

Architectuurdocument

ACI Rental system

Fontys Hogeschool | Tilburg

Auteur: Job Verwiel, Siebren Kraak, Marco Ketelaars, Myron Antonissen, Angelo Bruggemans
Locatie: Tilburg
Versie: 0.2
Datum: 02 maart 2021

1 Versiebeheer

Versie	Wijziging	Datum
0.1	Opzet document	02-03-2021
0.2	Toevoegen beide database diagrammen. Toevoegen code C4 Code diagrammen	29-03-2021

Inhoud

1	Versiebeheer	2
2	Context	4
3	C4 model.....	5
3.1	C1 system context diagram	5
3.2	C2 container diagram	6
3.3	C3 Component diagram.....	7
3.4	C4 Code diagram	8
4	Persistentie per component.....	11

2 Context

Fontys Academy for Creative Industries (Fontys ACI) is onderdeel van Fontys Hogescholen, dat met ruim 40.000 studenten en ongeveer 4.000 medewerkers de grootste hbo-instelling is in het zuiden van Nederland.

Bij deze opleiding wordt er regelmatig gewerkt met apparatuur die voor de meeste studenten niet te betalen is. Denk hierbij aan professionele camera's, filmapparatuur en andere apparatuur. Daarom heeft ACI een balie waar deze apparatuur geleend kan worden zodat hier gebruik gemaakt van kan worden tijdens de opleiding.

De opdrachtgever van dit project meneer Harmen Kuppens. Binnen ACI is Harmen Kuppens onderwijsassistent en medewerker bij de uitleenbalie. Tijdens dit project zal hij de rol van 'product owner' op zich nemen. Ook zal Jacques de Roij regelmatig bij gesprekken aanwezig zijn en zich bezighouden met dit project.

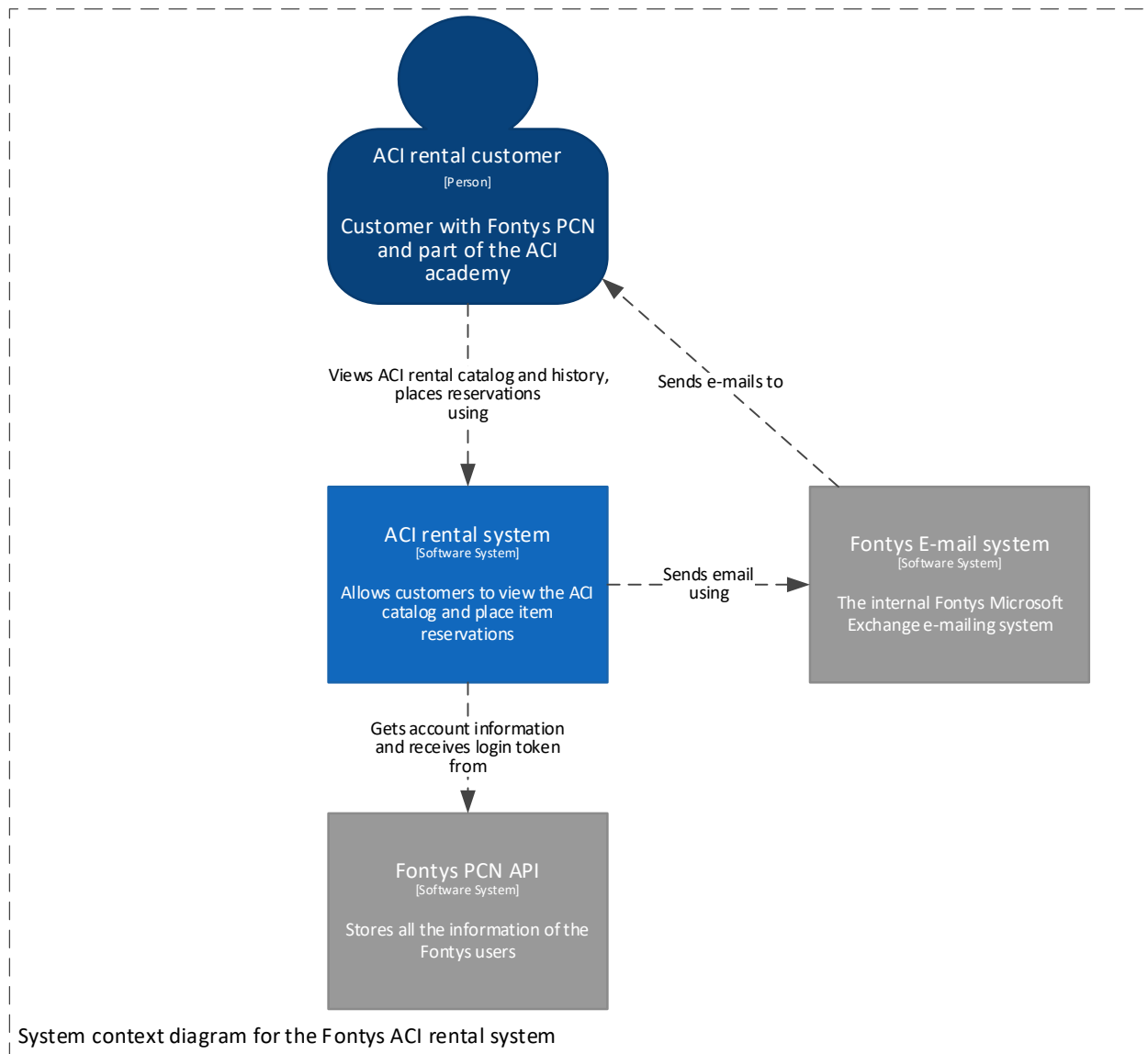
Echter voldoet de huidige oplossing niet aan alle eisen die door ACI worden gesteld. Omdat er meer dan 40.000 studenten en 4.000 medewerkers zijn die potentieel gebruik maken van het nieuwe systeem is het van belang dat de software goed schaalbaar en uitbreidbaar is. Ook is het gewenst dat de nieuwe software niet in één keer de oude software vervangt, maar dat elke sprint een nieuw stuk wordt opgeleverd en geïmplementeerd wordt. Het maken van microservices is met deze requirements onmisbaar.

Het systeem zal features moeten bevatten voor medewerkers en voor studenten. Studenten moeten apparatuur kunnen bekijken en reserveren, oude reserveringen in kunnen zien en boetes kunnen bekijken en betalen. Medewerkers moeten ook apparatuur in kunnen zien, de inventaris wijzigen, reserveringen eerder beëindigen (eerder teruggebracht), notities plaatsen en reserveerhistorie van studenten kunnen bekijken.

Wanneer het over duurdere apparatuur gaat, kan er mogelijk borg of een additionele check op een apparaat zitten. Deze borg moet dan ook betaald kunnen worden via de applicatie en automatisch teruggestort worden. Wanneer er een additionele check nodig is, moet een medewerker goedkeuring geven wanneer een reservering aangevraagd wordt. Dit kan gedaan worden op basis van de reserveerhistorie van de student die de reservering heeft aangevraagd.

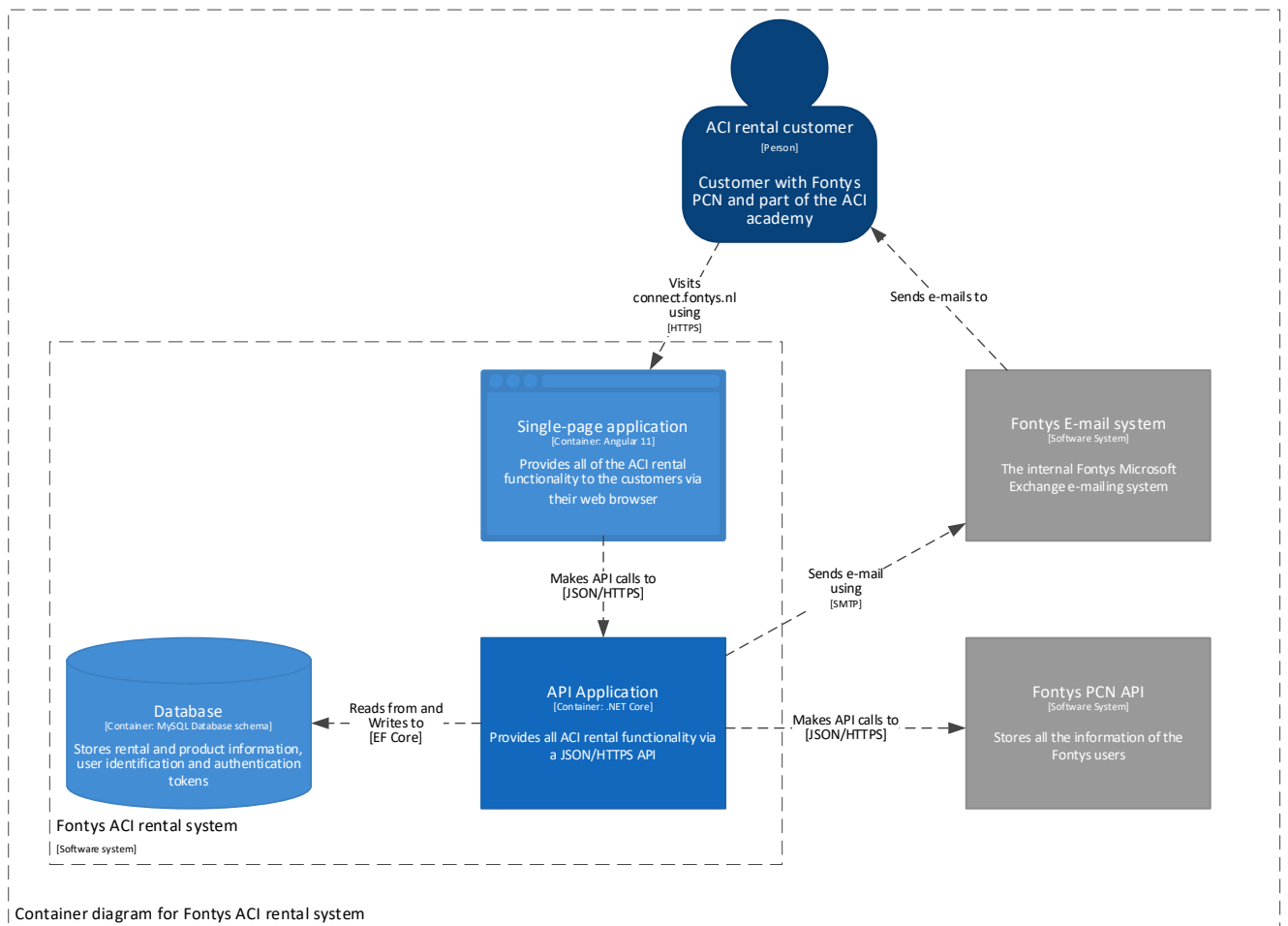
3 C4 model

3.1 C1 system context diagram



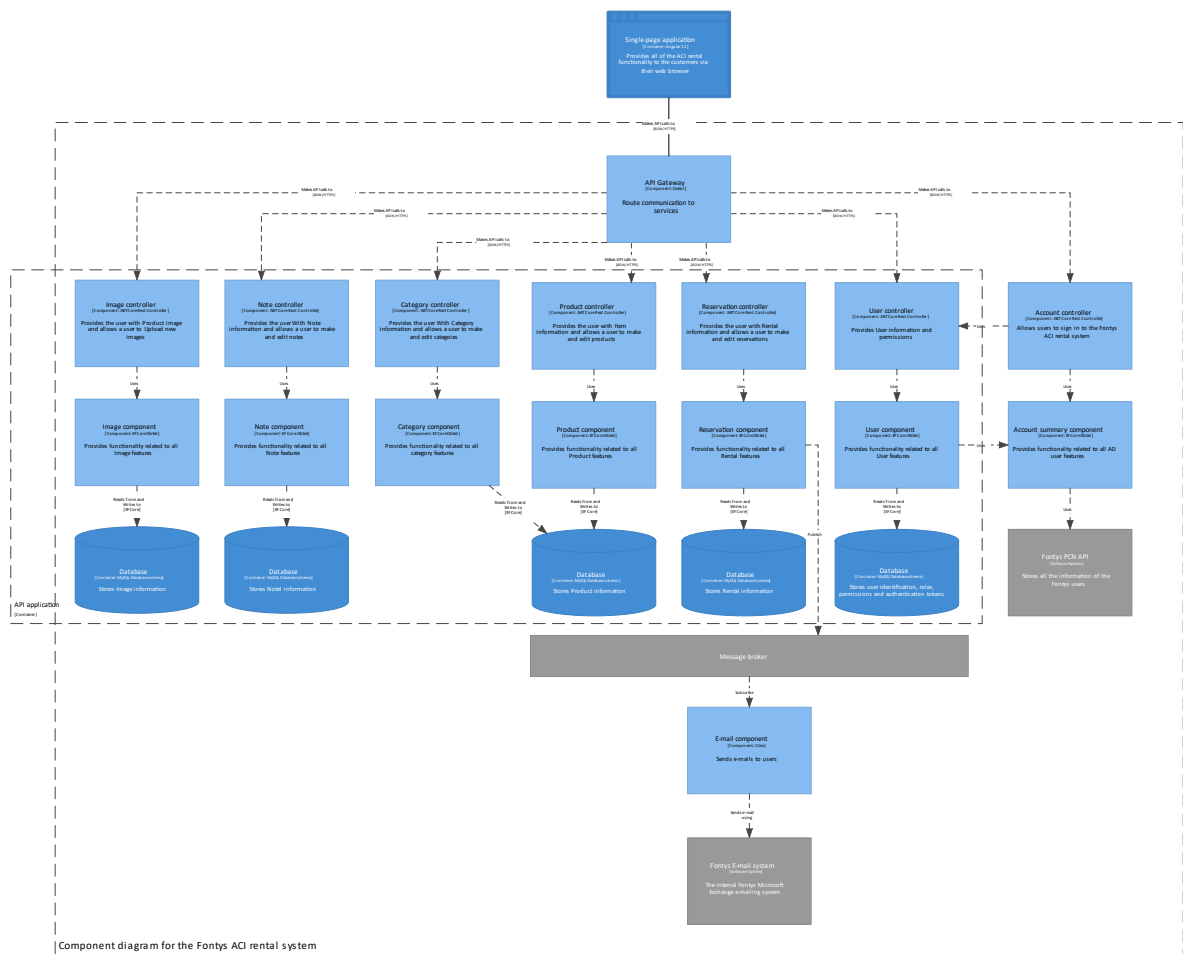
Figuur 1 C1 System context diagram van het Fontys ACI rental systeem

3.2 C2 container diagram



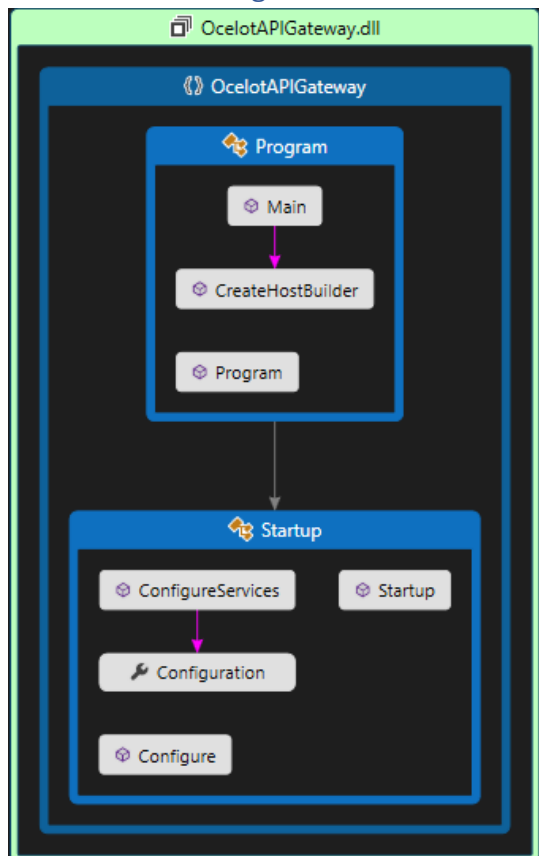
Figuur 2 C2 container diagram voor het ACI Fontys rental systeem

3.3 C3 Component diagram

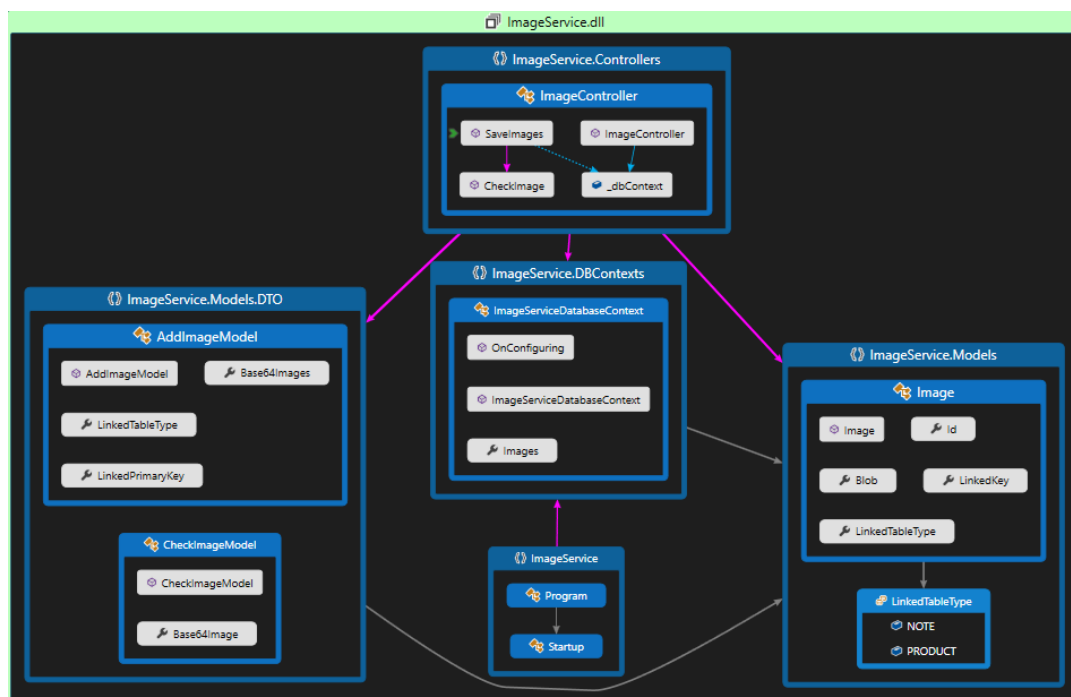


Figuur 3 C3 component diagram voor het ACI Fontys rental systeem

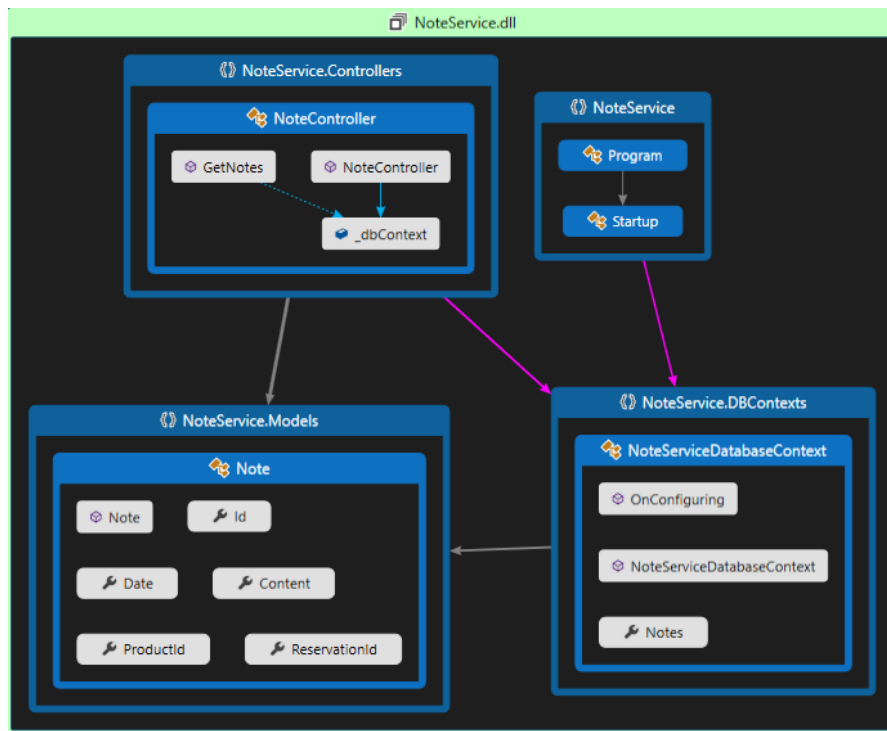
3.4 C4 Code diagram



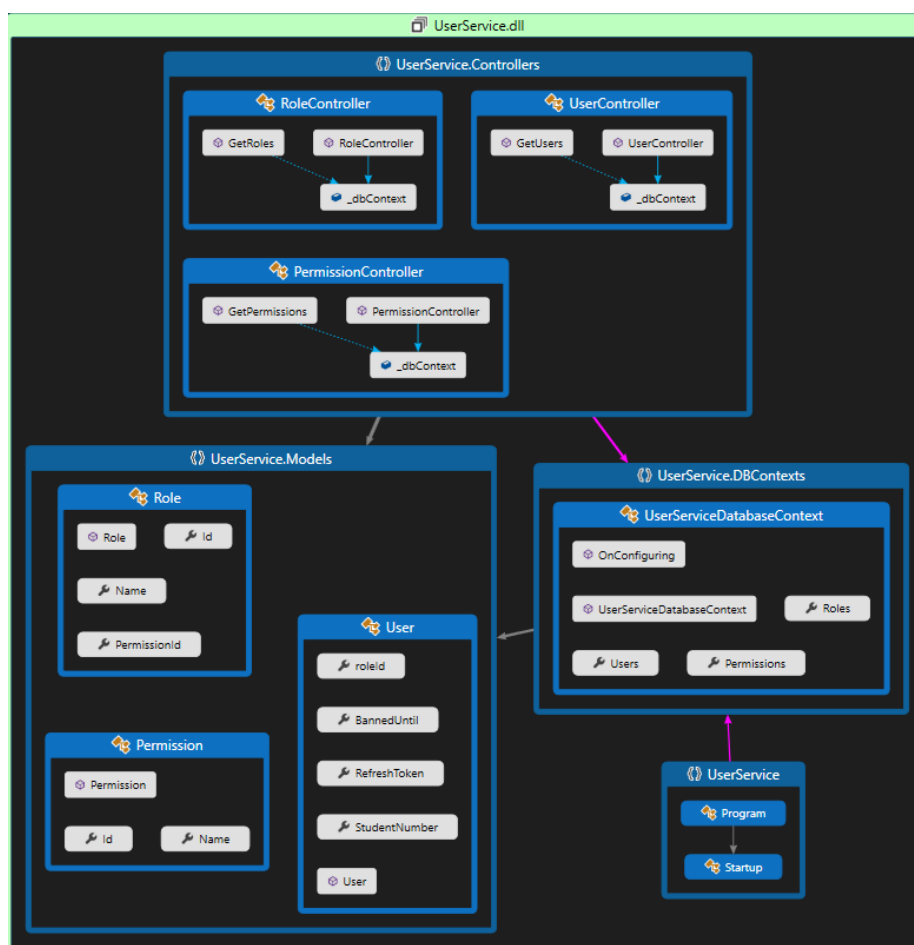
Figuur 4 OcelotAPIGateway.dll



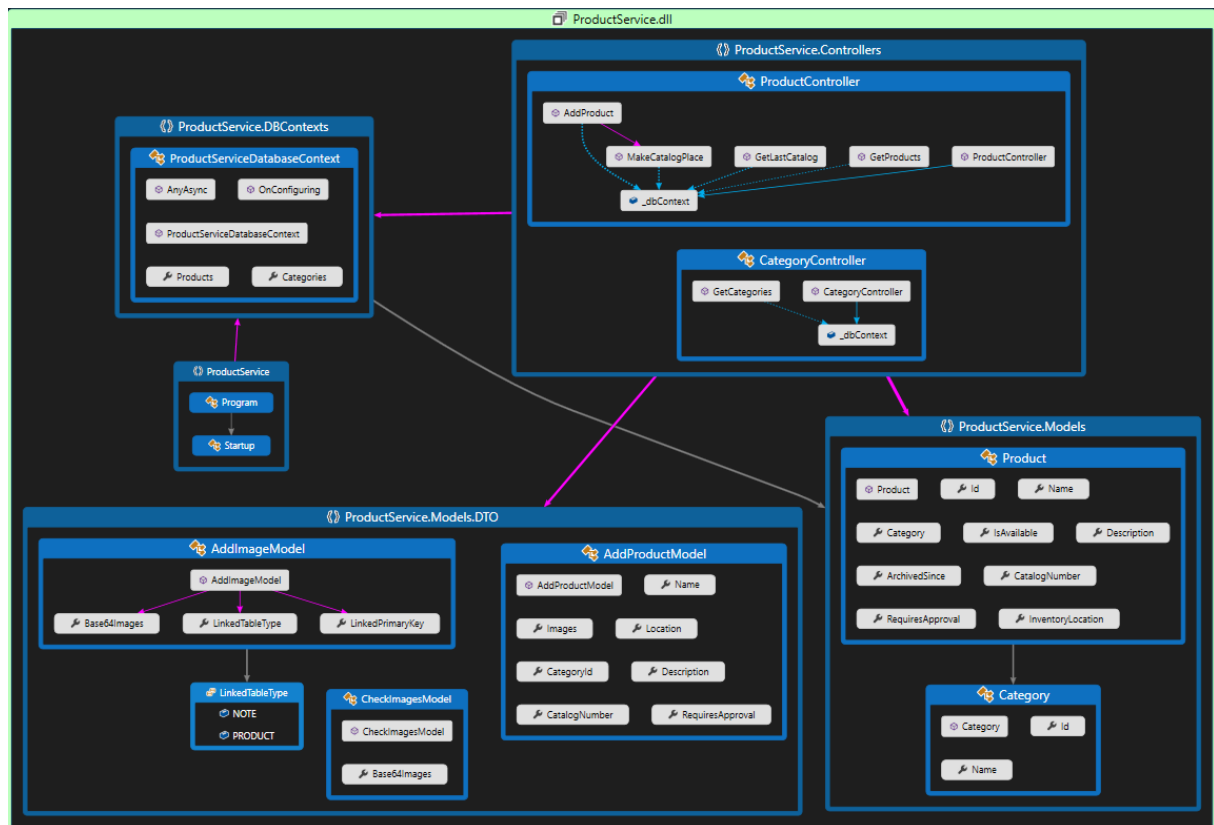
Figuur 5 ImageService.dll



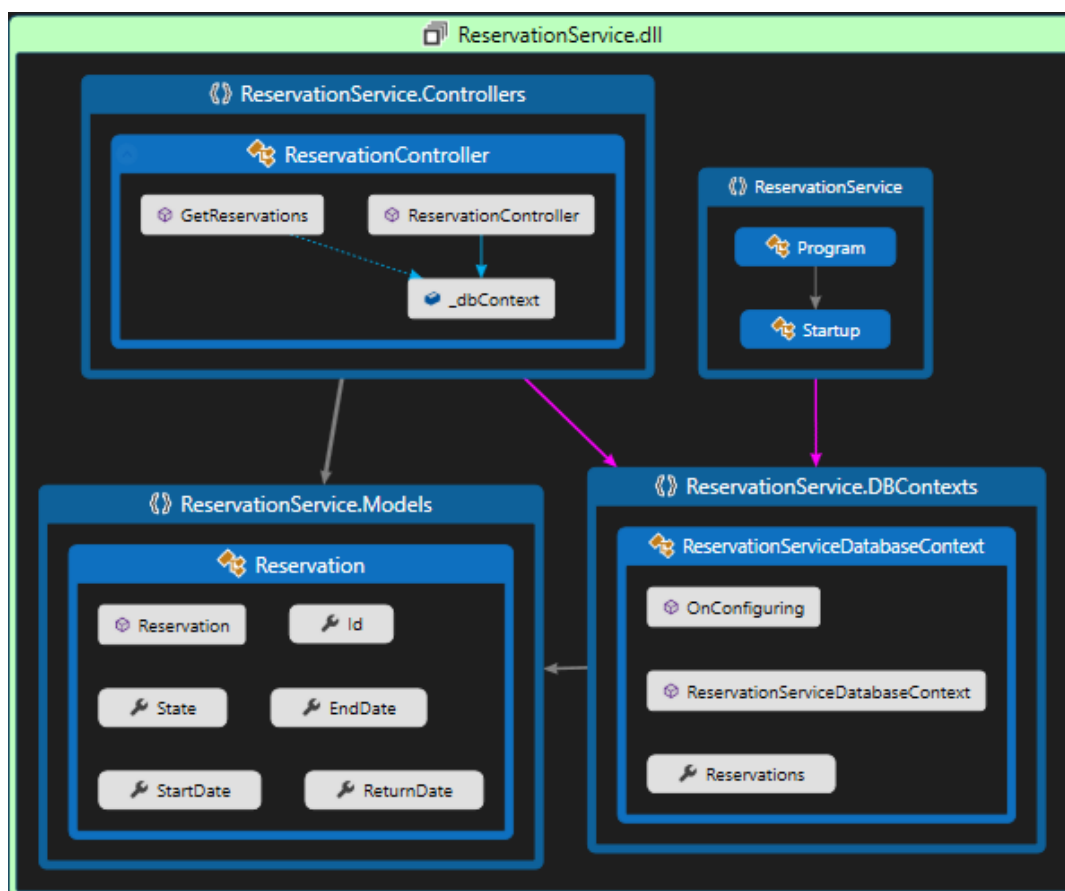
Figuur 6 NoteService.dll



Figuur 7 UserService.dll



Figuur 8 ProductService.dll

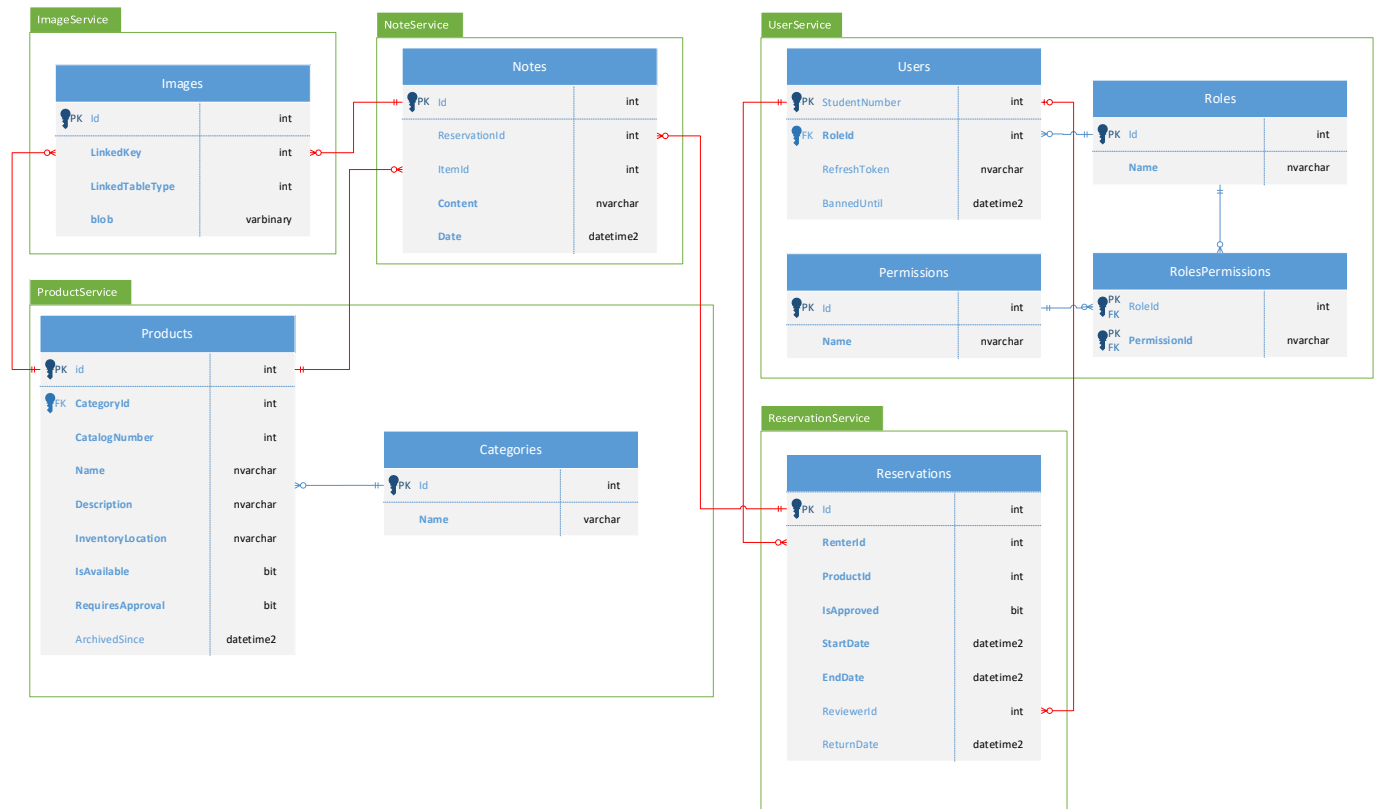


Figuur 9 ReservationService.dll

4 Persistentie per component

Hieronder staat het eindresultaat van de gemaakte database. Deze database wordt automatisch gegenereerd, op basis van de classes, door het Entity Framework. Echter was dit diagram niet het eerste diagram wat getekend werd. Het eerste diagram is te vinden in de OneDrive van het project onder de naam '[Database diagram draft.vsd](#)'.

Omdat de databases zijn opgesplitst, kan er geen gebruik gemaakt worden van echte foreign keys. Om toch in het diagram aan te tonen dat er tabellen gelinkt zijn aan elkaar, zijn er rode relatielijnen gebruikt.



Figuur 10 Database model