | / | / | / |
| Toegang (1) | / | / | / |
| Dag (1) | / | / | / |
| Huurder (3) | / | / | / |
| Programma (2) | / | / | / |
| Tijd (1) | / | / | / |
| Kwaliteit (2) | / | / | / |
| Huizen (3) | Huizen (3) | Huizen (3) | / |
| Diversiteit (1) | / | / | / |
| Applicatie (3) | / | / | / |
| Programma's (1) | / | / | / |
| Inbreng (2) | / | / | / |
| Doelgroep (1) | Doelgroep (1) | / | / |
| Huis (4) | Huis (4) | Huis (4) | / |
| Gevoel (1) | / | / | / |
| Klant (4) | Klant (4) | Klant (4) | / |
| Klanten (2) | Klanten (2) | Klanten (2) | / |
| Applicatie (3) | Applicatie (3) | Applicatie (3) | / |
| Applicatie process (2) | Applicatie process (2) | / | / |
| Velden (1) | / | / | / |
| Aanvragen (2) | Aanvragen (2) | Aanvragen (2) | / |
| Knoppen (1) | / | / | / |
| Invoering (1) | / | / | / |
| Pand (10) | Pand (10) | Pand (10) | Pand (21) |
| Beheer (1) | / | / | / |
| Gegevens (3) | Gegevens (3) | / | / |
| Afbeelding (2) | Afbeelding (2) | / | / |
| Contactinformatie (1) | Contactinformatie (1) | / | / |
| Beslissing (4) | Beslissing (4) | / | / |
| Uren (2) | / | / | / |
| Druk (1) | / | / | / |
| Manager (2) | / | / | / |
| Technologieën (2) | / | / | / |
| Platform (3) | / | / | / |
| Potentiele  Verhuurder (8) | Potentiele  Verhuurder (8) | Potentiele  Verhuurder (8) | Potentiele  Verhuurder (26) |
| Ouders (1) | / | / | / |
| Villa (1) | / | / | / |
| Band (1) | / | / | / |
| Sukkelaar (1) | / | / | / |
| Mensen (2) | / | / | / |
| Kot (1) | / | / | / |
| Email (6) | Email (6) | Email (6) | Email (6) |
| Eindoordeel (4) | Eindoordeel (4) | / | / |
| Opmerking (3) | Opmerking (3) | / | / |
| Input (1) | / | / | / |
| Aanpassing (3) | Aanpassing (3) | / | / |
| Beslissingstabel (8) | Beslissingstabel (8) | / | / |
| Aanvraagformulier (3) | Aanvraagformulier (3) | Aanvraagformulier (3) | / |
| Afbeeldingen (4) | Afbeeldingen (4) | / | / |
| Systeem (2) | / | / | / |
| Waarde (2) | / | / | / |
| Vraag (1) | / | / | / |
| Punten (3) | Punten (3) | / | / |
| Eindresultaat (4) | Eindresultaat (4) | / | / |
| Commentaar (2) | Commentaar (2) | / | / |
| Eindscore (4) | Eindscore (4) | / | / |

Woordenboek

Nadat we de dubbels geschrapt hebben bekwamen we een overzicht van de zelfstandige naamwoorden die we zullen moeten gebruiken in het klassendiagram. We kunnen nog steeds niet aan het klassendiagram beginnen. We moeten eerst een woordenboek samenstellen met de zelfstandige naamwoorden die we bekomen zijn.

Dit woordenboek zorgt ervoor dat de klassen die we zullen gebruiken duidelijk zijn voor iedereen die ermee zal moeten werken, zodat er later geen misverstanden ontstaan.

**Screener:** Een werknemer van OnehouseStand die de aanvraag van een potentiële verhuurder beoordeelt.

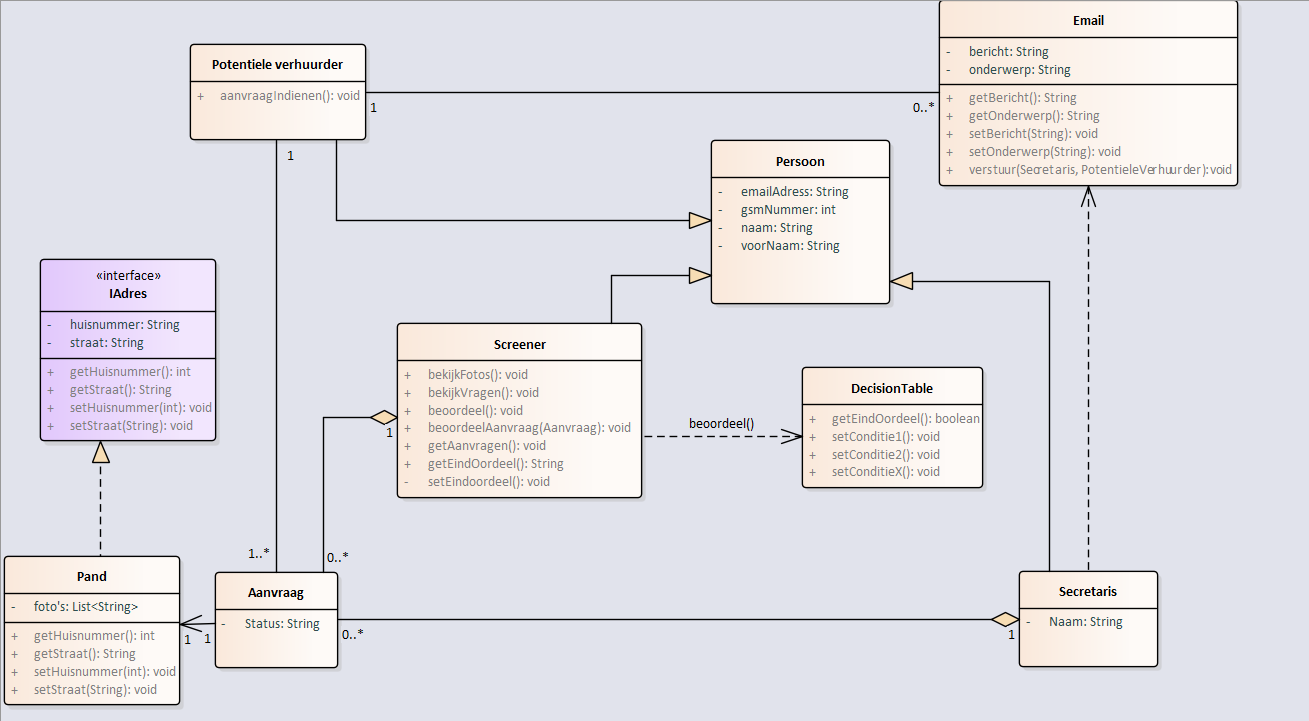
**Potentiële verhuurder:** Dient een aanvraag in om zijn pand te keuren, zodat OneHouseStand zijn pand kan verhuren.

**Secretaris:** Communiceert het eindoordeel van een aanvraag terug naar de potentiële verhuurder.

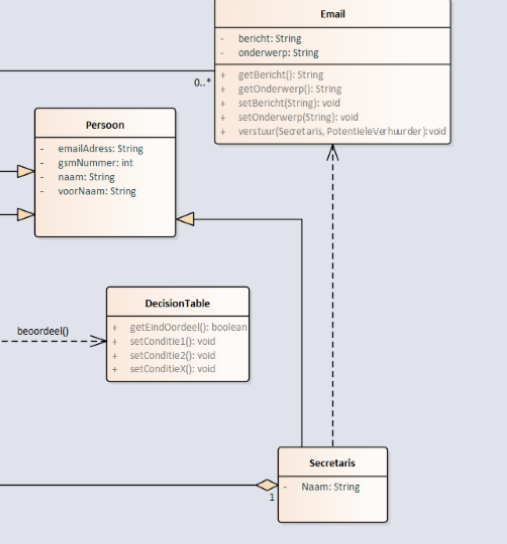
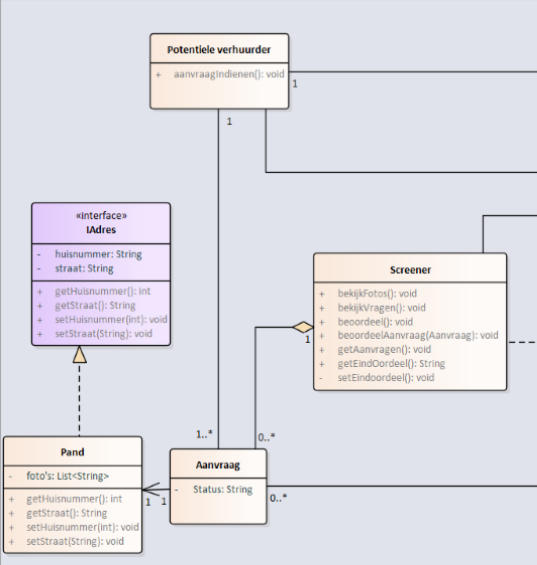
**Pand:** Dit is het huis/eigendom van een potentiële verhuurder, die hij aangeeft in zijn aanvraag en zou willen verhuren.

**Aanvraag:** Dit is wat een potentiële verhuurder indient om zijn pand te verhuren. In dit formulier zitten de algemene gegevens van zijn pand samen met de foto’s. Dit is wat beoordeeld zal worden door de screener.

**E-mail:** Berichtgeving tussen secretaris en potentiële verhuurder, waarin het eindoordeel van de aanvraag staat.

Klassendiagram pre feedback

Hierboven staat het volledige klassendiagram dat we gemaakt hebben.



Hierboven staan 2 afbeeldingen van hetzelfde klassendiagram, maar uitvergroot zodat het beter leesbaar is.

### Uitleg bij klassendiagram

In dit diagram is er niet echt een begin dus neem ik een van de belangrijkste klassen als beginpunt, namelijk de Persoon klasse.

Dit is een superklasse, het bevat de algemene attributen van een persoon, een naam, voornaam, email en gsmnummer.

De subklassen van Persoon zijn Secretaris, Screener en Potentiele verhuurder; Potentiele verhuurder bevat enkel de Methode aanvraagIndienen() omdat potentiele verhuurder enkel dit moet kunnen, hiervoor heeft hij dan een bi-directionele relatie met de aanvraag klasse.

De Screener klasse is een van de grootste, want de screener gaat het meeste functionaliteit hebben binnen het systeem, hij kan foto's bekijken, vragen bekijken, een oordeel maken, beoordelen, aanvragen opvragen , en eindoordelen van aanvragen zien en zetten. Screener heeft ook een relatie met aanvraag omdat die die moet kunnen bekijken /evalueren.

Ten slotte heeft de screener een relatie met decision table, de klasse die hij gaat gebruiken om de finale score van een aanvraag te berekenen.

De Secretaris heeft een naam (een fout die via feedback verwijderd is geweest), en heeft een relatie met Email, Persoon (overerving) en aanvraag.

De relatie met email bestaat omdat de Secretaris het communicatie-midden is die via email de potentiele verhuurder op de hoogte gaat houden van veranderingen aan zijn aanvraag (vandaar ook de relatie met aanvraag).

De Email klasse zelf spreekt voor zichzelf, het heeft een bericht en een onderwerp, en heeft relaties met de enige klassen die inboxen hebben en dus mails kunnen versturen of ontvangen.

De klasse aanvraag zelf heeft dan enkel een status in de vorm van eens tring, dit bevat dan bv “onbehandeld”, “in behanding”, “goedgekeurd”, “afgekeurd”, “in afwachting aanpassing door potentiele verhuurder”. Op basis van deze status gaat de secretaris communiceren met de potentiele verhuurder.

Ten slote is er de klasse Pand, die bevat een lijst met foto’s, en deze implmenteerd de interface IAdres die alle algemene adres-attributen bevat zoals huisnummer, straatnaam, gemeente, etc...

Klassendiagram na feedback

### Feedback van andere groepen

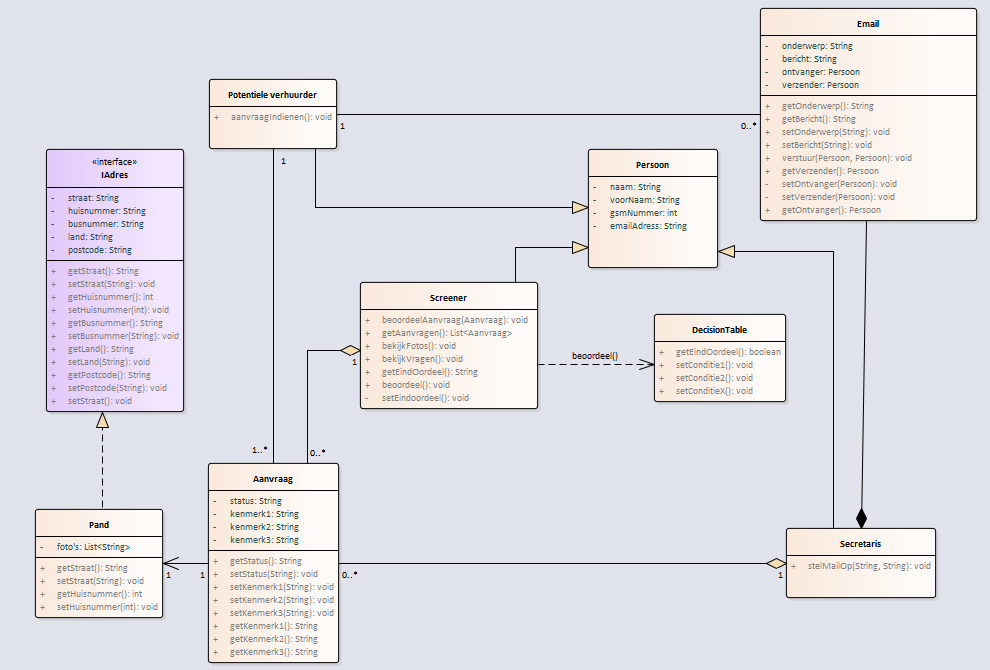
Wij hebben ons eerste klassendiagram getoond aan 3 medestudenten, elk van een andere software groep.

Dit waren hun opmerkingen die uiteindelijk geleid hebben tot veranderingen in onze klassen diagram.

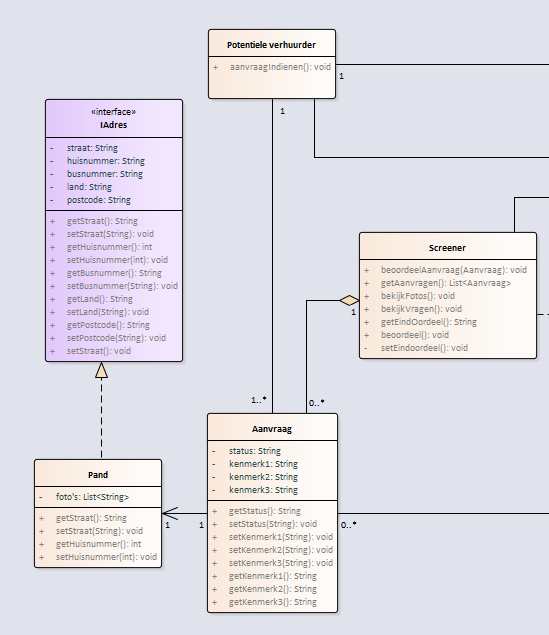
* **Diana:**
  + “Land toevoegen” => Land toevoegen bij IAdress interface
  + “Naam staat bij secretaris EN bij persoon” => Naam verwijderen bij de Secretaris klasse
* **Maxime:**
  + “Busnummer toevoegen”=> Busnummer toegevoegd aan IAdres interface
  + “Source en destination toevoegen bij email klasse” => Idem opmerking
  + “Secretaris moet methode hebben “verzend email”” => StelEmailOp() toegevoegd aan Secretaris
  + “Potentiele verhuurder moet email kunnen lezen” => Niets
  + “Er zou extra info over het pand moeten bijgehouden worden (aantal kamers, oppervlakte, etc)” => niets
* **Eilish:**
  + “Getaanvragen is een void, dat is niet oke” => getaanvragen verwijderd van de screener.
  + “Naam bij secretaris EN bij persoon” => Naam verwijderd van Secretaris klasse
  + “IAdress: land toevoegen, stad toevoegen, postcode toevoegen, busnr toevoegen” => al de vorige toegevoegd aan IAdress.

### Klassendiagram

Dit is de laatste versie van ons klassendiagram. Tussen deze versie en het klassendiagram voor de feedback zijn nog extra aanpassingen gemaakt, deze worden later in dit document besproken.



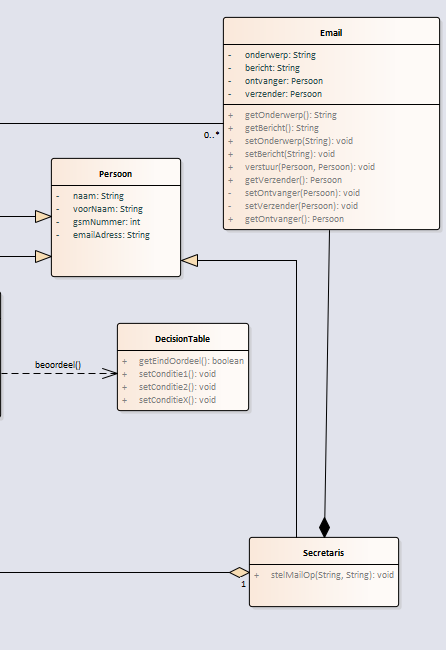
Hierboven staat het volledige klassendiagram dat we gemaakt hebben.



Een Potentiële verhuurder is een persoon die een aanvraag kan indienen.

Een aanvraag heeft een status, en heeft een aantal kenmerken (bijvoorbeeld parkeerplaatsen, grootte…). Een aanvraag gaat over één pand, die de interface IAdres implementeert.

Een screener is een persoon die een aanvraag kan beoordelen.



De screener maakt gebruik van de klasse DecisionTable om een eindoordeel over een aanvraag te vellen.

De secretaris is een persoon die een mail op kan stellen, en die zo contact onderhoudt met de potentiële verhuurder.

# Deel 3

User stories die gevarieerd gedrag van de objecten vragen

**User story**

Als screener wil ik mijn scores van een aanvraag in de decisiontable invoeren, zodat ik er een eindresultaat op krijg. (Goedgekeurd, afgekeurd)

Als screener wil ik een aanvraag a.d.h.v. een decisiontable beoordelen, zodat...

Acceptatietesten

**Feature: A**anvraag van een pand beoordelen

Als screener wil mijn scores van een aanvraag in de decisiontable invoeren, zodat ik er een eindresultaat op krijg. (Goedgekeurd, afgekeurd)

**Background:** Er is een aanvraag ingediend en de aanvraag is nog niet gekeurd; de info was goed ingevoerd en de foto’s waren duidelijk. De screener wil nu de hele aanvraag scoren a.d.h.v. de decision table.

**Given** Er is een aanvraag voor een pand, de info is volledig, de foto’s voledoen aan de vereisten en de screener heeft de decision table ingevuld.

**AND**

De screener is op de pagina waar hij een onafgewerkte aanvraag, scores moet geven a.d.h.v. de decision table.

**Scenario’s:** Aanvraag behaald score van minder dan 23.

**Given:** De screener vult de scores in

**AND** de score wordt berekend door de decision table

**When:** de berekende eindscore is 21

**AND** de aanvraag status is “ongekeurd”

**Then:** het systeem zet de status van de aanvraag op "afgekeurd”.

**Scenario’s:** Aanvraag behaald score van meer dan 23.

**Given:** De screener vult de scores in

**AND** de score wordt berekend door de decision table

**When:** de score is 24

**AND** de aanvraag status is “ongekeurd”

**Then:** het systeem zet de status van de aanvraag op "goedgekeurd”.

**Scenario’s:** Aanvraag behaald score gelijk aan 23.

**Given:** De screener vult de scores in

**AND** de score wordt berekend door de decision table

**When:** de score is 23

**AND** de aanvraag status is “ongekeurd”

**Then:** het systeem zet de status van de aanvraag op "goedgekeurd”.

**Scenario’s:** De screener probeert een al afgekeurde aanvraag te keuren

**Given:** De screener vult de scores in

**AND** de score wordt berekend door de decision table

**When:** de score is 23

**AND** de aanvraag status is “afgekeurd”

**Then:** het systeem geeft een foutmelding “de aanvraag is al gekeurd”

**AND:** de status van de aanvraag is "afgekeurd”.

// Bijvoorbeeld wanneer twee screeners op hetzelfde moment op een aanvraag drukken

**Scenario’s:** De screener probeert een al afgekeurde aanvraag te keuren

**Given:** De screener vult de scores in

**AND** de score wordt berekend door de decision table

**When:** de score is 23

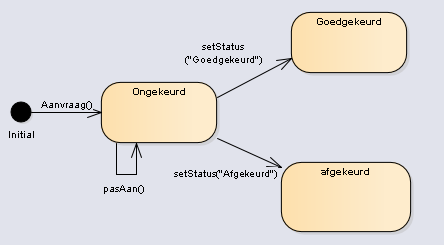
**AND** de aanvraag status is “goedgekeurd”

**Then:** het systeem geeft een foutmelding “de aanvraag is al gekeurd”

**AND:** de status van de aanvraag is "goedgekeurd”.

## Toestandsdiagrammen van de (super)klassen

Aanvraag

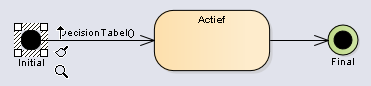


Bij het aanmaken van een aanvraag door de potentiële verhuurder heeft deze de status ongekeurd.

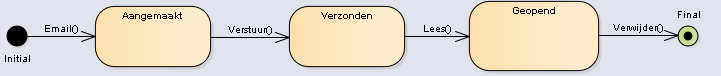
Wanneer de status van een aanvraag op ongekeurd staat, kan op dit object de methode pasAan() uitgevoerd worden. Deze methode staat toe om een aanvraag aan te passen (bijvoorbeeld wanneer een screener heeft ondervonden dat bepaalde gegevens onvolledig ingevuld waren.

Vanuit de status ongekeurd kan de status veranderd worden naar goedgekeurd, of afgekeurd.

DecisionTable



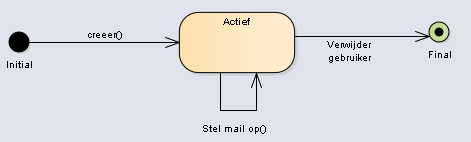
Email



Pand



Potentiele verhuurder

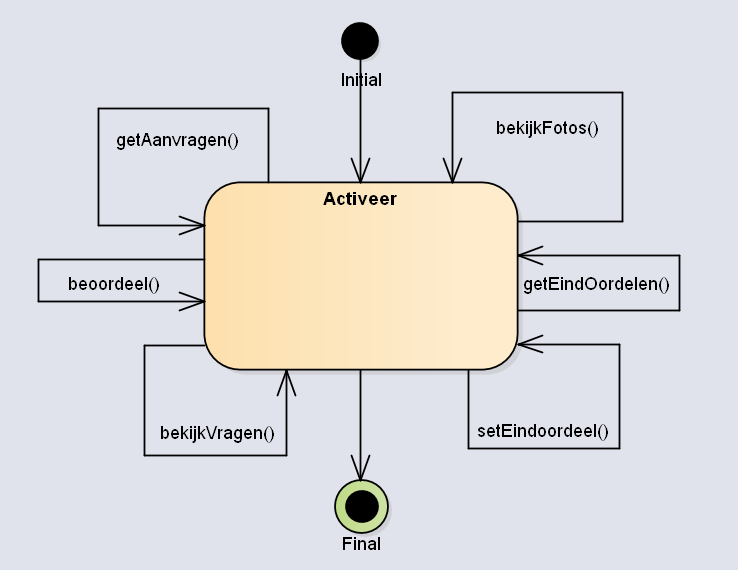


Dit state diagram toont triviaal gedrag aan, al had dit niet zo hoeven zijn. Dit hebben we niet uitgewerkt, maar een potentiele verhuuder zou misschien het aanvraag systeem misbruiken en voortdurend rommel indienen. Dit hadden we kunnen oplossen de gebruiker een status te geven waar als hij bv 3 maal een aanvraag indiend, hij geblacklist wordt. Dit zou dan ook het werk van de screener verlichten.

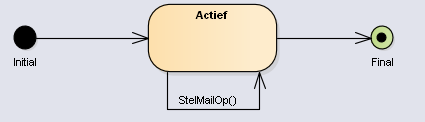
De uitwerking van deze feature valt buiten de scope van het project.

**Nieuwe userstory:** Nieuwe User story. Als screener wil ik een potverhuurder kunnen blacklisten...

Screener

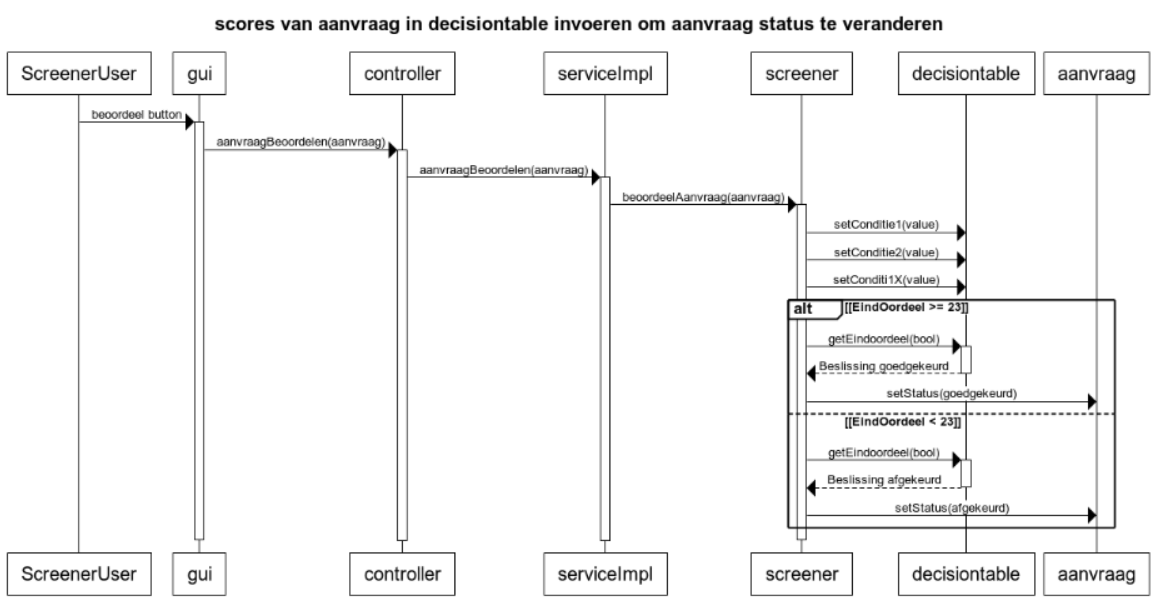


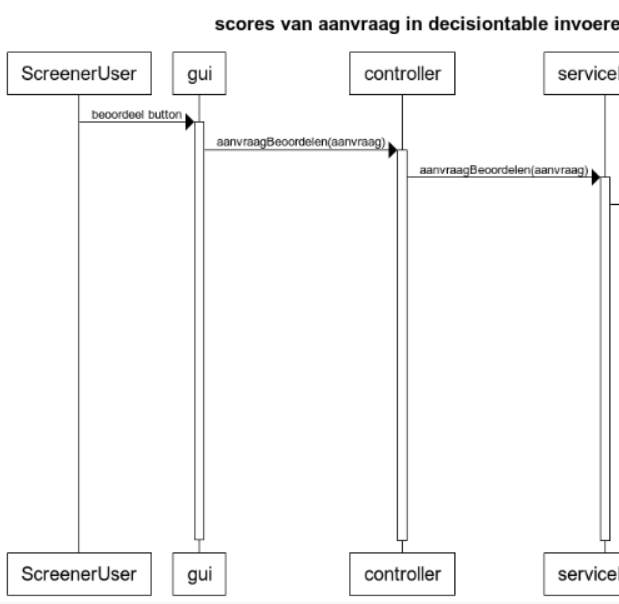
Secretaris



Sequentiediagram

We hebben gekozen om een sequentiediagram te maken voor het verkrijgen van de eindbeslissing rond de aanvraag van een nieuw pand.

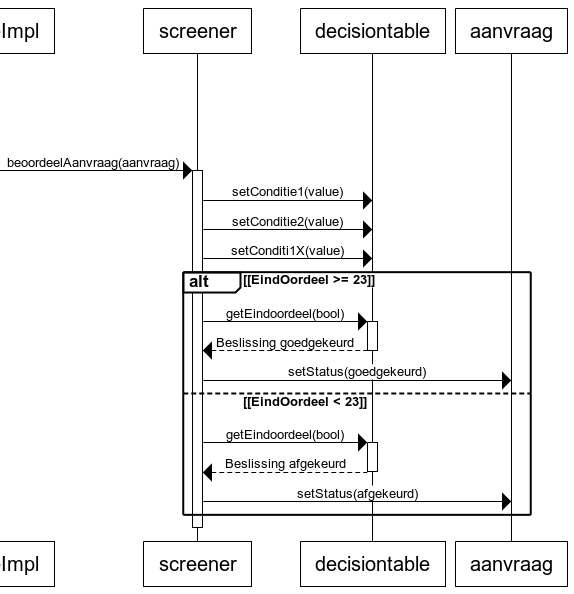




Op het moment dat de gebruiker op de knop beoordeel klikt, wordt de procedure in gang gezet.

Vanuit de gui wordt de controller aangesproken. Die spreekt op zijn beurt de service implementatie aan.

Van hieruit wordt de methode beoordeelAanvraag opgeroepen in de screener klasse.



De screener klasse gaat op zijn beurt de waarde per conditie van deze aanvraag toevoegen in de decisiontable.

Deze decisiontable berekend de totale score van dit pand, en deze wordt opgehaald met de methode getEindoordeel. Deze methode returnt een boolean die true is wanneer de totale score groter of gelijk aan is aan 23, anders is dit false.

Op basis van deze boolean wordt de status van een aanvraag veranderd naar “goedgekeurd” wanneer de boolean true is. Als deze boolean false is wordt de status veranderd naar “afgekeurd".

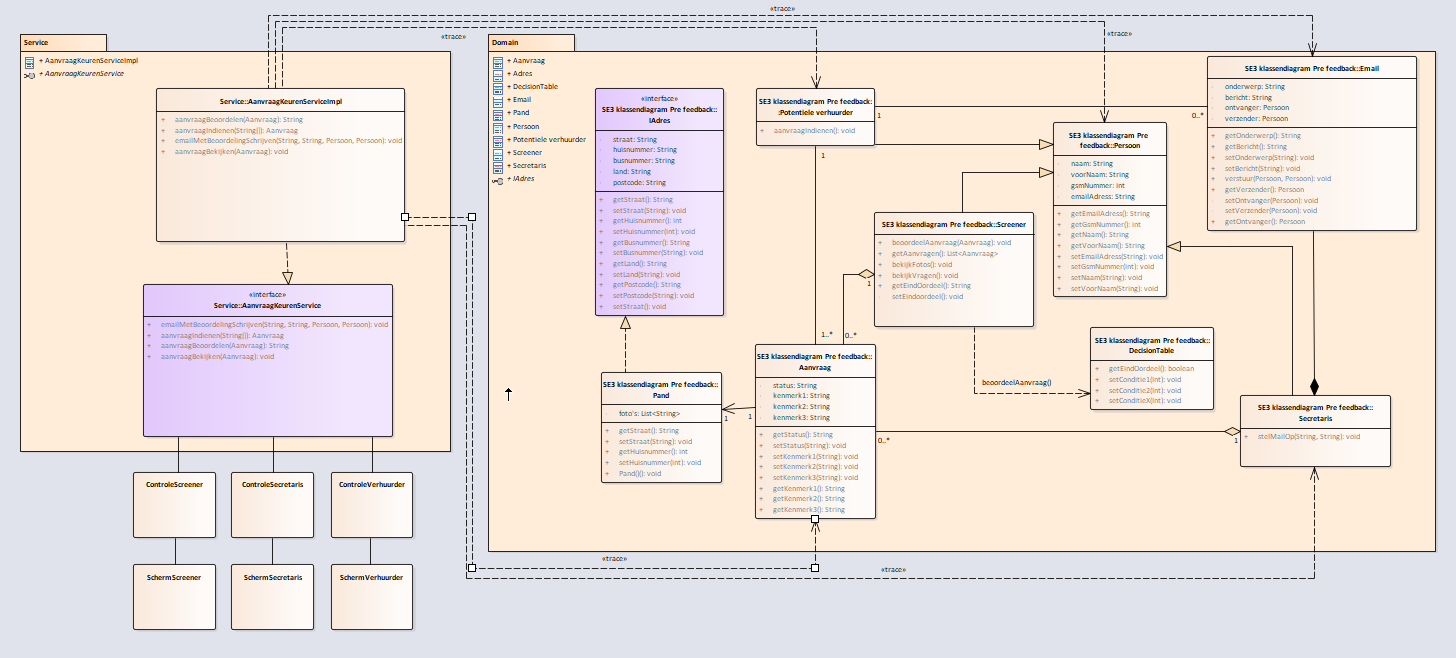
## Code generatie:

Het volledige project met gegenereerde code en unit testen bevindingen zich in de bijlage.

De volgende code stukjes zijn uitegkozen omdat ze een beter inzicht gaven in het project en zorgden voor aanpassingen binnen het project.

**Klasse Screener:**

Origineel:



Bij het programmeren van de methoden werd echter duidelijk dat al deze operaties niet nodig zijn.

* getAanvragen:

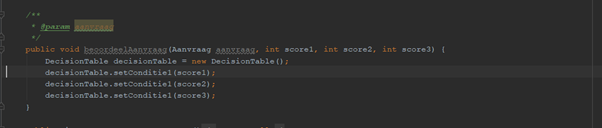
Deze werd oospronkelijk toegevoegd met het idee dat je een lijst van aanvragen nodig hebt om een afspraak te beoordelen. Tijdens het programmeren werd duidelijk dat een afspraak parameter al wordt meegegeven aan “beoordeelAanvraag()” en de lijst niet meer nodig is.

* beoordeelAanvraag():

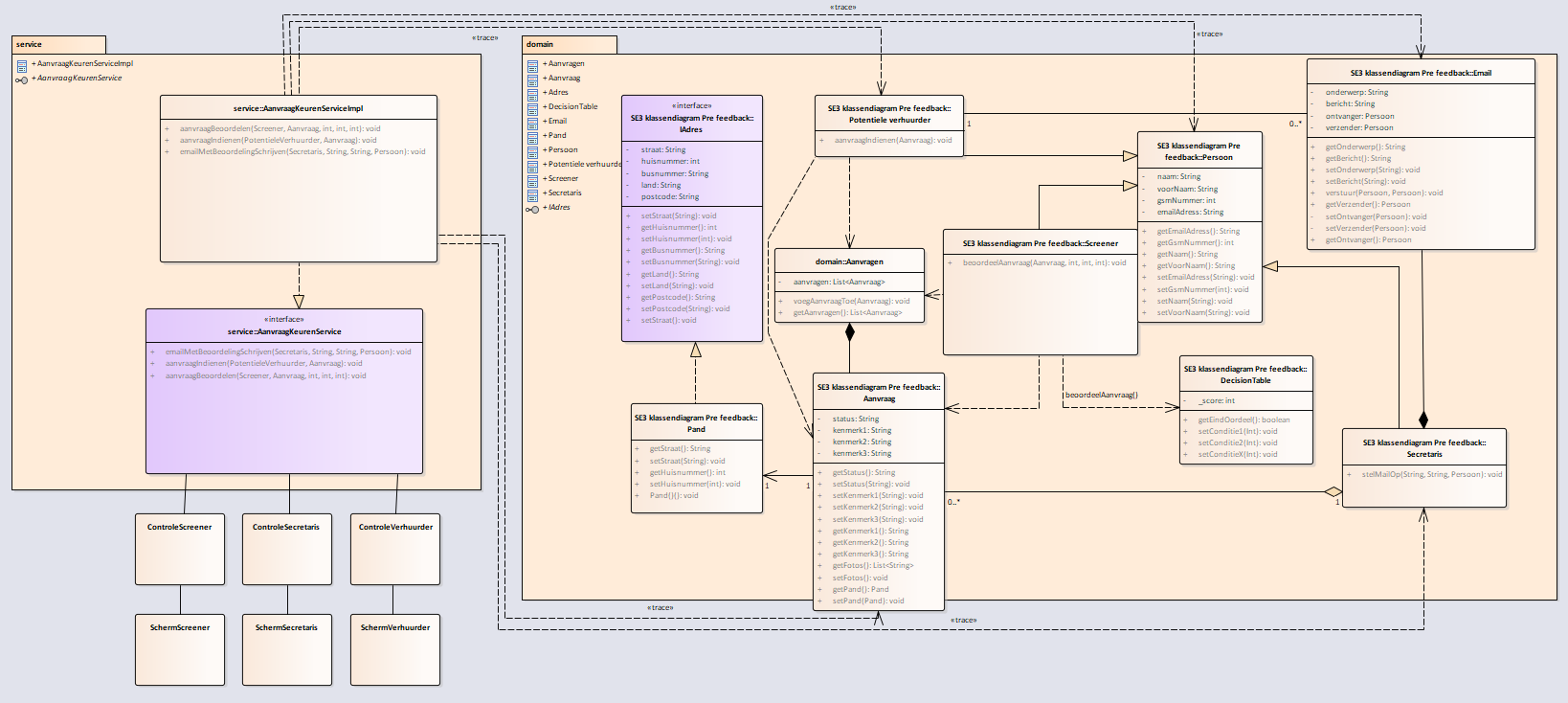
Heeft 3 parameters meegekregen die nodig zijn om de aanvraag ook daadwerkelijk te kunnen beoordelen.

* BekijkFotos() + bekijkVragen():

Is verplaatst naar klasse Aanvraag.



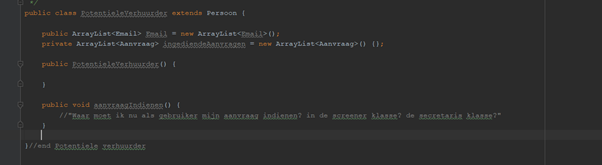
Resultaat:



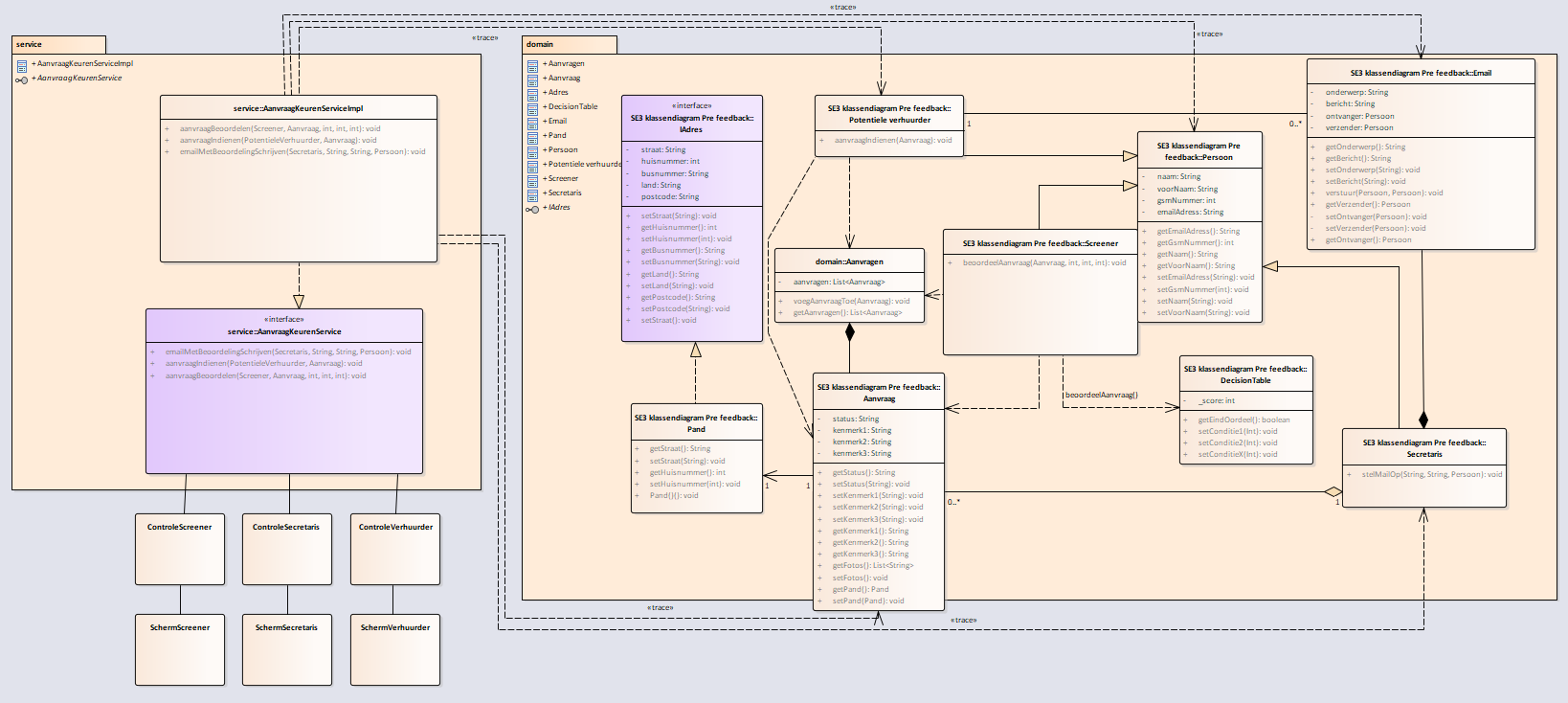
**Klasse aanvragen:**

Als we bij de klasse “PotentieleVerhuuder” de methode “aanvraagIndienen()” wouden schrijven wisten we niet wat met de methode te doen. Moesten we dit dan naar een screener klasse sturen zodat hij dit kon bekijken? En wat als er dan 2 screeners zouden zijn?

Dit hebben we opgelost door een helper klasse toe te voegen, namelijk “Aanvragen” die alle aanvragen van elke potentiele verhuurder bijhoudt.

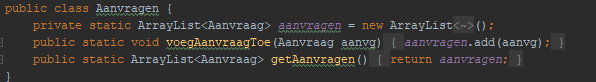


Oplossing:



De Aanvragen klasse heeft nu een compositie relatie met Aanvraag. Screener en Potentiele Verhuuder zullen “Aanvragen” gebruiken in hun operaties.

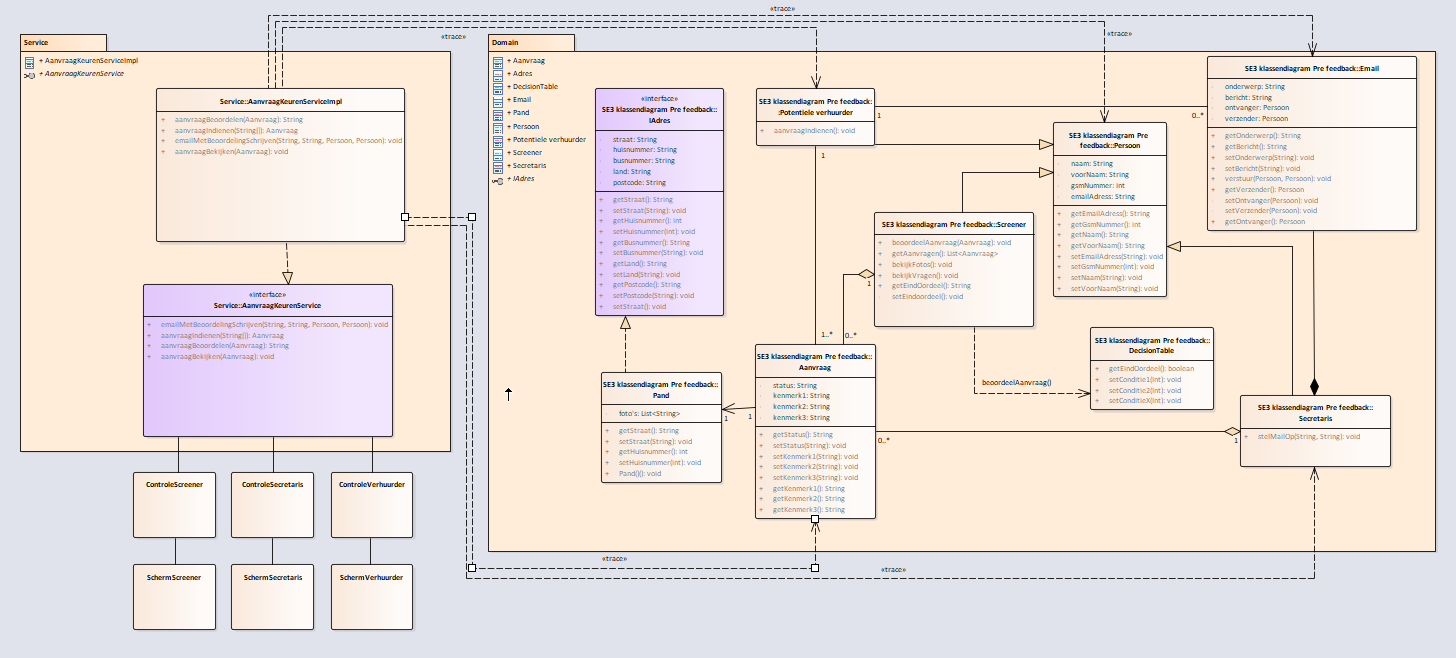




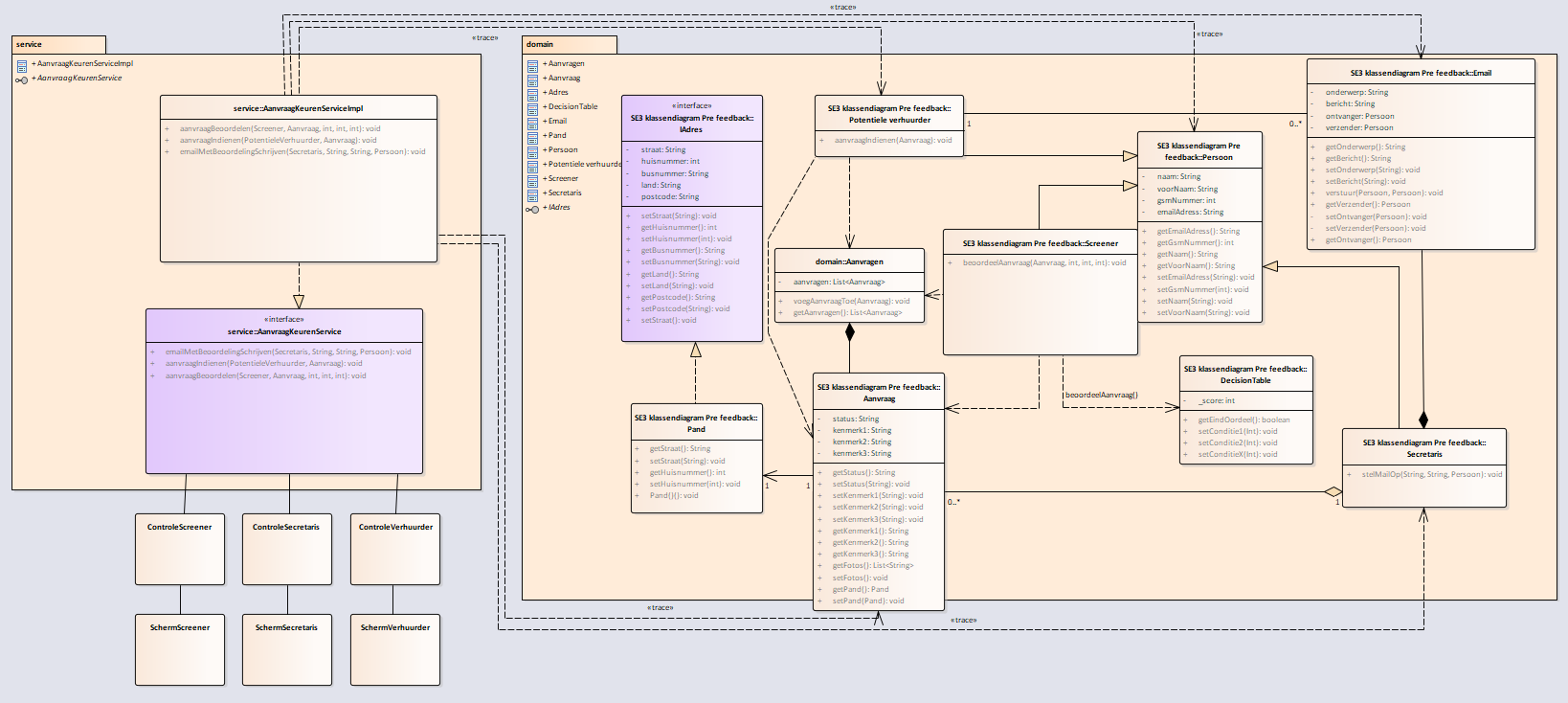
**! Opmerking: de attributen en operaties van “Aanvragen” moeten statisch zijn. (Cursief) !**

**Klasse Aanvraag:**

Origineel:

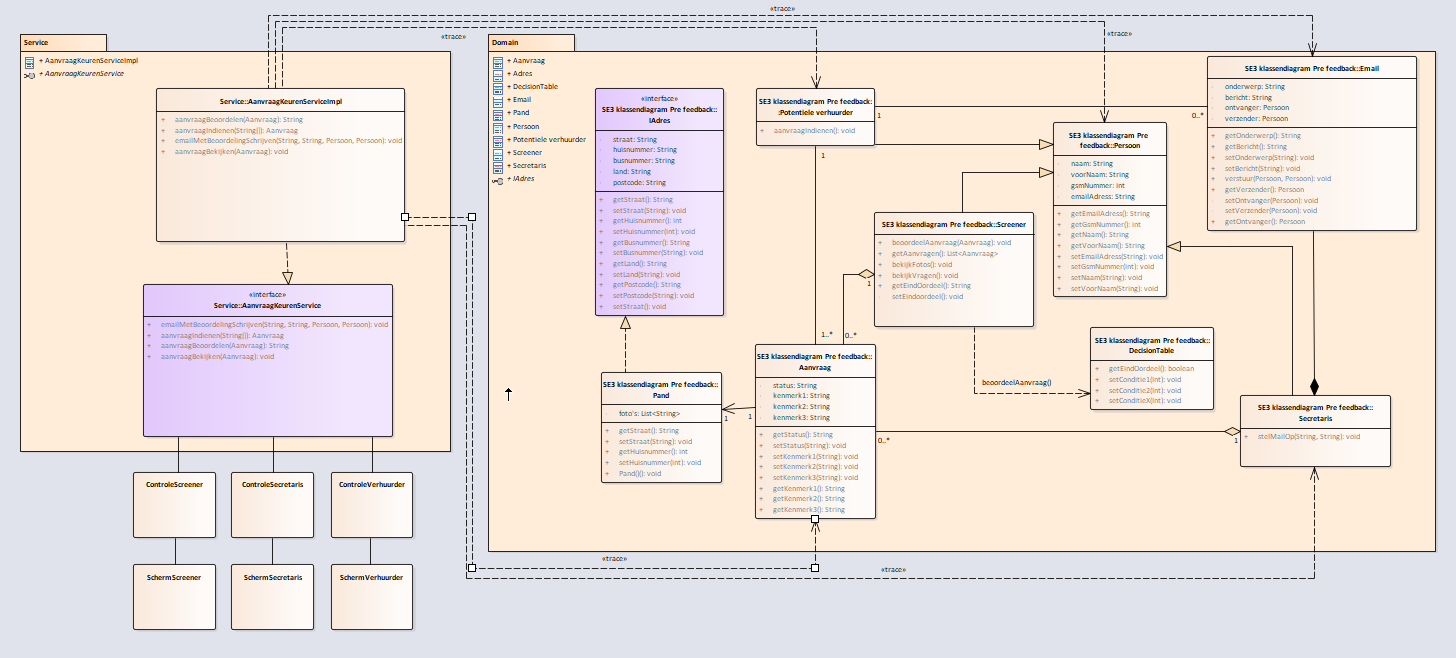


Aangepast in:



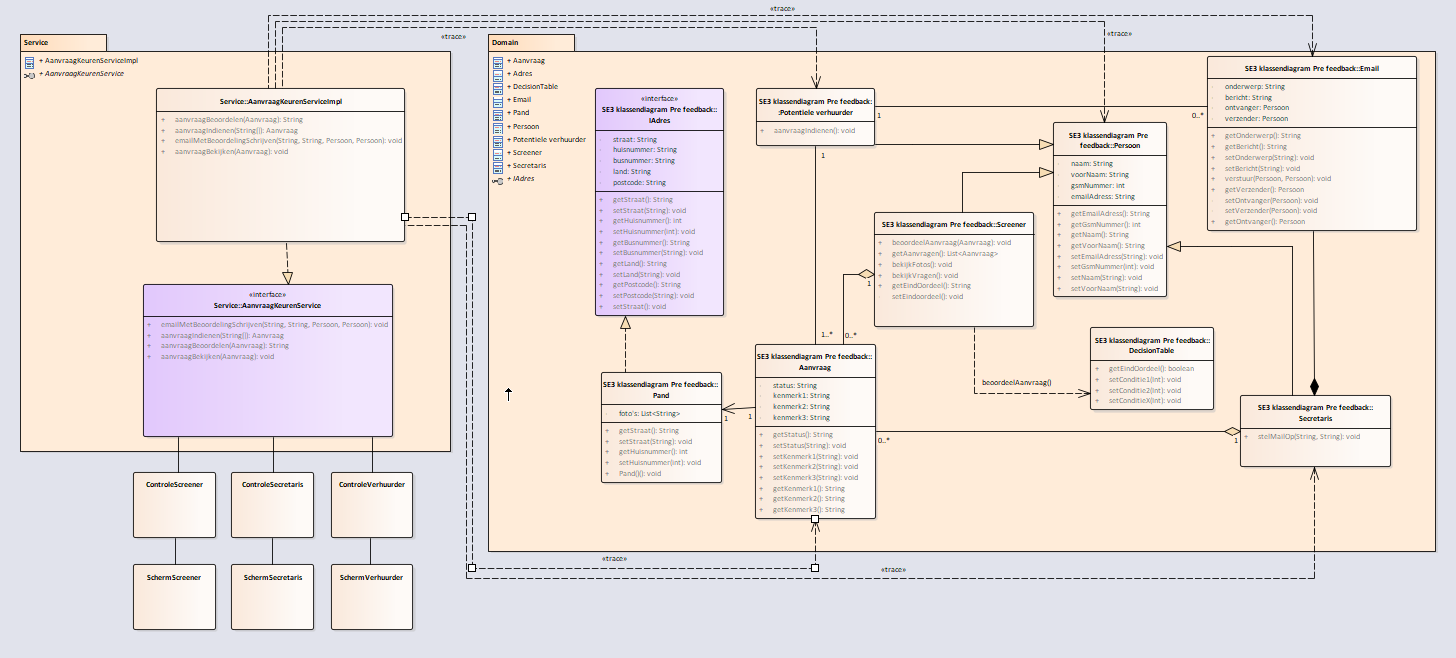
De properties getFotos, setFotos, getPand en setPand zijn hieraan toegevoegd.

**Klasse Pand:**

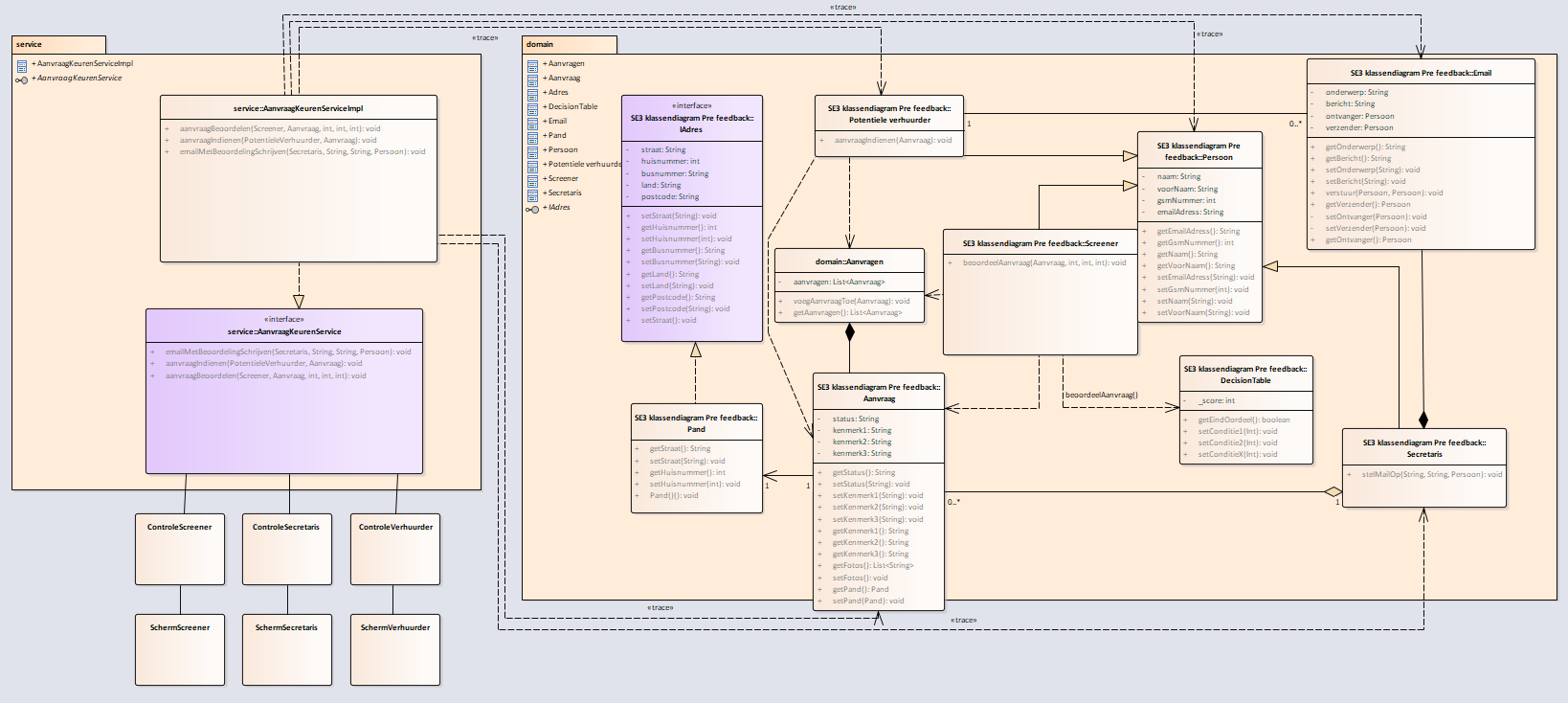


Hier is het attribuut “fotos” verwijderd omdat die al gebruikt werden in “Aanvraag”.

**Klasse Secretaris:**

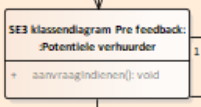
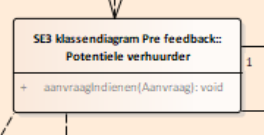


Deze methode mist een attribuut om de verzender in mee te geven, en is hier dus in veranderd:



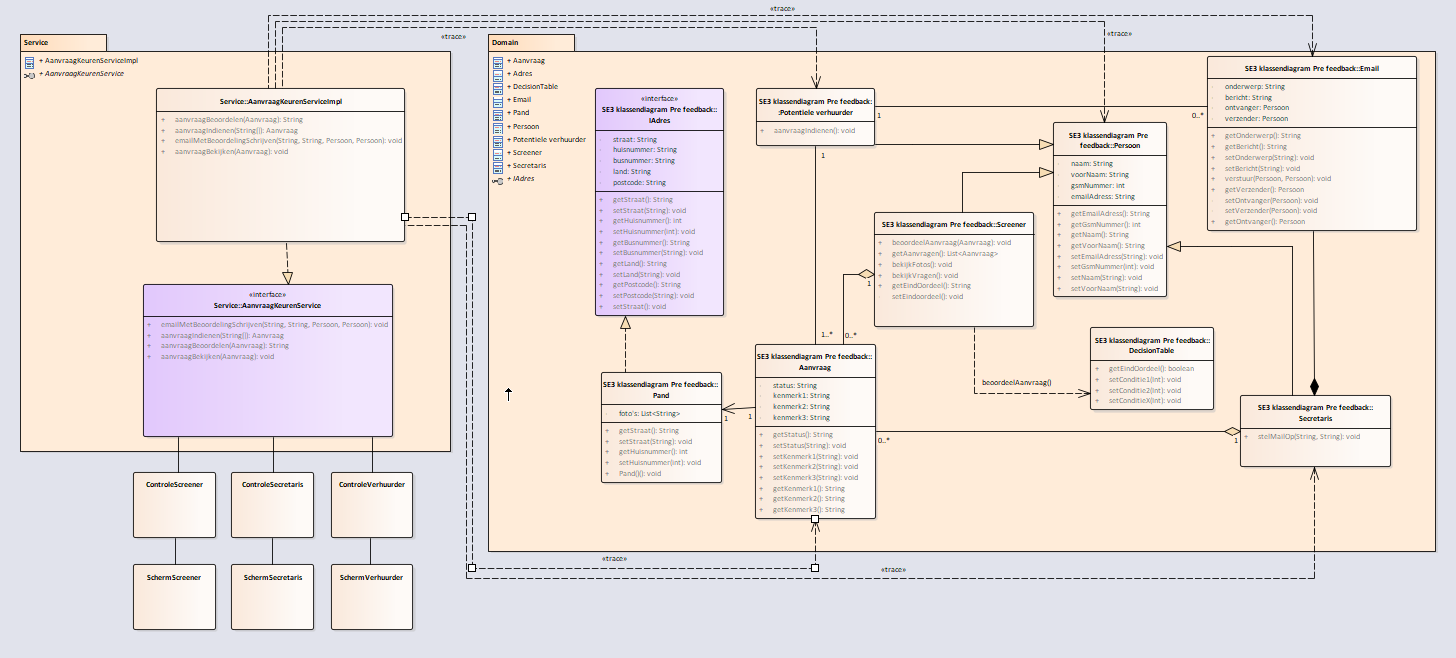
**PotentieleVerhuuder**

Deze methode mist een parameter “Aanvraag”.

🡪 

**Service Klassen:**

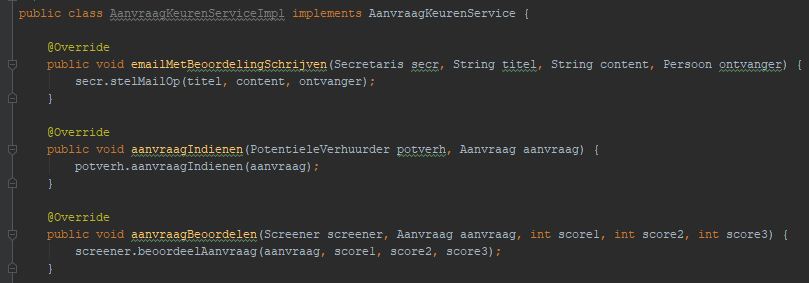
**Interface AanvraagKeurenService**



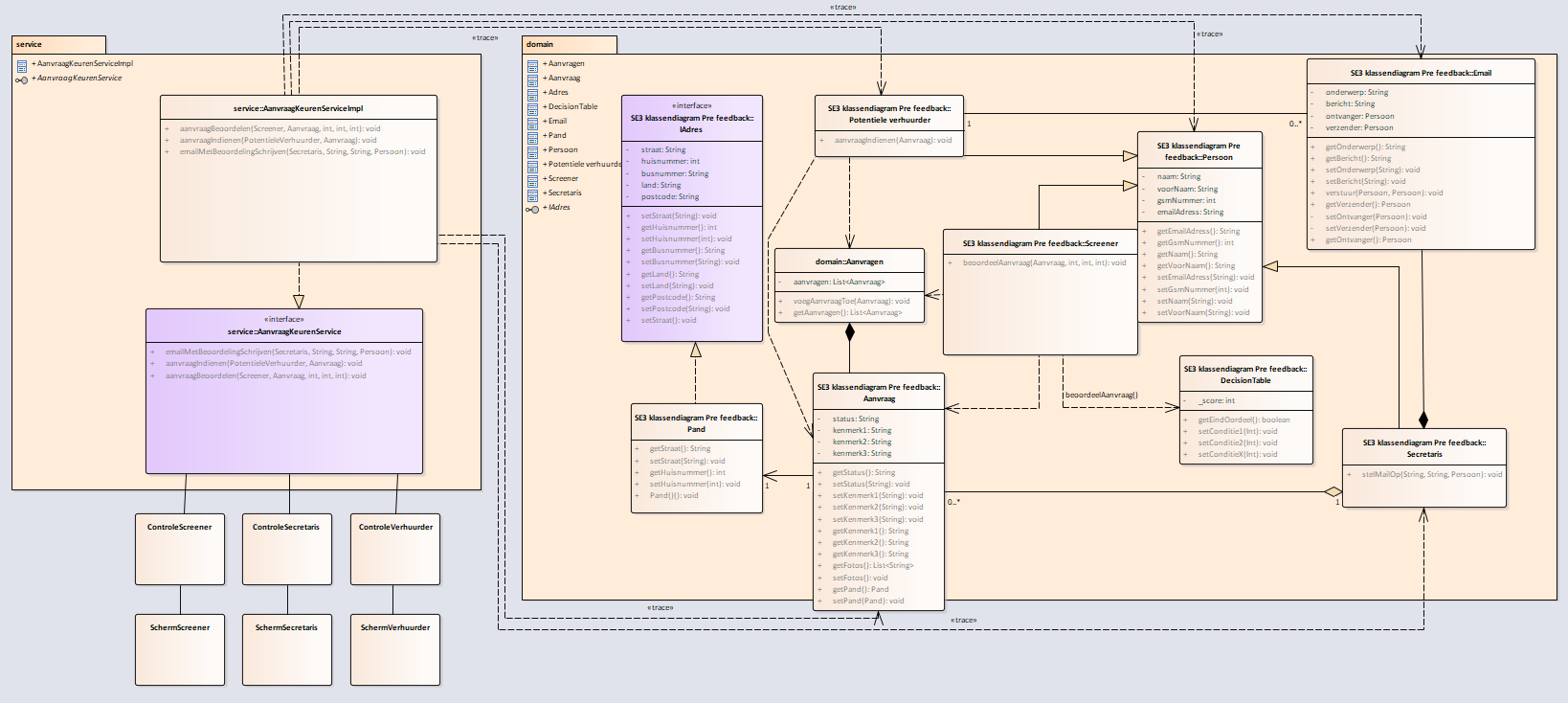
Dit is een onderdeel dat we initieel niet zo goed onder de knie hadden. Nadat de code gegenereerd werd en we hier zelf code in wouden plaatsen werd duidelijk. Dat er toch een aantal dingen incorrect waren.

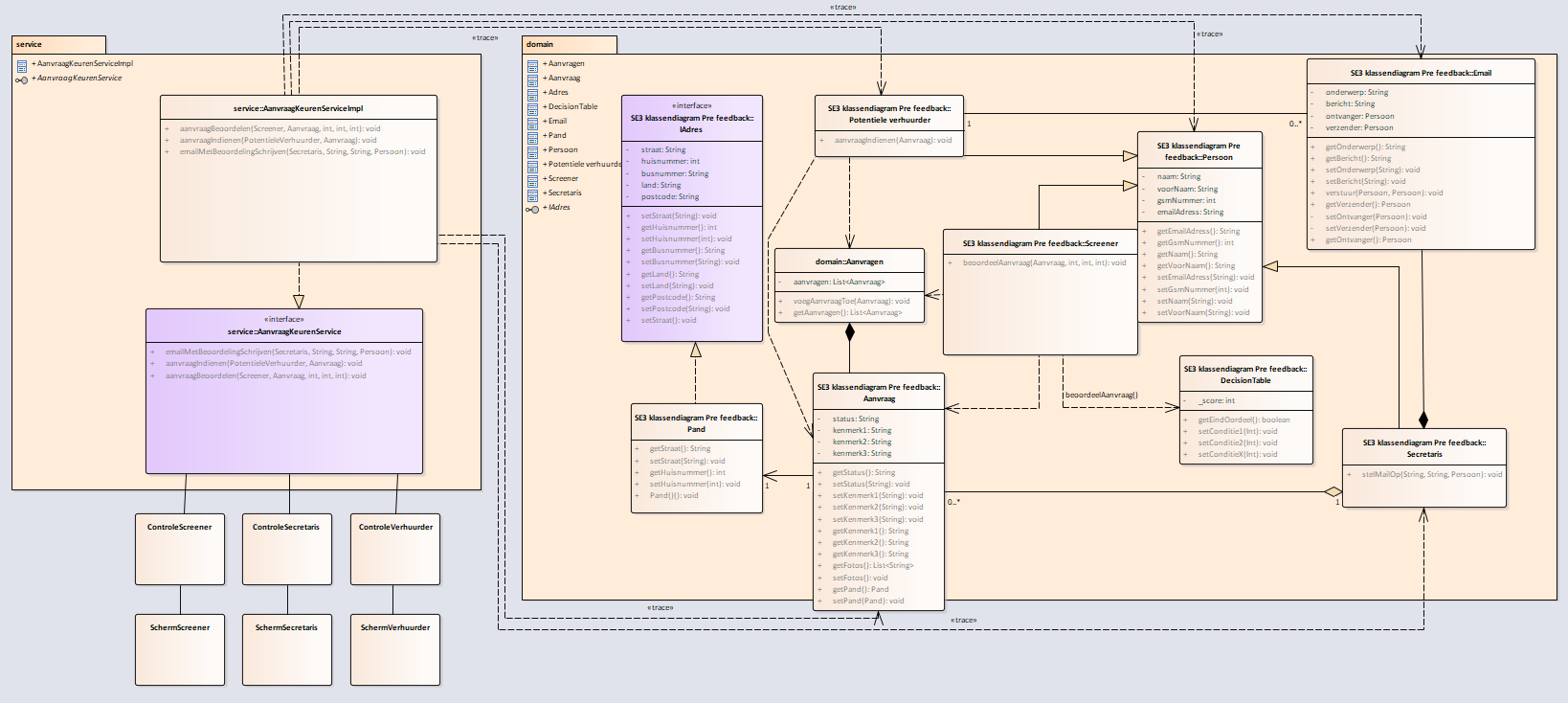
**Aanpassingen:**

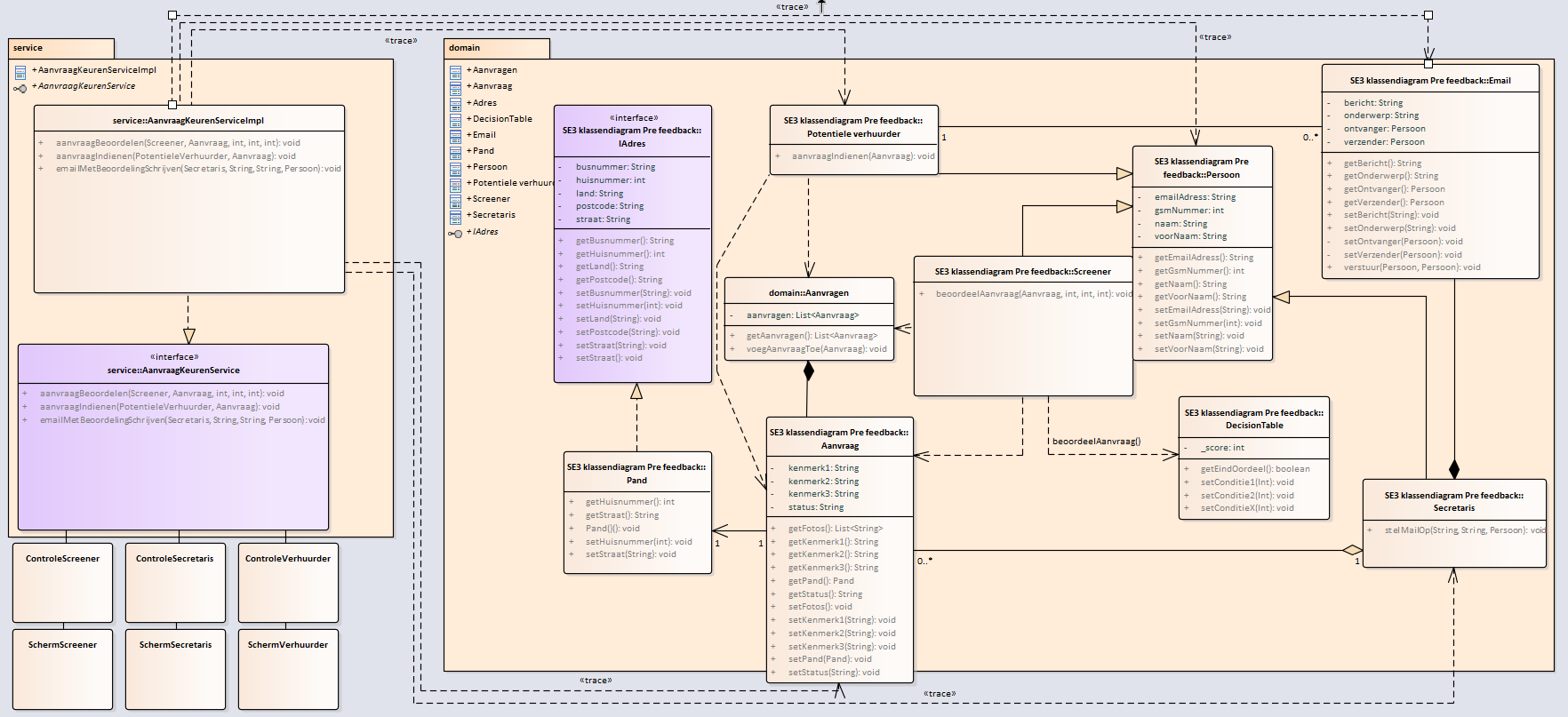
* aanvraagBeoordelen:
  + Extra parameters van wie de aanvraag beoordeeld en wat de ingevoerde scores zijn van de aanvraag.
  + Retourneert void
* aanvraagIndienen:
  + De parameter array verwijderd
  + Nieuwe parameter voor potentiele verhuurder + parameter voor aanvraag.
  + Retourneert nu niets meer
* emailMetBeoordelingSchrijven:
  + parameters gewijzigd
* aanvraagBekijken:
  + methode verwijderd omdat ze niet meer nodig is.



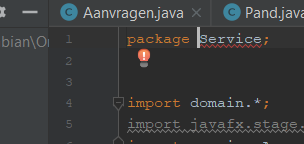
Resultaat:



Final



**Package namen:**  
Ten slotte moesten de package namen nog aangepast worden van Hoofdletters naar kleine letters.



Anders ontstond deze fout.

**Eindresultaat van het klassen diagram:**

## Unit testen:

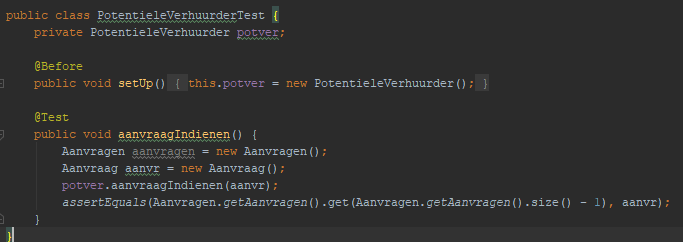
**! Omdat het sequence diagram over het zelfde onderwerp gaat als de al gemaakte “given-when-then” acceptatietest scenario’s, zijn deze niet opnieuw gemaakt. !**

**Er zijn natuurlijk wel unit testen gemaakt.**

Alle unit testen zullen uiteraard niet in de paper geplakt worden. Hier zullen enkel de wat interessante testen te vinden zijn met uitleg. **Voor een volledige lijst van unit testen verwijs ik u naar de bijlage.**

Unit test van PotentieleVerhuurder:

Elke test heeft een setUp methode waar we de te testen klasse aanmaken.

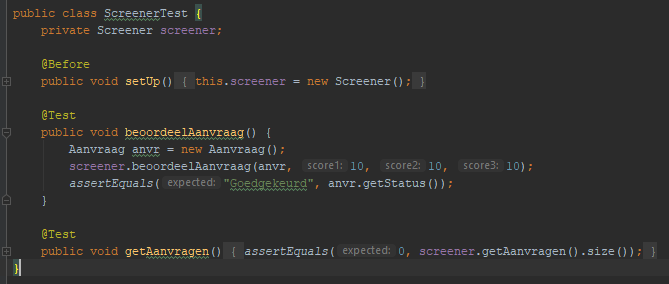


aanvraagIndienen:

Deze test zal een aanvraag aanmaken en vervolgens deze aanvraag indienen in de klasse Aanvragen. Vervolgens wordt gecheckt of de aanvraag ook daadwerkelijk is toegevoegd aan de lijst door naar het laatste element te kijken en te verifiëren of dat dat element gelijk is aan de ingevoerde afspraak.

Dit is een zeer interessante test omdat in deze methode niet alleen de methode van aanvraagIndienen getest worden, maar ook het maken van een nieuwe aanvraag en of de statische getAanvragen in “Aanvragen” werkt.

ScreenerTest



Deze is vooral interessant omdat het een erg belangrijk onderdeel is van het project en de methode beoordeelAanvraag achterliggend nog gebruik maakt van een DecisionTable klasse die een resultaat berekent op de ingevoerde gegevens.

Groepsreflectie

Het merendeel van de groep werkte steeds gemotiveerd mee aan de opdracht. Elke dinsdag spraken we af op discord. Omdat de werkdruk hoger lag in opdracht 2, spraken we nu ook telkens af in het weekend. De takenverdeling liep vlot.