

Ontwerpdocument eTransport



Studentnaam: Owen de Bree

Studentnummer: I461496

Klas: S2-DB01

Datum: 26-03-2021

Inhoudsopgave

Projectbeschrijving	3
Hoofddoel	3
Belangrijkste gebruikers.....	3
Vergelijkbare applicaties.....	3
Klassendiagram	4
Database ontwerp	5
Architectuur.....	6

Projectbeschrijving

Hieronder heb ik kort in de verschillende kopjes toegelicht wat de focus van mijn project is en wat ik wil bereiken met het maken van deze applicatie.

Hoofddoel

Het hoofddoel van deze applicatie is het toegankelijker maken van speciaal transport voor zowel de particulieren als de bedrijven. Dit ga ik doen door zowel openstaand transport van locatie tot locatie beschikbaar te maken (ridesharing) en het afhuren van privé transport. Ik ga me aan de sensible default houden voor de verschillende keuzes van hoe ik het project ga realiseren. Dit betekent dat ik gebruik ga maken van asp.net core, me ga houden aan de principes van SOLID, gebruik ga maken van UML klassendiagrammen, git gebruik voor versiebeheer en een SQL database gebruik om gegevens op te slaan. Verder ga ik voor mijn front end gebruik maken van het Blazor framework zodat ik de javascript interactie uit mijn project kan halen en kan vervangen met C#.

Belangrijkste gebruikers

Er zullen hoogst waarschijnlijk 2 rollen beschikbaar worden binnen de applicatie: Gebruikers en de Administrator(s)

De gebruikers hebben de mogelijkheid om zich in te boeken op al bestaande transporten of kunnen een nieuw transport aanvragen.

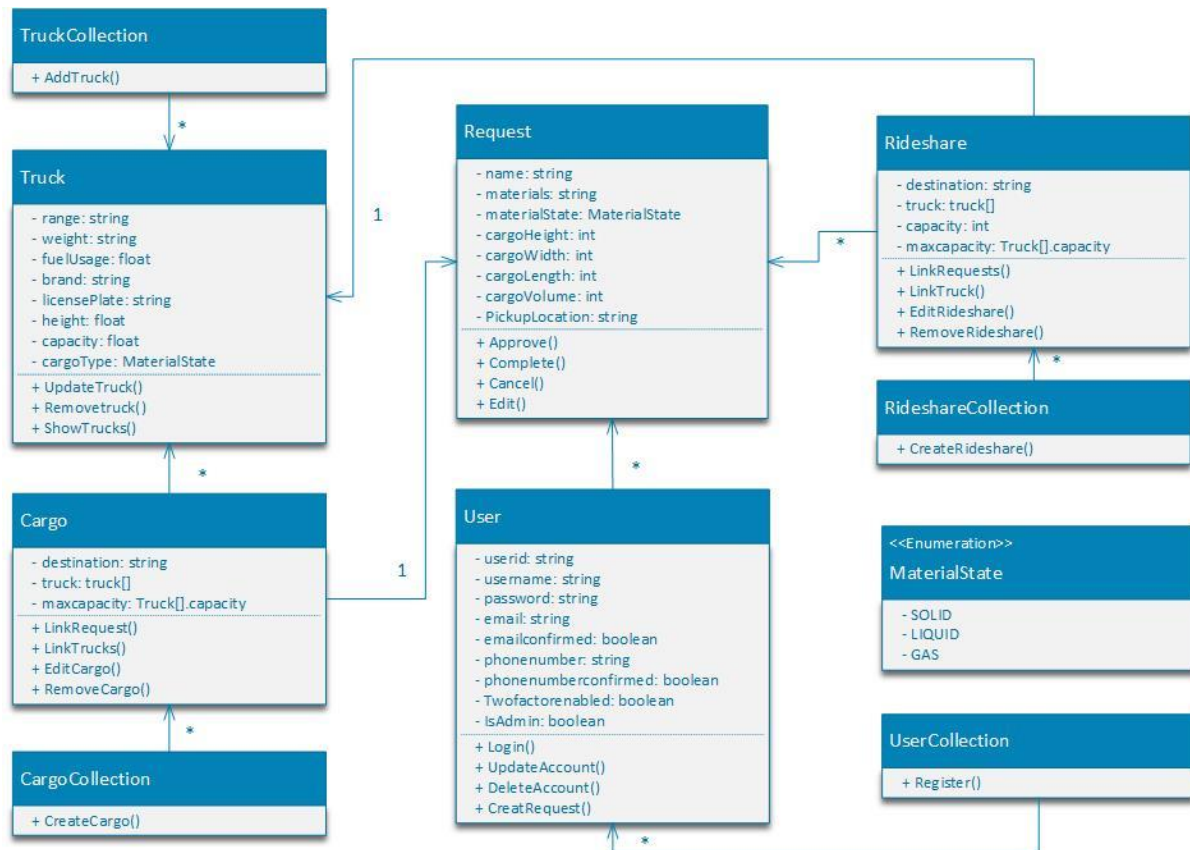
De administrator kan accounts beheren, routes/aanvragen goedkeuren, nieuwe rideshare routes maken, inboekingen op rideshare goedkeuren en nieuwe transporttypes toevoegen aan de applicatie.

Vergelijkbare applicaties

Er zijn op dit moment geen duidelijk vergelijkbare applicaties voor het huren van speciaal transport. In de meeste gevallen is speciaal transport namelijk alleen te huren door het invullen van een online offerte.

Klassendiagram

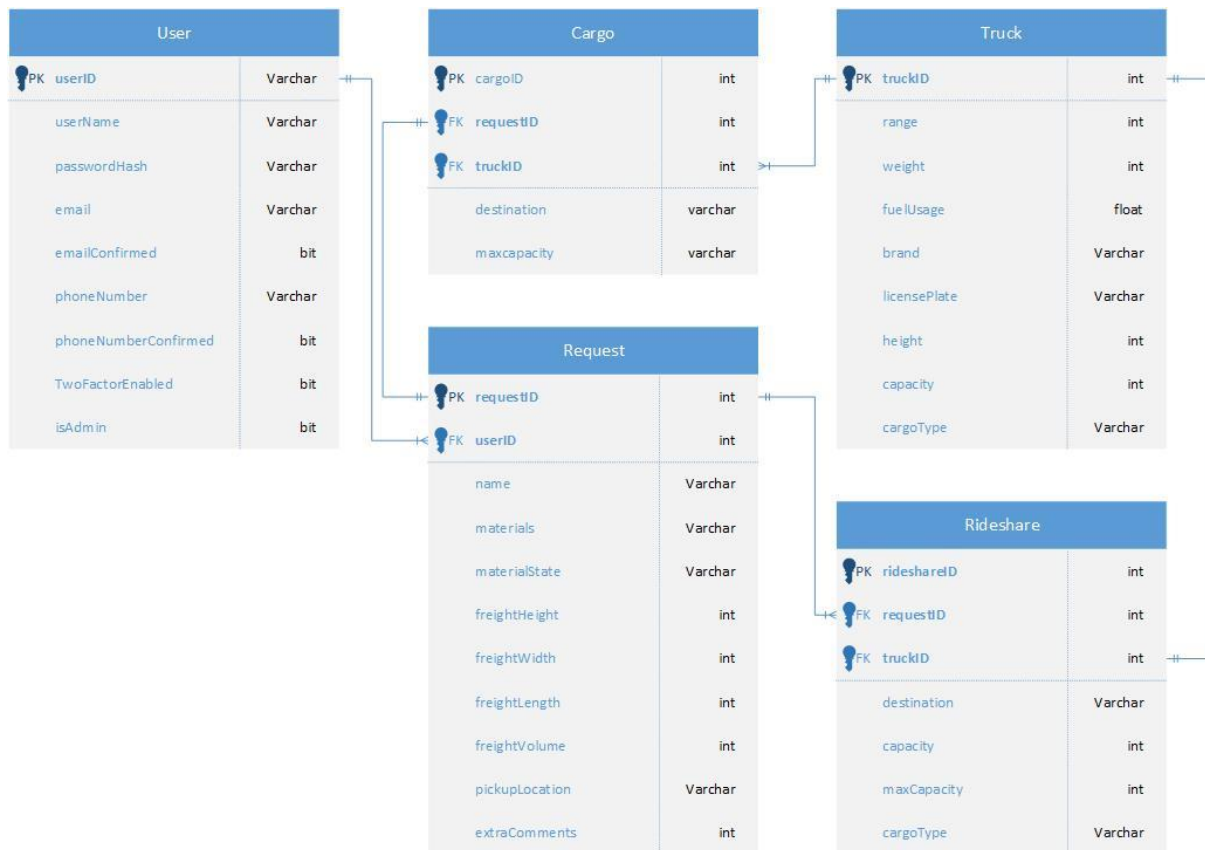
Hieronder staat mijn laatste versie van mijn klassendiagram met alleen de logica laag.



Alle collections zijn voor het toevoegen van nieuwe objecten van de bijbehorende klassen. Elke user heeft de mogelijkheid om vrachtoffertes aan te vragen en deel te nemen aan rideshares. Ik heb alleen een admin als actor nodig en niet als entiteit aangezien ik geen extra informatie over admins opslaat, vandaar dat er alleen een boolean met `isAdmin` in het diagram staat.

Database ontwerp

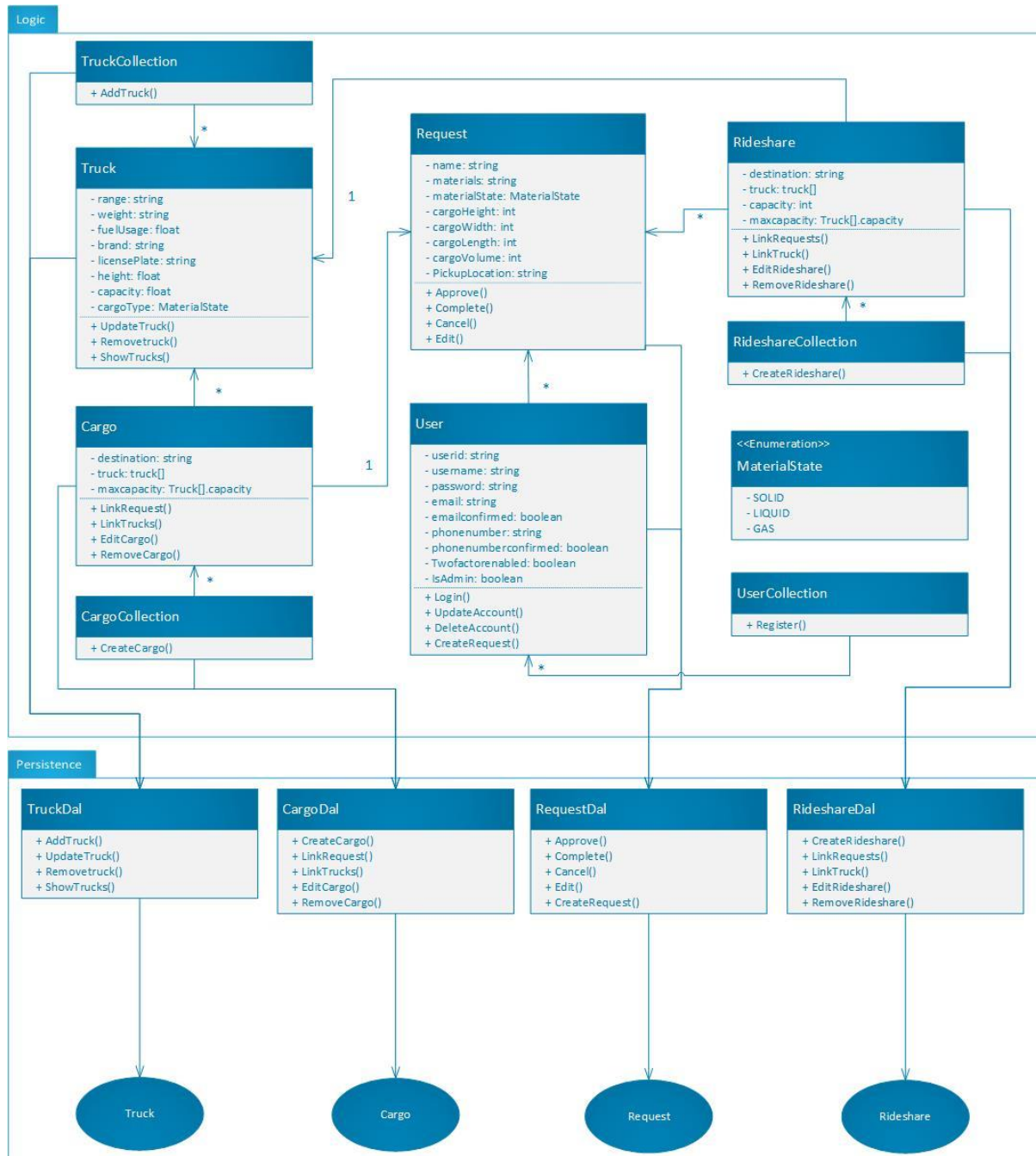
In het onderstaande ontwerp zijn mijn database relaties te vinden met alle bijbehorende typen.



Ik heb ervoor gekozen om de admin als boolean toe te voegen aan de user table aangezien er geen andere entiteiten nodig zijn waar ik een rollen systeem voor zou moeten maken.

Architectuur

Hieronder staat de eerste versie van mijn architectuur met de tot nu toe behandelde informatie hierover uit de workshops. Dit betekent dat alleen de persistentie laag en logica laag aan bod komen tot nu toe.



Ik heb op dit moment, conform de dingen die in de workshop zijn uitgelegd, 2 lagen van mijn architectuur gemaakt. Dit zijn de persistentie en logica laag. Ik heb de dalclass die ingebouwd zitten bij de user hoort er uitgelaten aangezien hier een aantal extra dingen in staan die niet van belang zijn op het moment.