

Software Design Essentials

Opgave 3

1 Class diagram: Garage

We ontwerpen een applicatie voor een garagehouder. In de garage worden auto's verkocht van verschillende types: sportwagens, cabrio's, gewone sedans, terreinwagens, bestelwagens en kleine stadswagens. Van elk type wagen worden er een aantal modellen verkocht, waarbij elk model een aantal varianten kan hebben: aantal deuren (3-, 4- of 5-deurs uitvoering), motorisatie, afwerkingsniveau, ... In de showroom werken een aantal verkopers die natuurlijk proberen om zoveel mogelijk wagens te verkopen. Van elke verkoop wordt bijgehouden welke verkoper betrokken was, aan welke klant de wagen werd verkocht en over welke wagen het gaat. Daarnaast worden de auto's die in de showroom staan bijgehouden in het programma.

Naast de verkoop van nieuwe wagens, voorziet de garage ook in het onderhoud van de wagens. Hierbij moet rekening gehouden worden met een schedule. Er kunnen namelijk maar een aantal wagens per dag hersteld worden. In de schedule wordt voor elke week en dag van het jaar bijgehouden welke wagens onderhouden moeten worden. Er zijn 2 types onderhoud: klein en groot onderhoud. Voor het grote onderhoud wordt 5 uur gereserveerd, het kleine onderhoud duurt ongeveer 2u. Afhankelijk van de onderdelen die bij het onderhoud worden vervangen, wordt een bepaalde prijs aangerekend. Wanneer een onderhoud wordt gescheduled, verbinden we die meteen aan een techniek die het onderhoud zal uitvoeren. Zo kunnen we bij klachten ook eenvoudig nakijken wie het onderhoud heeft uitgevoerd. De techniek verbonden aan een onderhoud moet achteraf wel wijzigbaar zijn, zodat een verandering in de schedule mogelijk is.

Probeer zelf inventief te zijn en niet alleen strikt de opgave te vertalen in een class diagram. Denk zelf na over je model om een zo realistisch mogelijke weergave te bekomen. Voeg zelf de nodige attributen en operaties toe, maar voorzie zeker in volgende mogelijkheden:

- Registreren van een nieuwe verkoop
- Toevoegen van een nieuwe wagen in de showroom
- Toevoegen van een nieuw model, nieuw type of nieuwe variant van een wagen
- Schemen van een klein/groot onderhoud

- Checken of er nog plaats vrij is in de shedule
- Wijzigen van een schedule (andere techniekeer, ander uur, ...)
- ...

2 System Sequence Diagram: Reserveringssysteem voor een bibliotheek

Je bent gevraagd om een deel van een systeem te ontwerpen voor het reserveren van boeken in een bibliotheek. Het systeem moet gebruikers in staat stellen om online een boek te zoeken en te reserveren. Als een gebruiker een boek vindt en wil reserveren, moet het systeem controleren of het boek beschikbaar is. Als het boek beschikbaar is, wordt de reservering vastgelegd na bevestiging, en wordt het boek gemarkeerd als "gereserveerd". Zo niet, dan moet het systeem de gebruiker informeren dat het boek momenteel niet beschikbaar is en alternatieven aanbieden.

Gebruik een System Sequence Diagram (SSD) om de interactie tussen de gebruiker en het systeem te modelleren voor het succesvol reserveren van een boek. Zorg ervoor dat je de volgende elementen in je diagram opneemt:

1. Actoren (in dit geval ten minste de gebruiker).
2. Het systeem waarbinnen de acties plaatsvinden.
3. De stappen die de gebruiker moet ondernemen om een boek te zoeken en te reserveren.
4. De system responses op de acties van de gebruiker, inclusief zowel het succesvol reserveren van een boek als het informeren van de gebruiker als het boek niet beschikbaar is.

Denk aan de details die je in elke stap zou verwachten, zoals het invoeren van zoekcriteria, het ontvangen van zoekresultaten, het selecteren van een boek, en het bevestigen van de reservering.

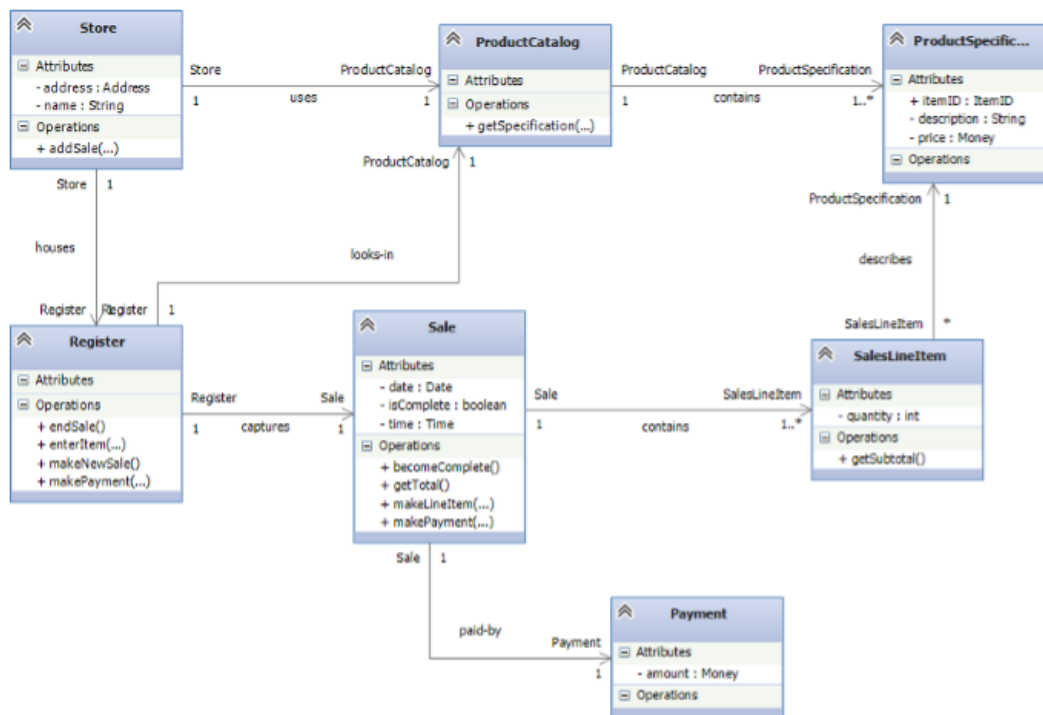
3 Interaction sequence diagram: Hotelreservering

Maak het sequentiediagram voor een hotelreservering. Het object dat de reeks berichten initieert, is een Reserveringsvenster. Het Reserveringsvenster stuurt een bericht `makeReservation()` naar een Hotelketen. De Hotelketen stuurt vervolgens een bericht `makeReservation()` naar een Hotel. Als het Hotel voor elke dag van de reservering een beschikbare kamer heeft, maakt het een Reservering en een Bevestiging.

4 Interaction sequence diagrams: Point-Of-Sale

Een voorbeeldapplicatie die we in het hoorcollege deels besproken hebben is de zogenaamde "POS" of Point-Of-Sale applicatie, de typische verkoopapplicatie die gebruikt

wordt in een grootwarenhuis voor de verkoop van producten aan klanten. Hieronder wordt een gedeeltelijk klasse-diagramma getoond van een mogelijke implementatie van zo'n POS-systeem.



Het klasse-diagramma biedt enkel een statisch zicht op de mogelijke kandidaat-classes met zijn attributen, operaties en relaties tussen de klassen onderling. In vele gevallen kan een interaction diagram meer inzicht bieden in de manier waarop interacties tussen objecten plaatsvinden. Maak interaction sequence diagram en denk daarbij ook na over de interacties tussen objecten onderling voor het implementeren van de use case “Process sale” waarin eerst een Sale object wordt aangemaakt om vervolgens product per product toe te voegen aan de sale. Op het einde van de verkoop wordt het totaalbedrag opgevraagd zodat dit bedrag kan getoond worden op het scherm.