SOFTWARE DESIGN DOCUMENT

Opdracht van RentIt

Merlijn Warps[621737] & Sandra ter Maat [620458]

Datum: 29-10-2020

Inhoud

1 Introduction

1.1 Overall Description

Zie hoofdstuk 1.1 van het SRS.

1.2 Purpose of this document

Het doel van dit document is het verschaffen van inzicht in de ontwerpkeuzes die gemaakt zijn. Het geeft een overzicht van de software architectuur en geeft uitleg over de interacties tussen de softwarecomponenten.

1.3 Definitions, acronyms, and abbreviations

Er wordt geen gebruik gemaakt van acroniemen of afkortingen.

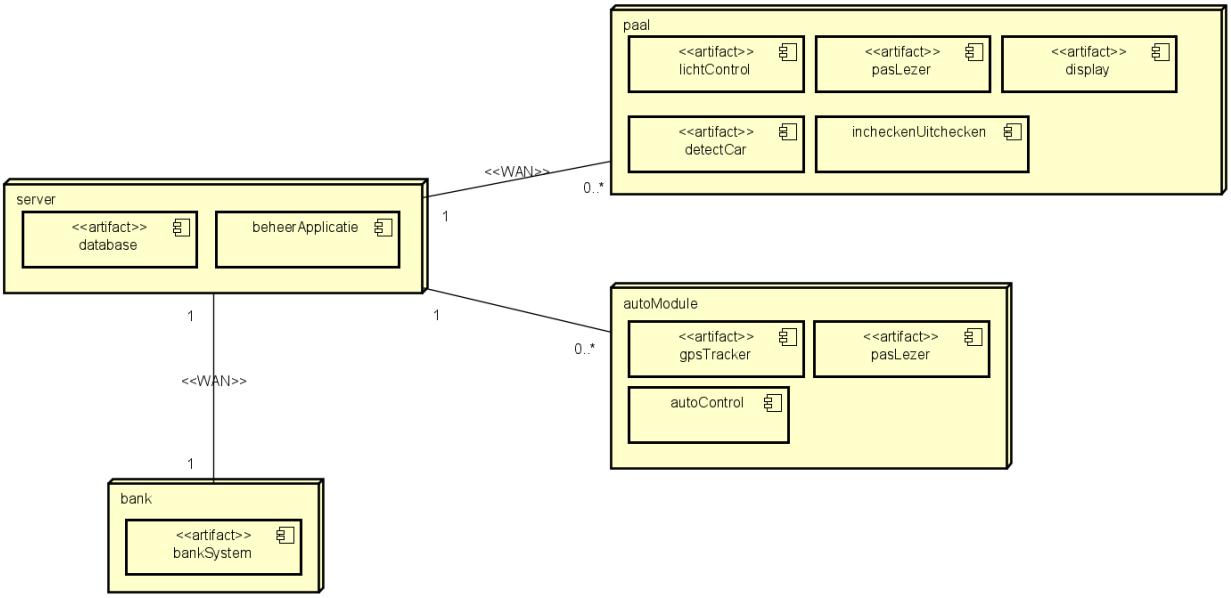
2 Architectural Overview

Het systeem bestaat uit een centrale computer waar de centrale server op draait. In elk van de leenauto's bevindt zich een module, deze module heeft een connectie met de centrale server. Ook staat de server verbonden met de palen op de parkeerlocaties van RedCars.

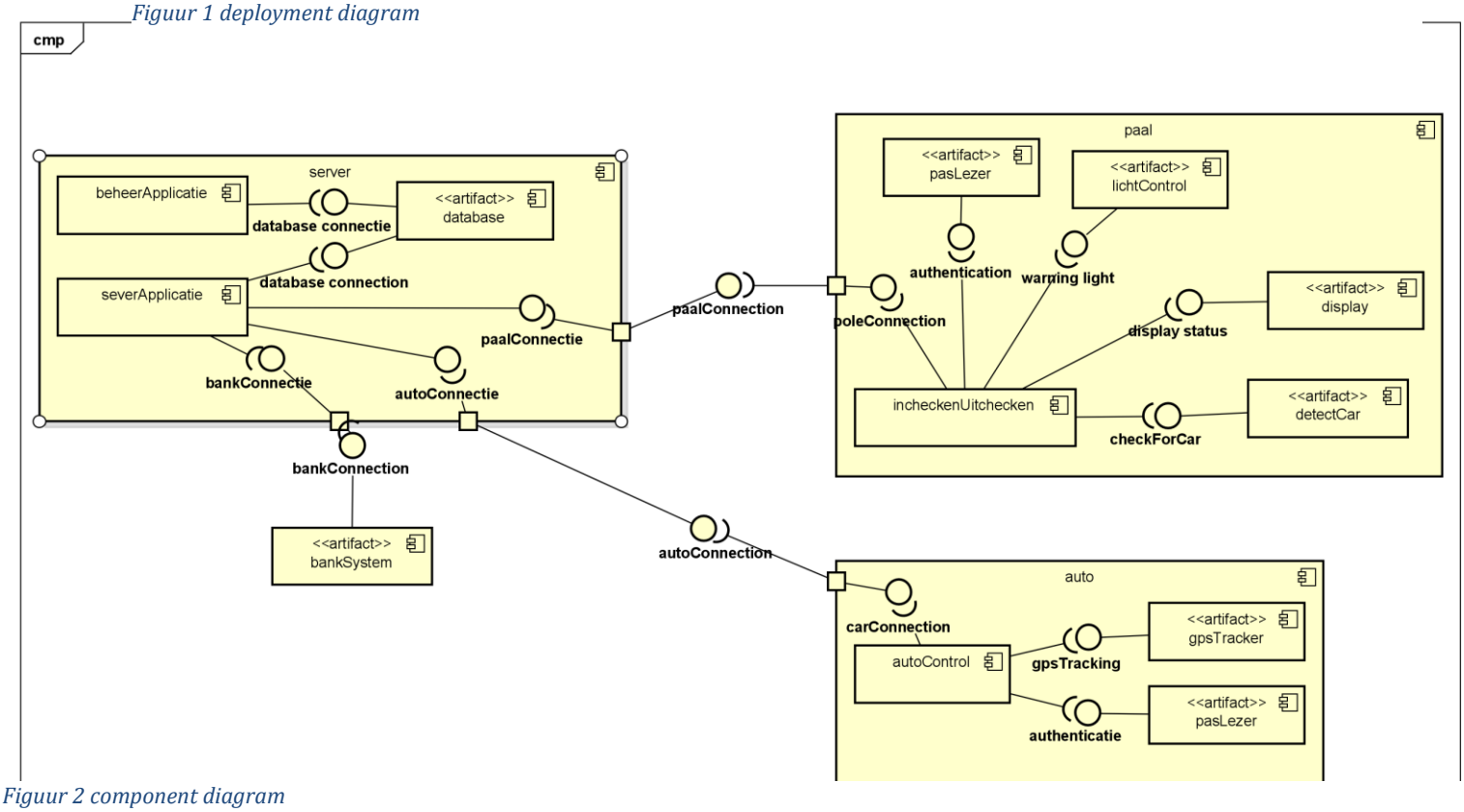
3 Detailed Design Description

3.1 Deployment en Component Diagram

Hieronder het deployment en component diagram. De server, de paal en de autoModule zijn onderdeel van het RedCars systeem. De bank is een extern systeem dat gebruikt wordt voor het afhandelen van de betalingen.



Figuur 1 deployment diagram



Figuur 2 component diagram

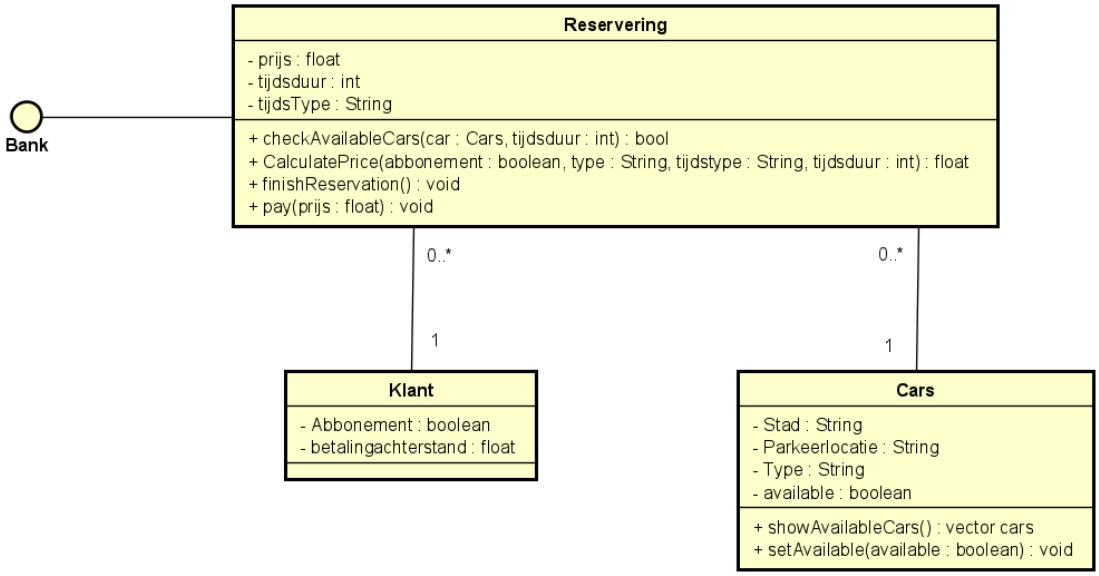
3.1.1 Design Decisions related to deployment en compopent

Alle functionaliteiten die via externe sensoren en libraries uitgevoerd worden zijn als artifact in het deployment diagram gezet. Daarna is bij de paal en de auto een component gemaakt dat de artifacts aanstuurt om de gewenste functionaliteit van het systeem te realiseren. Er is gekozen voor een centrale serverapplicatie die een directe verbinding heeft met zowel de database als de overige componenten (bank, paal en auto). Deze keuze is gemaakt zodat alle componenten toegang hebben tot de voor hen benodigde informatie. Ook is het hebben van een centrale dataopslag bevorderlijk voor de veiligheid van de data en overzicht voor de medewerkers.

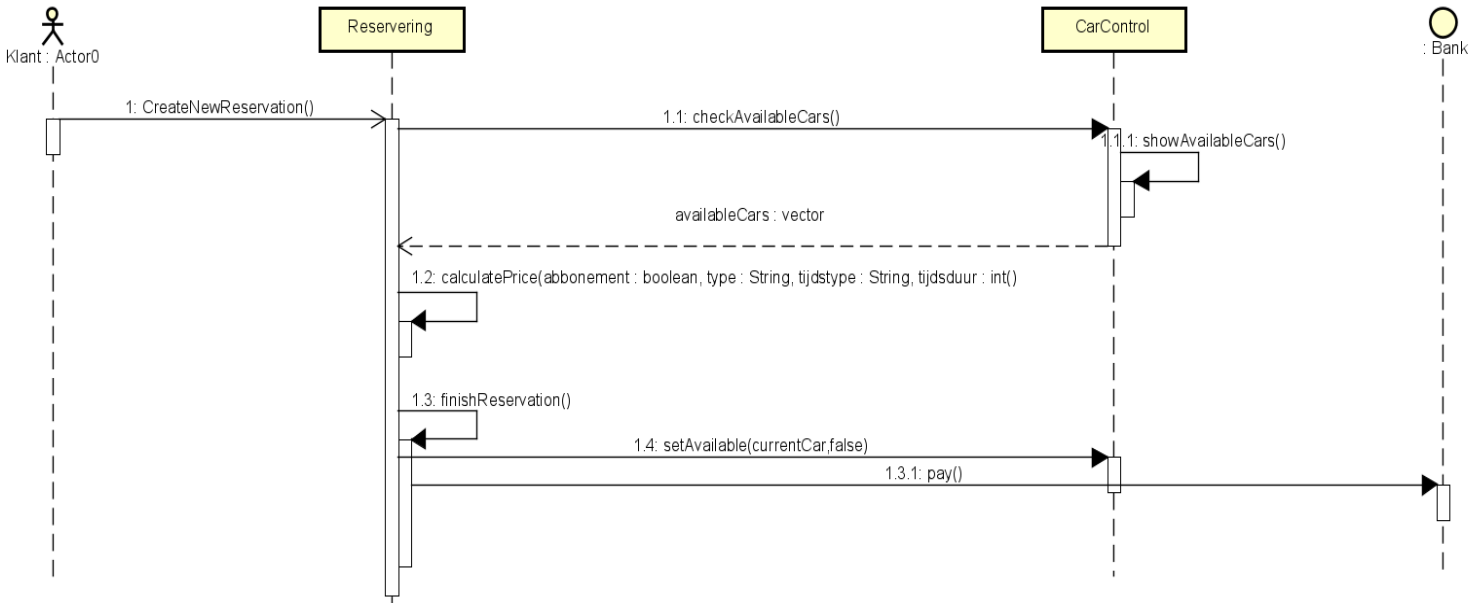
3.2 Reserveren en Betalen

Hier wordt het subsystem weergeven dat nodig voor het uitvoeren van de usecase Reserveren en betalen, deze staat uitgelegd in het SRS hoofdstuk 3.2.

3.2.1 Design Class Diagram



3.2.2 Sequence Diagrams



De klant maakt een nieuwe reservering aan. De reservering vraagt aan carcontrol welke auto's beschikbaar zijn in de stad en parkeerlocatie die de klant heeft opgegeven. CarControl geeft een vector terug met alle auto's die beschikbaar zijn. De klant kiest één van deze auto's, aangezien dit met een klik gedaan is en geen verdere acties oproept is dit niet opgenomen in het diagram. Na het kiezen wordt de prijs berekend aan de hand van het type abbonemten van de klant, het type auto, het tijdstype en de tijdsduur. Als de klant akkoord gaat wordt de reservering afgehandeld. De gekozen auto op onbeschikbaar gezet voor de gereserveerde tijdperiode en er wordt aan de bank gevraagd een automatische afschrijving naar de klant te sturen.

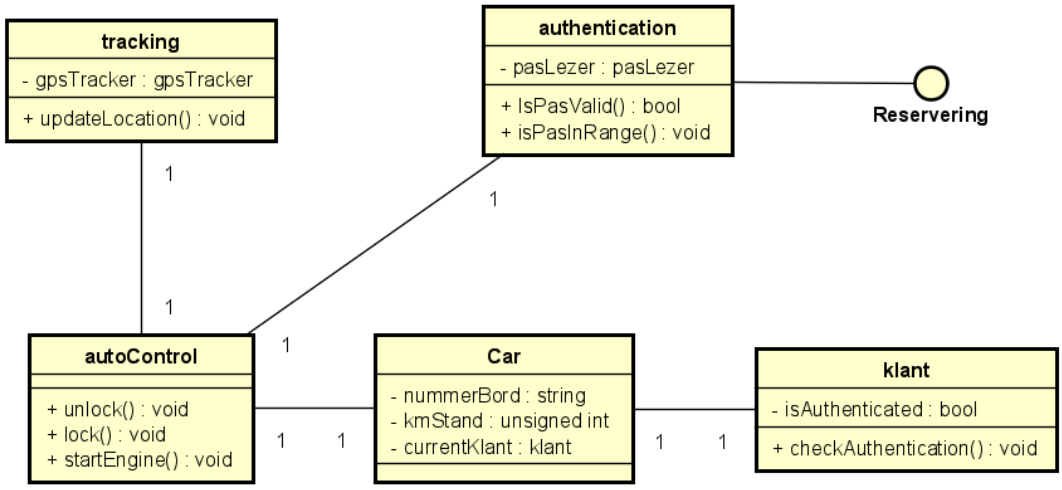
3.2.3 Design decisions made for the sub-system

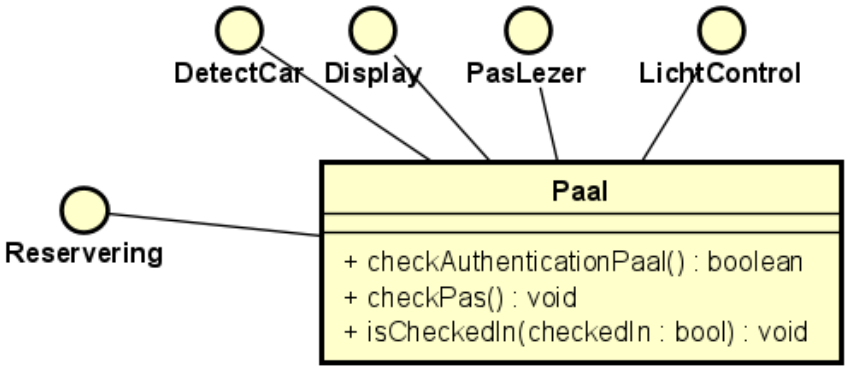
Het is gewenst dat de klant zo min mogelijk acties hoeft uit te voeren en dat elk onderdeel alleen de vereiste informatie van andere onderdelen bevat. Hierdoor is er voor gekozen om Reservering te laten communiceren met CarControl ovder de data van de auto's. Wanneer de availability van de auto verandert laat Reservering dit aan CarControl weten. Wat de bank doet staat los van het systeem.

3.3 Auto Ophalen

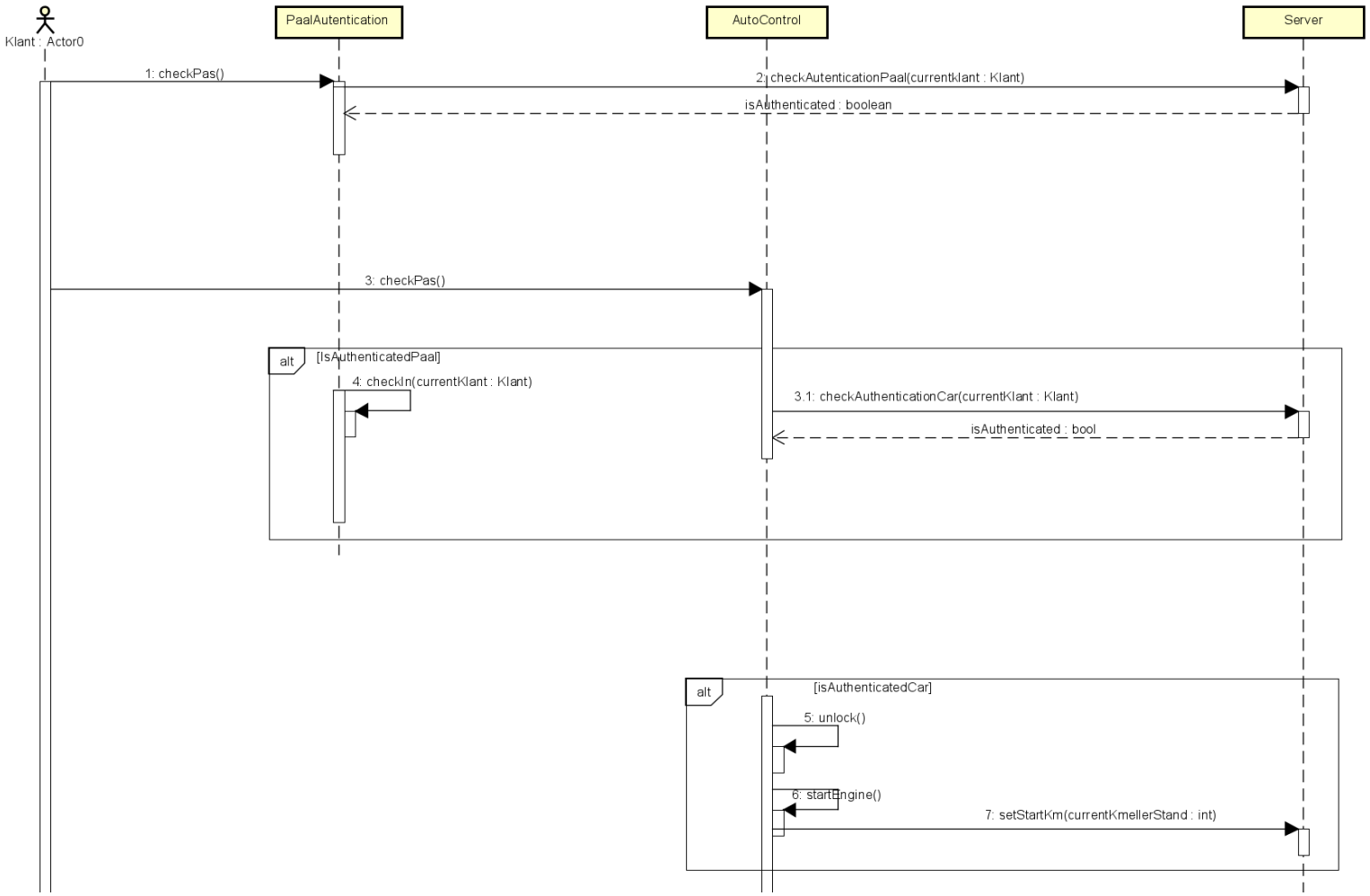
Hier wordt het subsystem weergeven dat nodig voor het uitvoeren van de usecase Auto Ophalen, deze staat uitgelegd in het SRS hoofdstuk 3.6.

3.3.1 Design Class Diagram





3.3.2 Sequence Diagrams



Het enige wat de klant hoeft te doen het scannen van zijn RedCars pas. Als eerste doet hij dit bij de paal, de paal vraagt aan de server of de desbetreffende klant nog niet is ingecheckt en of hij hier een auto mag inchecken. Wanneer dit het geval is, dan is de klant geautoriseerd om een auto mee te nemen. De klant scant nu zijn pas bij de auto. AutoControl vraagt aan de server of de klant deze specifieke auto mag nemen, wanneer dit het geval is, is de klant ingecheckt en gaat de auto open en kan de klant hem meenemen.

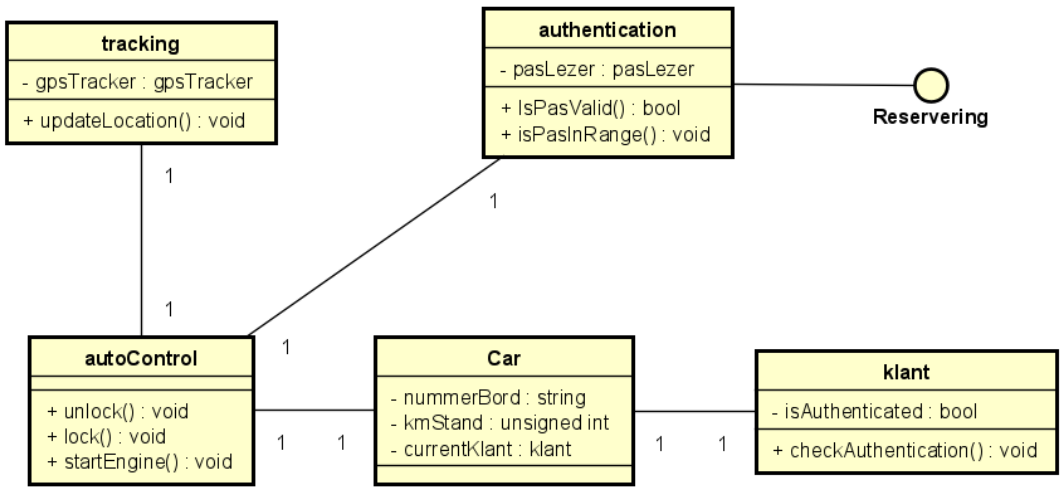
3.3.3 Design decisions made for the sub-system

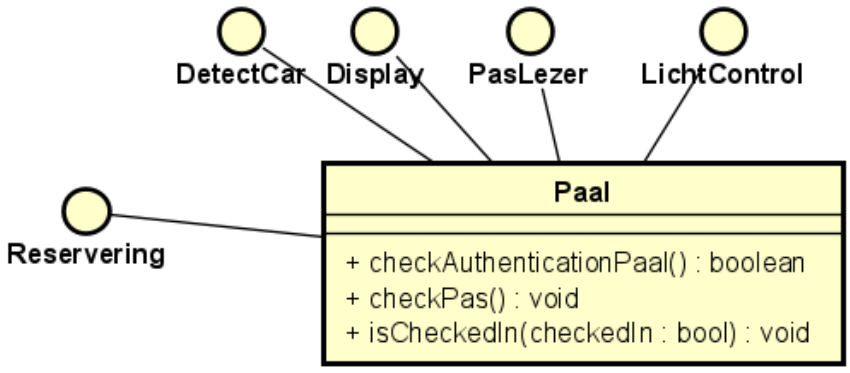
De interfaces detectCar, Display, Paslezer en LichtControl worden afgehandeld door externe libraries. Er is gekozen om deze interfaces gescheiden te houden volgens het interface segregation principe.

3.4 Auto Wegbrengen

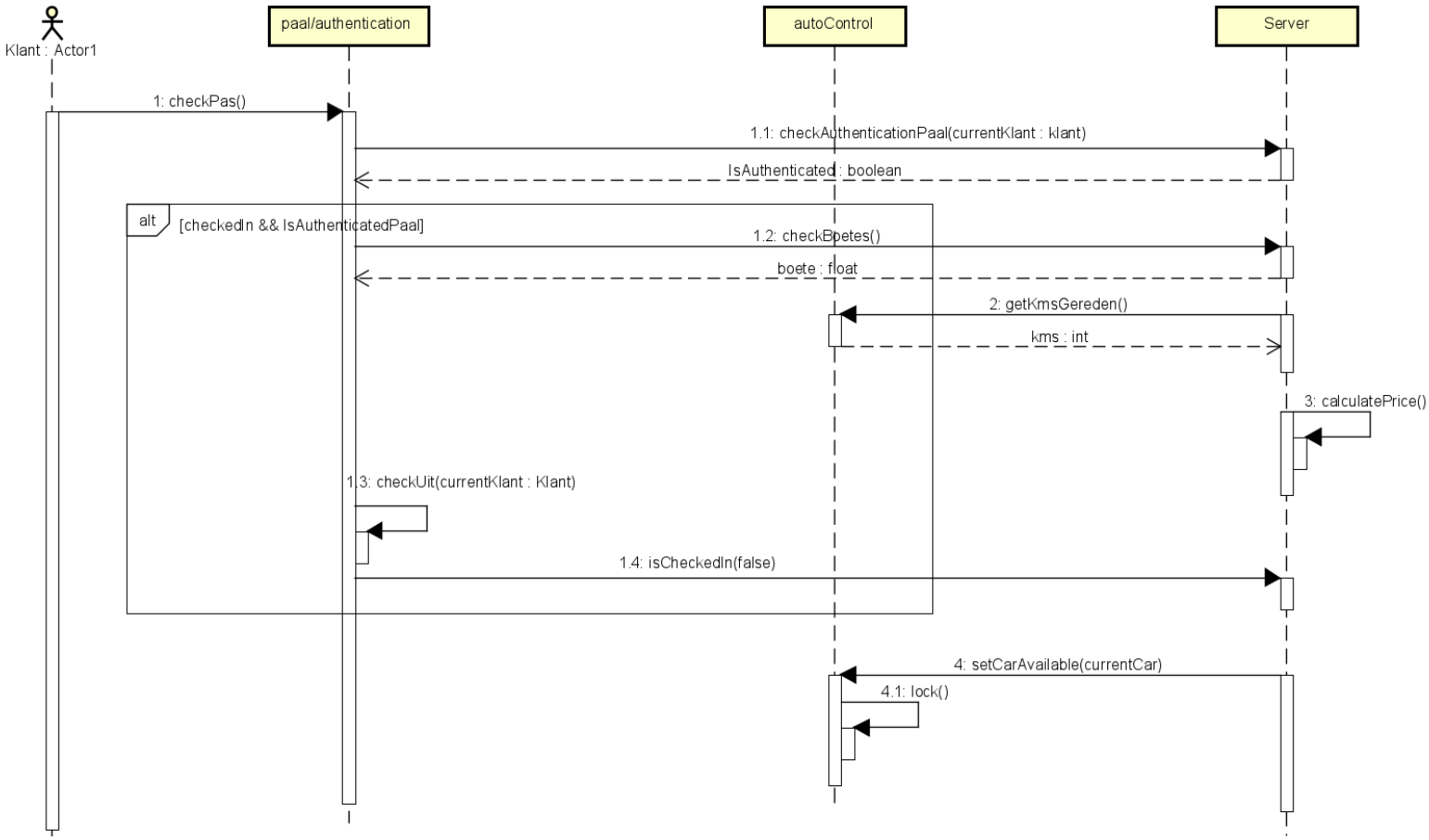
Hier wordt het subsystem weergeven dat nodig voor het uitvoeren van de usecase Auto Ophalen, deze staat uitgelegd in het SRS hoofdstuk 3.4.

3.4.1 Design Class Diagram





3.4.2 Sequence Diagrams



Het enige wat de klant hoeft te doen het scannen van zijn RedCars pas bij de paal. De paal vraagt aan de server of de klant in ingecheckt en een auto mag terugbrengen op deze locatie. Wanneer dit het geval is wordt er gecheckt op eventuele boetes. De Server vraagt het aantal gereden km's op aan autoControl en berekend de prijs van de rit. Bij het uitchecken van de klant wordt een automatisch incassering bij de bank aangevraagd. De auto wordt op beschikbaar gezet en zet zichzelf op slot.

3.4.3 Design decisions made for the sub-system

De interfaces detectCar, Display, Paslezer en LichtControl worden afgehandeld door externe libraries. Er is gekozen om deze interfaces gescheiden te houden volgens het interface segregation principe.