OSM-Modeling Casus

Tim Klaassen(568583) & Jochem Grootherder(598765)

Hogeschool van Arnhem en Nijmegen

OSM-M Richard Holleman

01-11-2018

Inhoud

[InhoudsOpgave 1](#_Toc528751183)

# Inleiding

In dit document staan alle gemaakt opdrachten voor de casus van OSM-M.

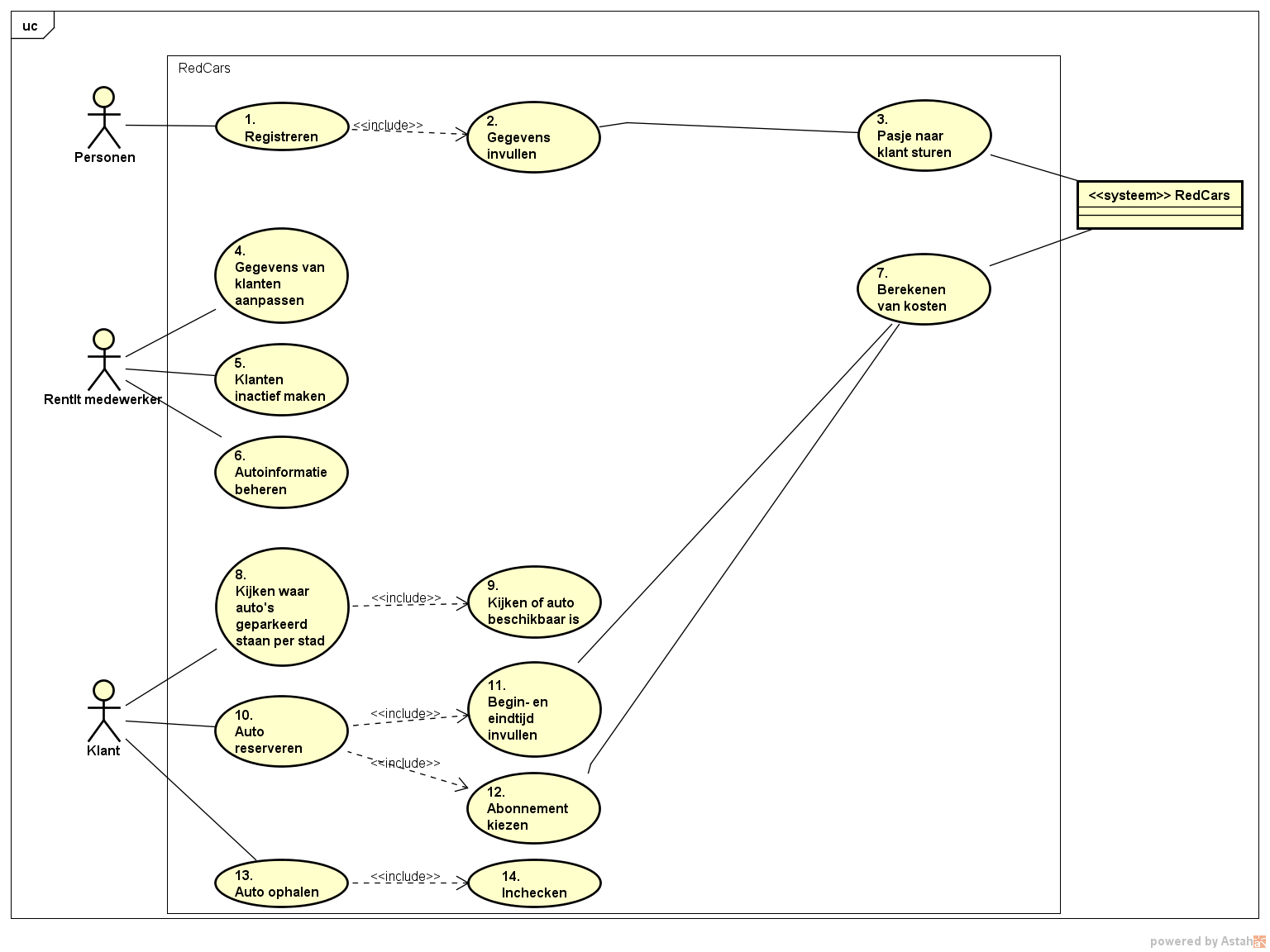
Bij alle gemaakte opdrachten staat een beknopte uitleg over wat het doel van de opdracht is en in welke vorm deze gemaakt is.

Na het lezen van dit document is de lezer bekend met alle gemaakte opdrachten en het beschreven systeem in de casus en zijn werking. Voor het systeem zijn extra vragen gesteld aan de opdrachtgever. Wanneer er naar aanleiding van de klant iets is aangepast word dit aangegeven. Ook wanneer er aannames gemaakt worden, wordt dit aangegeven

# Use case diagram

Een use case diagram is een simpele representatie van hoe een gebruiker op het systeem kan inwerken. Een use case diagram kan de verschillende types van gebruikers die het systeem zullen gebruiken identificeren. Een use case diagram wordt gebruikt om een “higher level view” van het systeem te krijgen. Dit betekent dat het een versimpelde en grafische versie van wat het systeem moet doen.

Het doel van deze opdracht was om een simpel beeld te krijgen van wat er uiteindelijk allemaal moet gebeuren.



## 2.1 Brief description

1. Registreren. Deze use case betekent dat een persoon zich moet kunnen registreren. Deze use case heeft use case 2 nodig anders weet hij niet welke gegevens de klant wilt gebruiken om zich te kunnen registreren.

2. Gegevens invullen. Deze use case houdt in dat wanneer een persoon zich wilt registeren hij zijn gegevens in moet vullen zodat hij deze later weer kan gebruiken om in te loggen.

3. Pasje naar klant sturen. Deze use case houdt in dat wanneer een persoon zich geregistreerd heeft en zijn gegevens ingevuld heeft een pasje thuisgestuurd krijgt door het systeem. Dit pasje moet de klant later gebruiken wanneer hij een auto wilt gebruiken

4. Gegevens van klanten aanpassen. Deze use case houdt in dat een RentIt medewerker de gegevens van een klant aan kan passen. Dit is er voor het geval de klant iets verkeerd gedaan heeft en dit opgelost moet worden.

5. Klanten inactief maken. Deze use case is er voor RentIt medewerkers om te zorgen dat klanten die zich niet aan de regels gehouden hebben inactief gemaakt kunnen worden. Dit is er zodat wanneer een klant zich niet aan de regels gehouden heeft hij geen gebruik meer zal kunnen maken van het systeem.

6. Autoinformatie beheren. Deze use case is er voor RentIt medewerkers om gegevens van auto’s aan te passen. Dit zal bijvoorbeeld gebeuren wanneer er bepaalde data van een auto niet meer klopt.

7. Berekenen van kosten. Deze use case wordt uitgevoerd door het systeem. Het houdt in dat de kosten die gemaakt worden door een klant bij elkaar opgeteld worden en verekend worden bij de klant. Dit gebeurt alleen nadat een klant een auto gereserveerd heeft.

8. Kijken waar auto’s geparkeerd staan per stad. Deze use case is er voor de klanten. Het houdt in dat een klant in staat moet zijn om te kunnen zien waar elke auto geparkeerd staat.

9. kijken of de auto beschikbaar is. Deze use case is er voor de klanten. Het houdt in dat wanneer een klant kijkt naar waar elke auto staat ook moet kunnen zien welke van deze auto’s beschikbaar is.

10. Auto reserveren. Deze use case is er voor de klanten. Het houdt in dat klanten in staat moeten zijn om een auto te reserveren. Dit is er zodat de klant later ook daadwerkelijk gebruik kan maken van een auto.

11. Begin- en eindtijd invullen. Deze use case is er voor de klanten. Dit is er zodat wanneer een klant een reservering wilt plaatsen hij in kan vullen van wanneer tot wanneer de klant de auto wilt gebruiken.

12. Abonnement kiezen. Deze use case is er voor de klanten. Dit wordt gebruikt wanneer een klant een reservering wilt plaatsen. Hier kan de klant voor een abonnement kiezen wat voor hem het meest gepast lijkt.

13. Auto ophalen. Deze use case is er voor klanten die een auto gereserveerd hebben en deze auto willen gebruiken binnen de tijd dat ze de auto gereserveerd hebben.

14. inchecken. Deze use case is er voor klanten die een auto op willen halen. Wanneer een klant een auto ophaalt moet hij eerst ingecheckt zijn, omdat een klant een auto niet mag gebruiken zonder ingecheckt te zijn.

# Niet functionele eisen

Registreren:

Beheer van gegevens:

Van auto’s moet kenteken, type en standplaats vastgelegd zijn. Usability

Gebruik van de auto:

Op de website moet staan waar auto’s geparkeerd staan. Usability

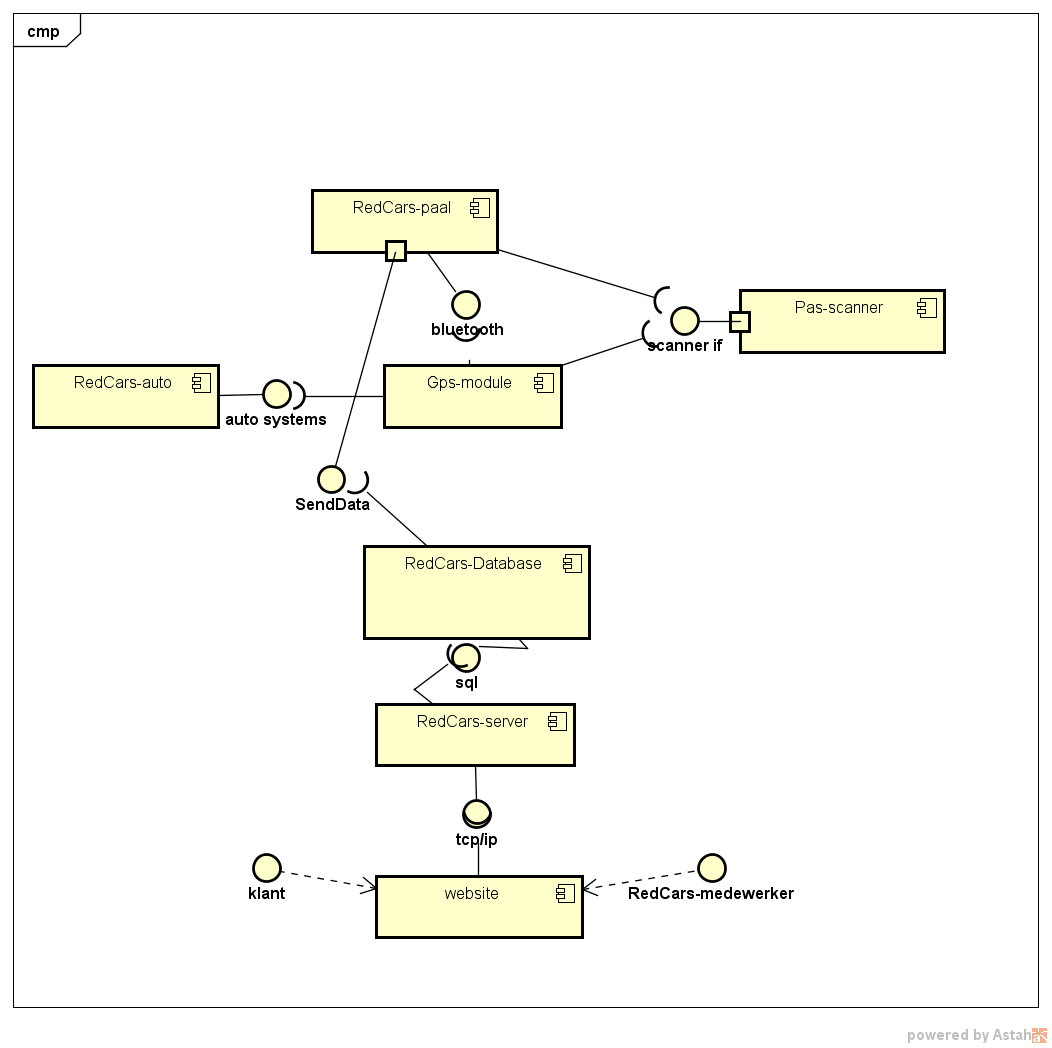
Het bekijken waar auto’s staan moet binnen 1 minuut kunnen, 90% van de tijd Performance

Op de website moet staan welke auto’s beschikbaar zijn. Usability

Het bekijken welke auto’s beschikbaar zijn moet binnen 1 minuut kunnen, 90% van de tijd

Performance

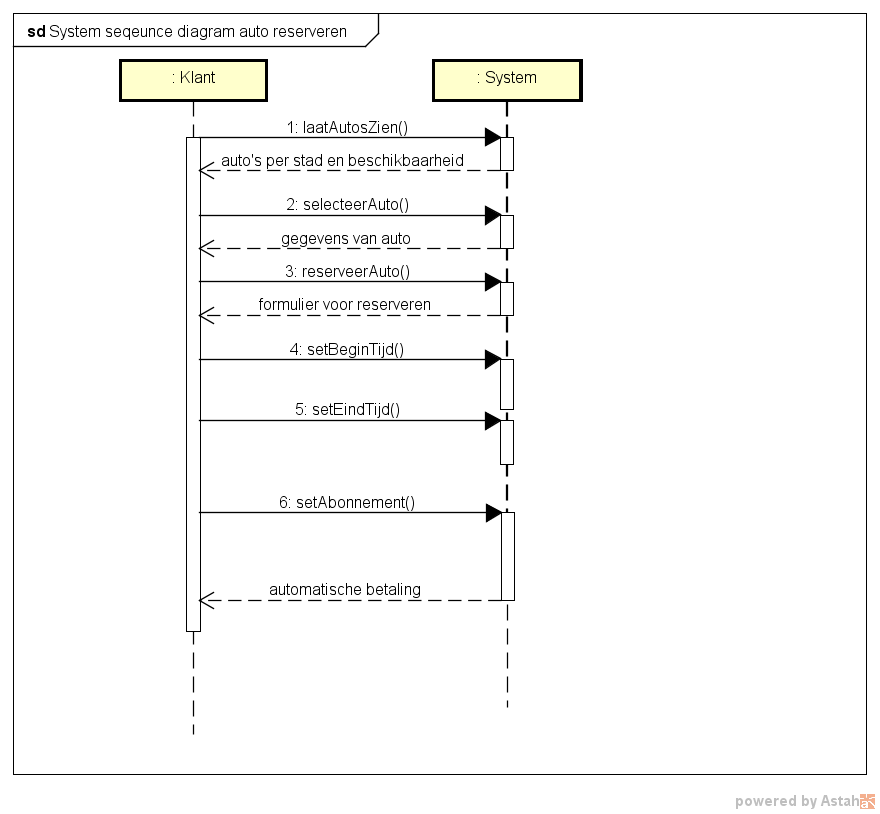
# Componenten diagram

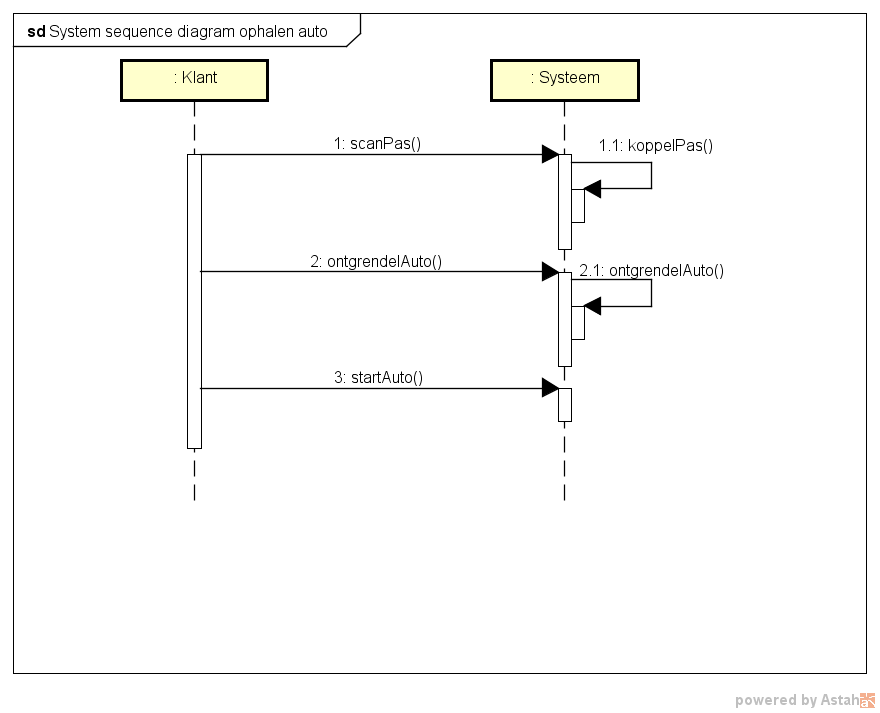


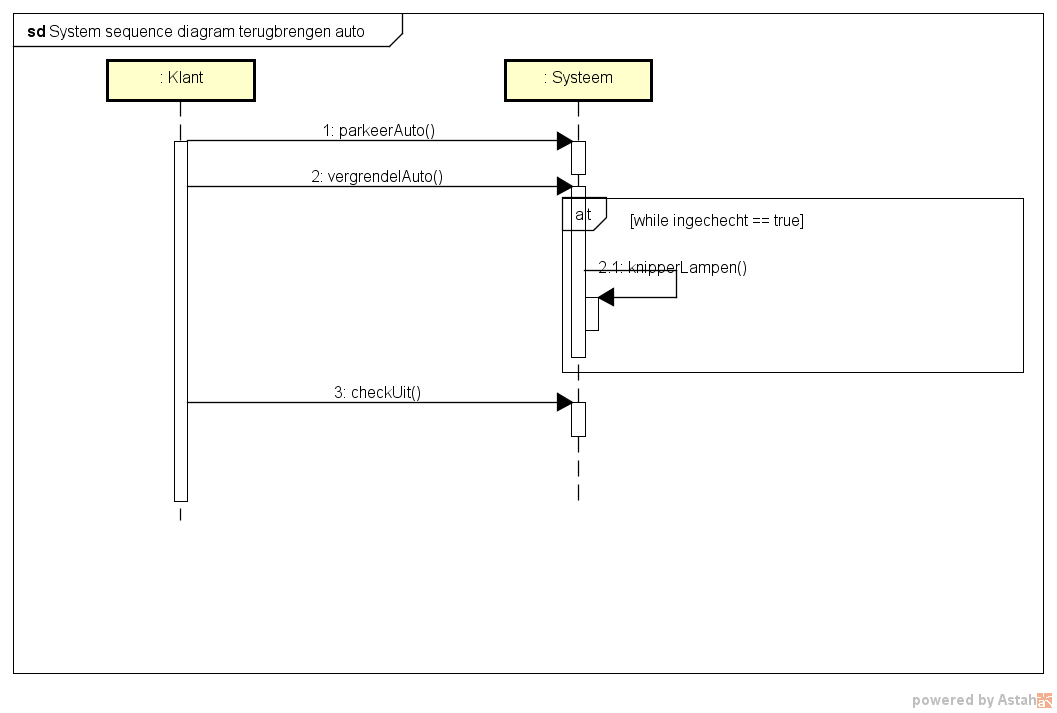
# Deployment Diagram

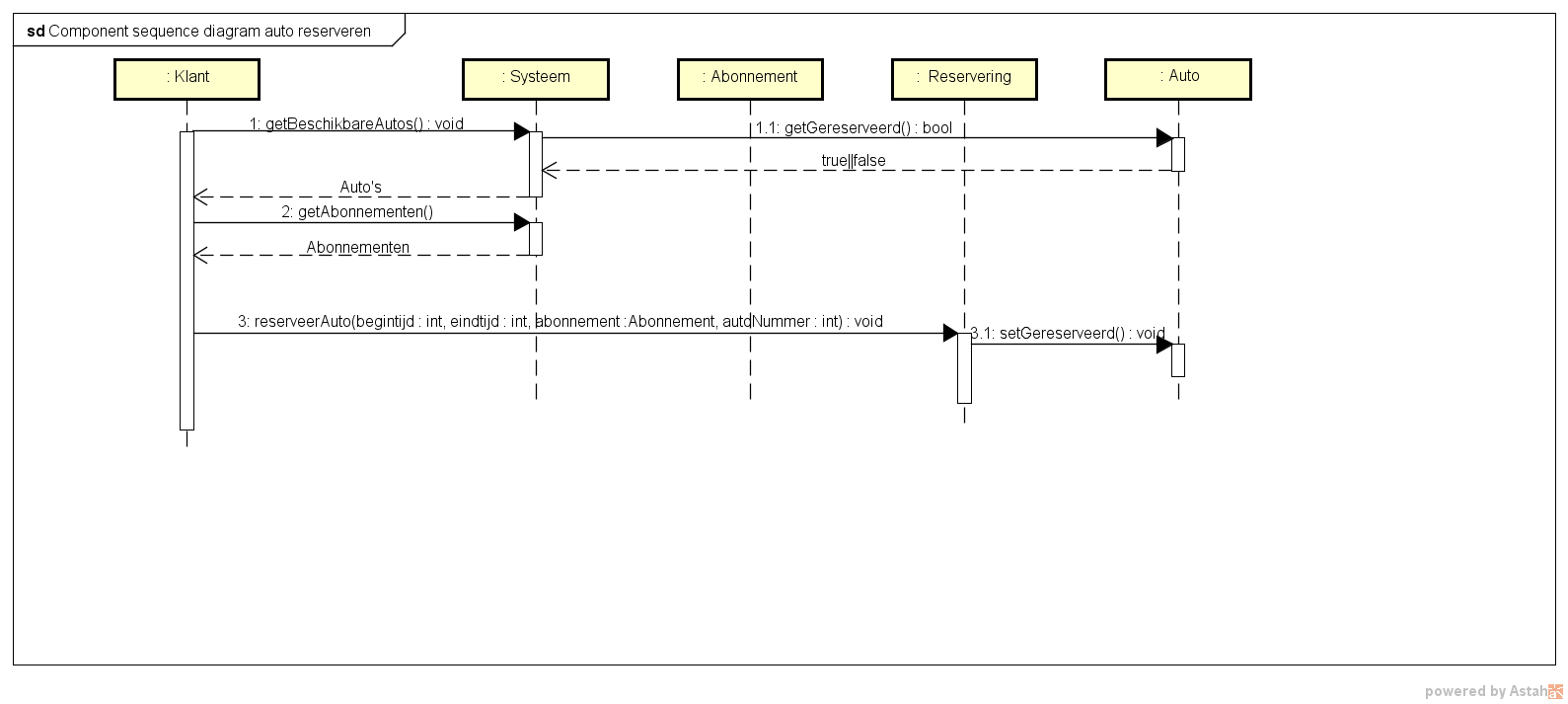
# Domeinmodel

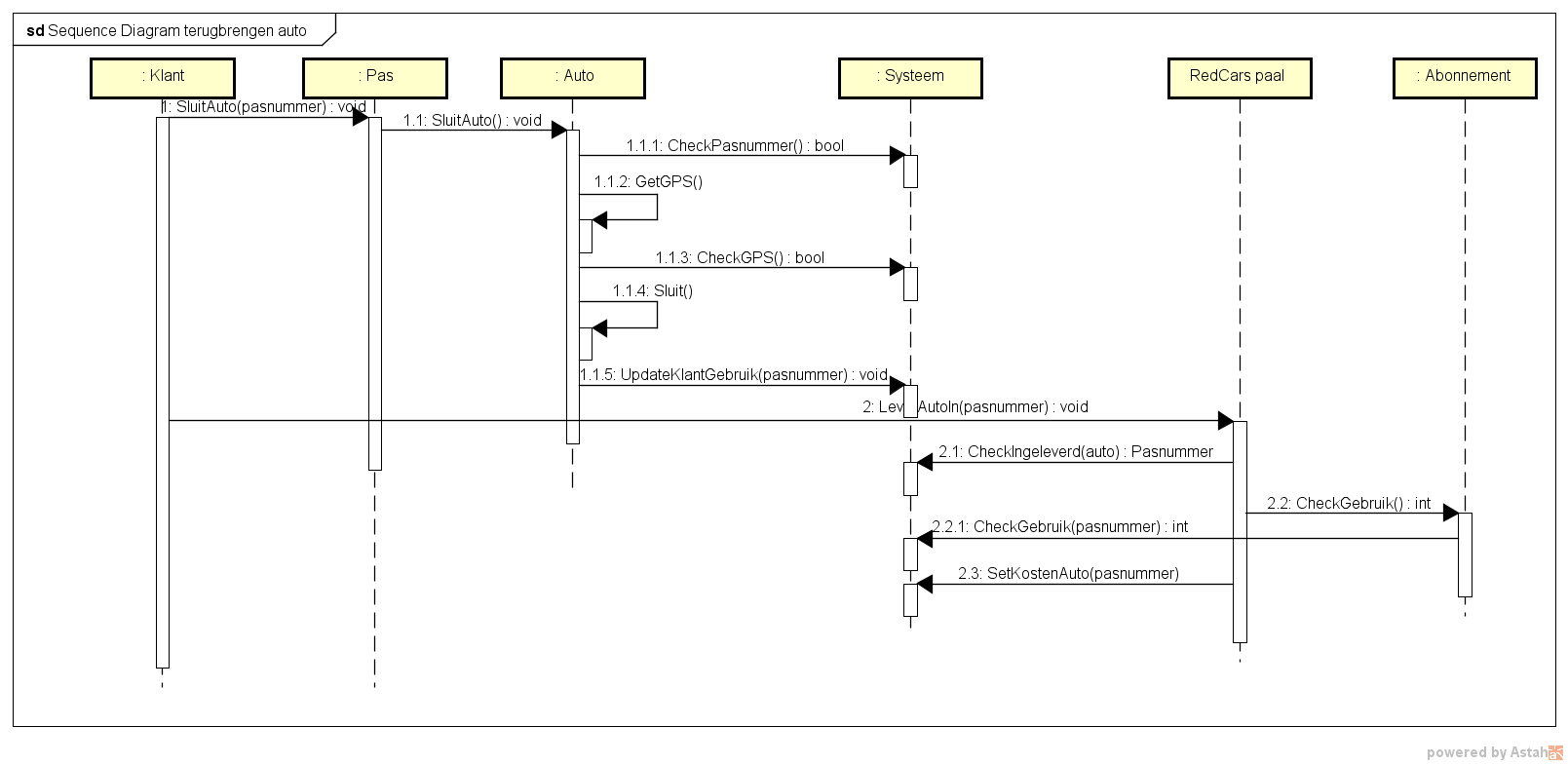
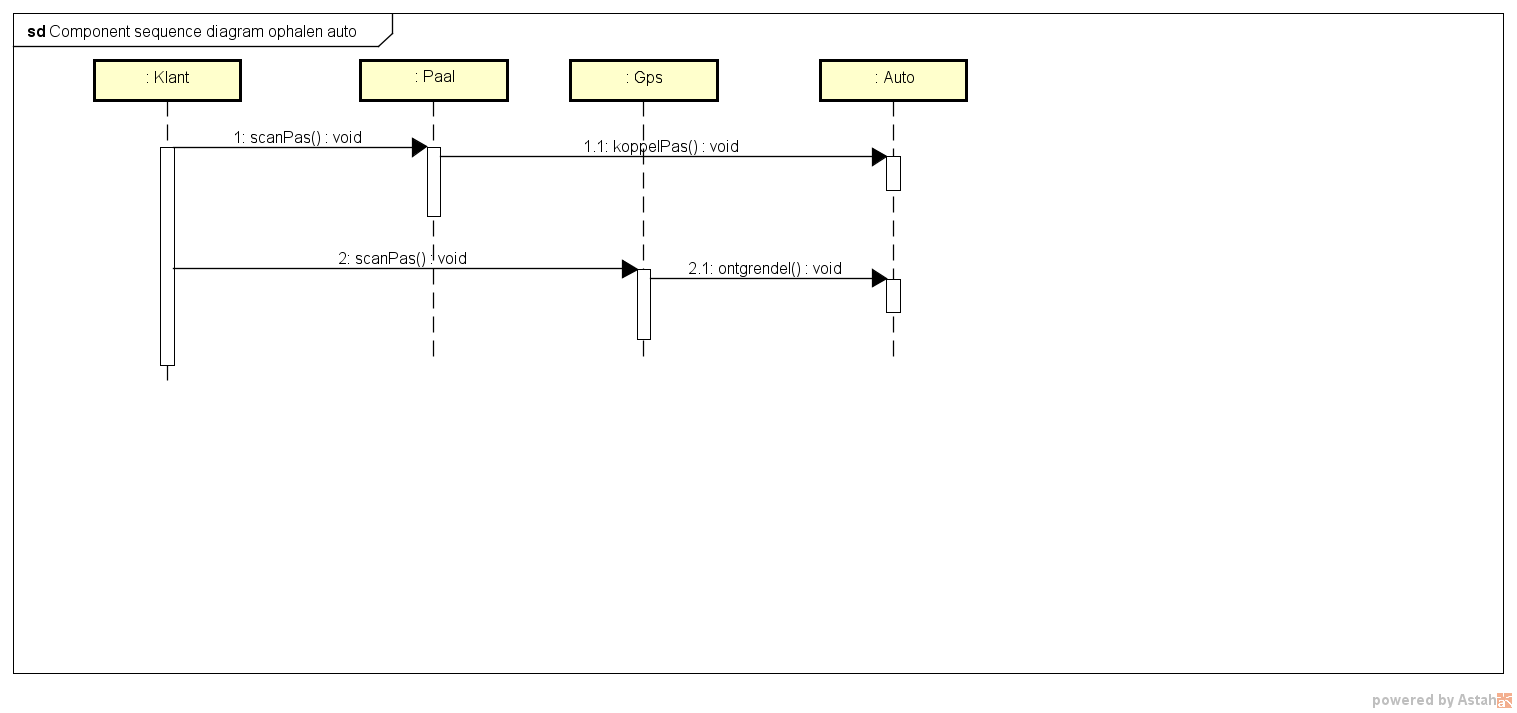
# Sequence diagrammen



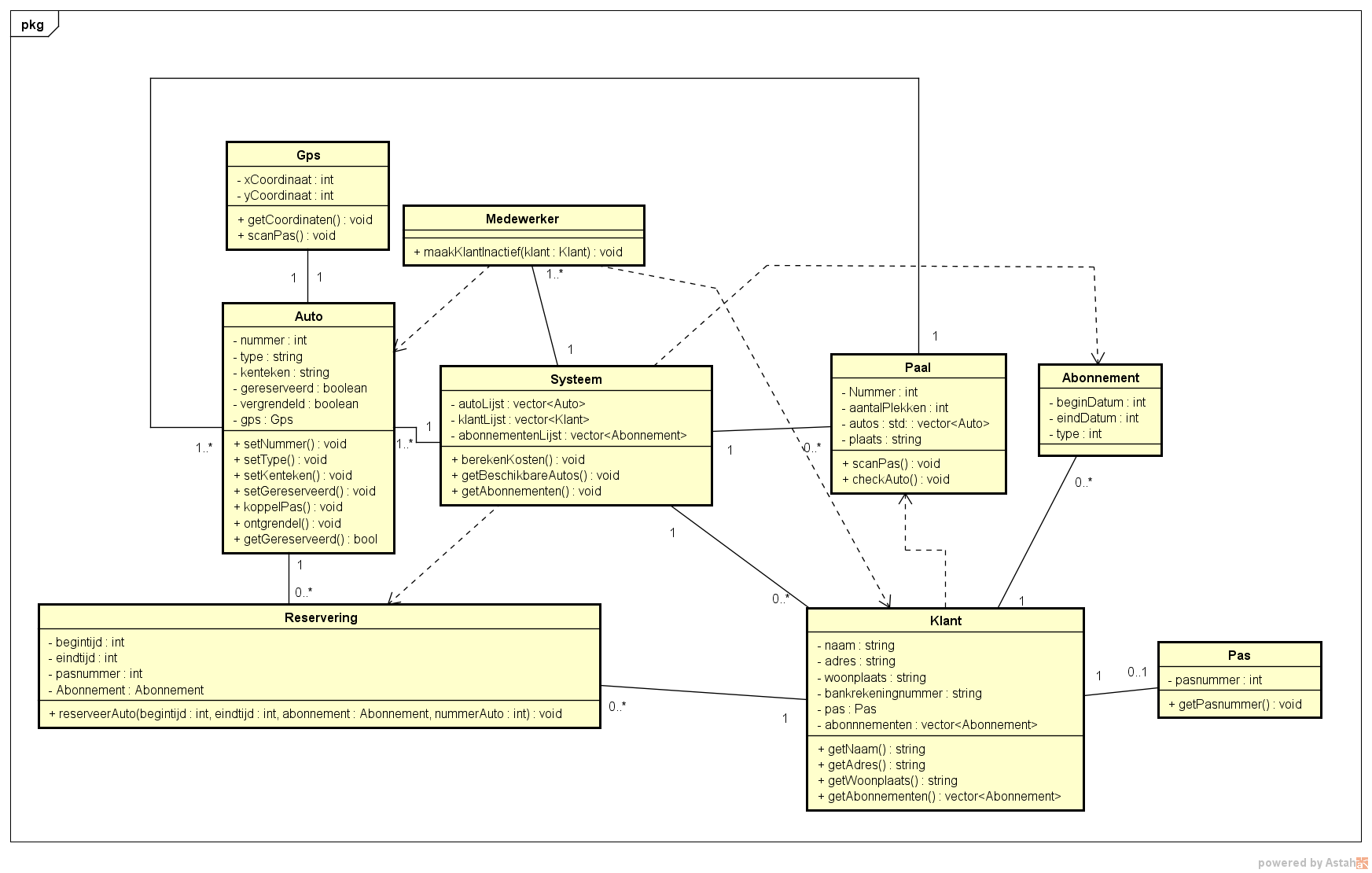








# Design class diagram



# Applicatie