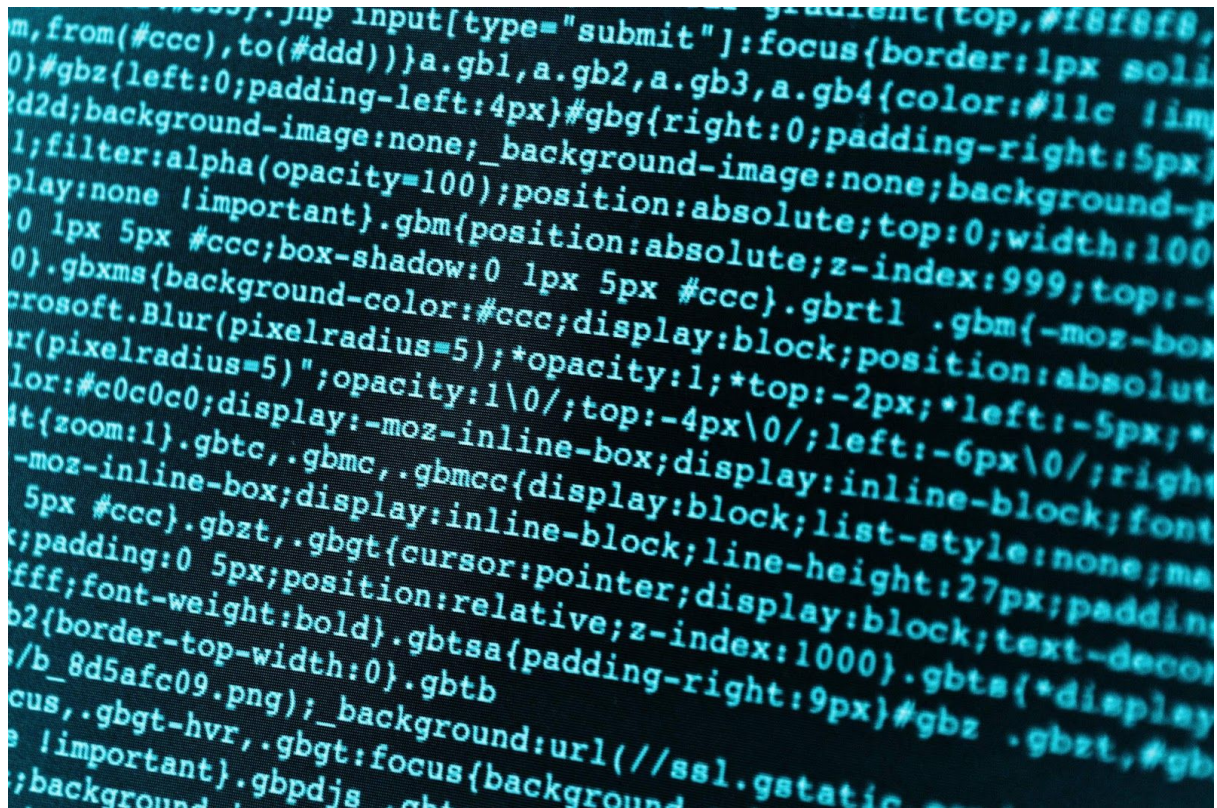


Elaboration Rapport



Skrevet af: Ian, Jonas, Tobias, Ahmed og Nikolaj

Projektbeskrivelse

Iterations oversigt / Tidsplan

Inception phase #1

Elaboration - Iterations plan #1

Elaboration - Iterations plan #2

Elaboration - Iterations plan #3

Unified Process

Beskrivelse af UP

Forklaring af Unified process

Iterativ

Incremental

Risiko drevet

Arkitektur centreret

Inception

Project Description

Technical Tools

Group Contract

Project Management

Business analysis

SWOT analyse

Styrker

Svagheder

Trussel

Opportunities

Porter's 5 Forces

Threat of new entrants

Power of suppliers

Power of buyers

Threat of substitute products (Lav)

Competitive rivalry (Høj)

The Business Model Canvas

IT-strategi

Beskrivelse

Mr. Truckers nye systems skal kunne følgende:

Inception phase

Purpose

Use-case diagram

Elicit stakeholders requests

[Vores primære aktører er](#)

[Vores sekundære aktører er](#)

[Offstage aktører er: Investorer, banken, bilvask etc.](#)

[Use-Cases](#)

[Use-Case tekst](#)

[Prioriteret Liste af FURPS](#)

[Risikoanalyse](#)

[Introduction – Shared vision](#)

[Positioning](#)

[Problem Statement](#)

[Product Position Statement](#)

[Stakeholder Summary](#)

[User Environment](#)

[Product Overview](#)

[Needs and Features](#)

[Other Product Requirements](#)

[Feasibility analyse](#)

[Legal feasibility](#)

[Schedule feasibility](#)

[Cultural feasibility](#)

[Resource feasibility](#)

[Technical feasibility](#)

[Economical feasibility](#)

[Elaboration](#)

[Beskrivelse af Elaboration phase](#)

[Use cases](#)

[Beskrivelse af use-case tekst](#)

[Opsummering](#)

[Opret salg](#)

[Opdatere og slette information](#)

[Domain model](#)

[Beskrivelse](#)

[System Sequence Diagram - SSD](#)

[Beskrivelse](#)

[SSD: Car](#)

[SSD: Sale](#)

[Arktitektur](#)

[Beskrivelse](#)

[MVVM – Model-View-View Model](#)

[Framework](#)

[Codebehind](#)

[Operation Contracts](#)[Beskrivelse](#)[Operation Contract - Car](#)[Operation Contract - Sale](#)[GUI & Prototype](#)[Beskrivelse](#)[Program brugt til fremstilling](#)[Prototype type](#)[Prototype typer](#)[High fidelity](#)[Low fidelity](#)[Design Sequence Diagram](#)[Beskrivelse](#)[Design Sequence Diagram for Sale](#)[Design Sequence Diagram for Car](#)[Design Class Diagram](#)[Beskrivelse](#)[Start Up Design Sequence Diagram](#)[Beskrivelse](#)[Start Up Design Sequence Diagram - Car](#)[Start Up Design Sequence Diagram - Sale](#)[Evaluerings af det færdige produkt](#)[Formelt](#)[Generelt](#)[Evaluerings af projektet](#)[Diskussion af Inception fase længde](#)[Evaluerings af gruppearbejdet](#)[Bilag](#)[Bilag 1: Use case create customer](#)[Bilag 2: Use case create salesperson](#)[Bilag 3:](#)[Domain model for bil](#)[Domain model for customer](#)[Domain model for sale](#)[Bilag 4](#)[SSD for Customer](#)[SSD for Salesperson](#)[Bilag 5](#)[Operation Contract - Salesperson](#)[Operation](#)

[Operation Contract - Customer](#)

[Bilag 6](#)

[Prototyper af Load, Customer og Salesperson](#)

[Bilag 7](#)

[Design sequence diagram](#)

[Design Sequence Diagram for Salesperson](#)

[Design Sequence Diagram for Customer](#)

[Bilag 8](#)

[Start Up App Design Sequence Diagram for Customer](#)

[Start Up App Design Sequence Diagram for Salesperson](#)

[Bilag 9](#)

[Dictionary](#)

Projektbeskrivelse

Projektet er 38 antal sider langt

Bilag udgør 10 antal sider

Vi har i perioden fra d. 12/02-2017 til d. 3/6-2017 beskæftiget os med at udarbejde et IT-system til en bilforhandler, der skulle opfylde kundens krav, disse er fundet ved hjælp af interviews, samtaler samt nedskrevne krav på papir.

Programmer vi vil beskæftige os med under projektforsløbet er følgende: Microsoft Visual Studio, Microsoft Visio, GitHub, Microsoft Office Word og Axure.

Det endelige IT-system kan oprette kunder, sælgere og biler, samt lave salg som binder en kunde, sælger og bil sammen.

Andre features er at programmet kan beregne kommission ud fra bilens pris til en givende sælger. Man kan update og delete informationer, hvor nogen information vil blive valideret. Dette bliver også omtalt i afsnittet "evaluering af det færdige produkt" (side 39)

Projektet udarbejdes efter Unified process (*se afsnittet vedr. s. 8*)

Iterations oversigt / Tidsplan

Inception phase #1

Opstart: 12/2-2017

Slut: 03/03-2017

Grunden til at vi laver inception phase, er så vi kan få et overblik over forretnings nuværende status, samt et klart billede af hvordan markedet er i givne tidsperiode (her bliver bl.a SWOT og Porter's 5 forces benyttet.)

Elaboration - Iterations plan #1

De use cases der er vigtigst for vores system at virker optimalt og for at få en prototype produceret, er nedenstående 3 use cases. Derfor har vi valgt at den første iteration i elaborations fasen, skal fokusere på disse 3 elementer:

Iteration opstart: 04/03-2017

Iteration slut: 30/04-2017

- Opret Bil -> Iteration 1
- Opret Kunde -> Iteration 1
- Opret Sælger -> Iteration 1

Elaboration - Iterations plan #2

Iteration opstart: 1/05-2017

Iteration slut: 15/05-2017

Når elaboration phase #2 er færdig, er vores system i sådan en tilstand, at den kan alt det basale for at forretningen kan fungere:

- Der kan oprettes sælgere, kunder og biler
- Man kan sælge en bil til den pågældende kunde
- Der kan udregnes provision af salget til den enkelte sælger

Den use case der er vigtigst for vores system i iteration 2 er nedenstående use case. Da det med at kunne oprette et salg, vil give stor værdi for forretningen.

- Opret salg

Elaboration - Iterations plan #3

Iteration opstart 15/5

Iteration slut 25/5

I iteration 3, kigger vi på ting i systemet, som kan digitalisere overtage manuelle opgaver, såsom at sælge en bil. Derudover vil vi implementere muligheden for at opdatere og slette oplysninger.

I denne iteration har vi lavet følgende:

- Opdatering af informationer hos kunder, sælger og bil
- Delete, slette informationer hos kunder, sælger og bil
- Find samlet kommission via. sælgers ID

Unified Process

Beskrivelse af UP

Vi har haft 3 udviklingsmetoder i tankerne, som er følgende:

1) Waterfall

- Man kan ikke lave ændringer på de foregående trin – derfor skal man have alle kravende specificerede i starten og hvert trin gennemarbejdes ned til mindste detalje.
- Bruges oftest til mindre projekter, så kravene ikke ændres og projektet går i vasken.

2) Agile

- Agile software metoder henviser til en række forskellige metoder, der er baseret på iterative processer. Her er vi mere skudt op på en "timeboks" (ca. 2 uger) hvor man skal vise kunden et stykke produkt. Vi kan stadigvæk ændre hvis kunden ikke er tilfreds.
- Bruges når alle krav fås direkte fra kunden og der er et tæt udvekslingsforhold af informationer som evalueres i timebokse. Der bruges i environment hvor der jævnligt er ændringer af kravene.

3) Unified process

- Metoden er iterativ (man laver jævne iterationer, så man hele tiden kan få evalueret det nuværende arbejde og se om krævende er korrekt forstået og om der er noget der ikke er som det burde og dermed *skal laves om*) Dette mindsker fejl og/eller mangler i sidste ende.
- Bruges som en blanding af agile og Water, vi kender nogle af kravene fra starten og udarbejde derefter via iteration evt flere krav - Hvilket gør at kunden og programmørerne skaber det ønskede produkt.

Forklaring af Unified process

Iterativ

- Hele forløbet bliver delt op i flere iterationer, hvor kunden efter hvert iterativ kommer og kigger på den tilsvarende "Releases"

Incremental

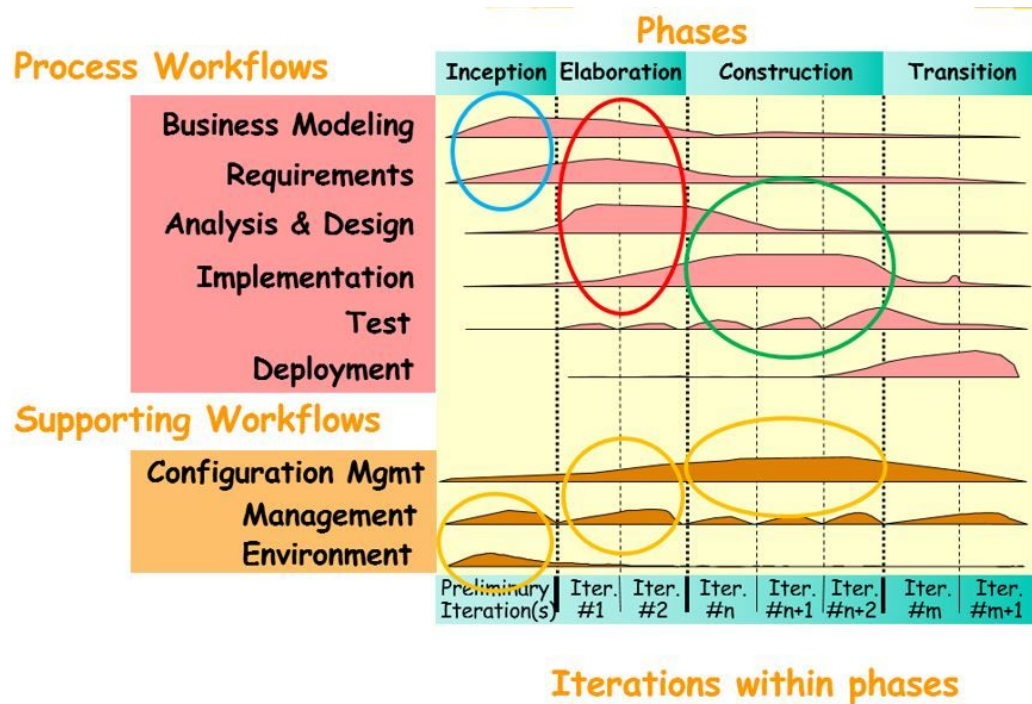
- Incremental udvikling betyder at systemet bliver udviklet stykke by stykke.

Risiko drevet

- Når man er nået til en Milestone, tjekker man risiciene i de forskellige dele og ser om det er muligt at mindre denne.

Arkitektur centreret

- Arkitektur omfatter strukturen af systemet og spille en central rolle i udviklingen. I de tidligere iterationer er der fokus på at bygge, teste og stabilisere en kerne arkitektur. Dette er vigtig da mangel på solid arkitektur i sig selv er en høj risiko.
- - Implementere det vigtigste **først**!



UP består af 4 faser: Inception, Elaboration, Construction, Transition. Hver af faserne bliver beskrevet i henhold til deres respektive afsnit.

Inception

Project Description

Rapporten tager udgangspunkt i 2 scenarier:

- 1) Forretningsanalysen er lavet ud fra en virkelig bilforhandleres (placering i Roskilde) samt nogle realistiske detaljer vi selv har "opdigtet" (på baggrund af erfaringer og realistiske antagelser)
- 2) Alle punkterne efter "IT-strategien" tager udgangspunkt i en udleveret tekst "Car dealership"

Technical Tools

- Visual Studio 2015
- MVVM
- Axure RP 8

Group Contract

- Er man syg, skal der sende en SMS rundt til hele gruppen.
- Hvis man påtager sig opgaver og får en deadline, skal den overholdes.
- Man skal være fleksible mht. møde tidligere og gå senere, samt hjemmearbejde.

Project Management

- Vi benytter os af programmet Trello, så vi hele tiden kan se hvad der bliver arbejdet med og hvem der arbejder med hvad.
 - o Det giver os overblik og sparer os tid i sidste ende.

Business analysis

SWOT analyse

Ejner Hessel Roskilde bilforhandler: lokalitet, lige ud til motorvejen. Ligger lige op ad et utal af andre bilforhandlere, der sælger de samme salgs biler i tilsvarende prisklasse.

Åbningstiderne er i tidsrummet 9-17 mandag til fredag, lørdag 10-14 kun service samt søndag efter aftale.

Ejner Hessel Hos Ejner Hessel sælger vi økonomiske biler og billige brugte biler.

Bil prisklassen ligger imellem 10.000-150.000kr. i mange forskellige modeller.

Forretningen ligger på hovedvejen, hvor der kører mange potentielle kunder forbi. Vi er populært blandt kunder, men føler vores IT-system kan forbedres.

Vi håber derfor I kan hjælpe os med at lave et nyt informationssystem. Vi har tre slags kategorier blandt vores biler: vi har familiebiler, sportsbiler og brugte biler som er gjort i stand. For dem som køber for et beløb større end 50.000kr tilbyder vi dem en månedlig vask for et helt år.

Vi har desuden også ugentlige tilbud på vores biler. Kunderne med interesse for at bytte deres egen bil for en prisreducering på et køb er også en realitet hos Ejner Hessel. Man kan dog kun bytte én bil af et salg.

Medarbejderne arbejder på minimumslønnen, men det meste af deres indkomst kommer fra deres kommission af bilsalg. Sælgerne belønnes yderligere ved en større kommission skal de sælge mere: 5% kommission for salg op til 100.000kr per måned, 7% for salg mellem 100.00-125.000kr og 10% for salg fra 125.000 og opefter. Kommissionen er regnet ugentlig og er en procentdel af deres total salg inkluderet yderligere muligheder og forlænget garanti osv. Vi holder øje med medarbejdernes indsats og kan belønnes med titlen som "Årets sælger", hertil følger en bonus.

Styrker	Svagheder
<ul style="list-style-type: none">- Du kan sælge din gamle bil og penge for den når du køber en ny.- Køber over 50.000 får en månedlig bilvask.- Weekly specials with discounts- Saleman of the year = bonus (holder moralen oppe og motivationen i top)- Flexibilitet i forhold til aftaler uden for åbningstider.	<ul style="list-style-type: none">- Medarbejder sælger på kommission (Lorte uge à dårlig løn à opsigelse à mindre arbejdskraft)- ineffektivt it-system.

Trussel <ul style="list-style-type: none"> - Der ligger mange bilforhandlere i Roskilde. (så der er hård konkurrence) 	Opportunities <ul style="list-style-type: none"> - Populær hovedvejen med høj færden. - De 17 årige må nu tage kørekort.
---	---

Porter's 5 Forces

Threat of new entrants

- Høj kapital (husleje, biler, reservedele, løsninger m.m.)

Eftersom at man skal have en relativt stor startkapital, medfører det at ikke hvem som helst kan åbne en ny bilforretning, da man skal godkendes af en evt. bank til lån, finde investorer m.m.

- Derfor er Threat of new entrants **"lav"**.

Power of suppliers

- Der er mange leverandører til det samme produkt (biler, reservedele m.m.)

Der er mange som sælger reservedele og biler, så derfor kan kunden få det ønskede produkt mange steder.

- Derfor er power of suppliers **"Lav"**.

Power of buyers

- Stort udvalg af bilmærker i samme prisklasse.
- Mange firkanter.

Som der også står i "power of suppliers" så har kunden (bilforhandleren) et bredt udvalg af fabrikater samt bilmærker.

- Derfor er power of buyers **"høj"**

Threat of substitute products (Lav)

- Offentlig transport. (busser, tog inkl. Taxa)

I stedet for folk købte nye eller brugte biler, kunne de i stedet benytte sig af offentlig transport, taxaer, cykler m.m.

- Ifølge DST er antallet af biler i Danmark stadig meget højt, derfor er threat of substitute products **"lav"**¹

¹ <http://www.dst.dk/da/Statistik/nyt/NytHtml?cid=22985>

Competitive rivalry (Høj)

- Mange forhandlere der har de samme biler til samme pris.
- Privat salg af brugte biler uden om forhandler.

Det er her vi kigger på hvordan markedet for vores bilforhandler er. Ud fra hvor vores bilforhandler ligger, kan vi se at der er 9 bilforhandler indenfor en radius af 1 km, hvor 5 af dem sælger biler i sammen prisklasse/modeller.

- Derfor er Competitive rivalry "høj"
-

The Business Model Canvas

Business Model Canvas sammenfatter og illustrere hovedindholdet i virksomheden. De mange forskellige punkter (fx key partners, channels m.m.) har en eller anden påvirkning på hinanden.

- Det giver derfor et godt overblik over hvilke byggesten der er i virksomheden og hvor man evt. kan lave forbedringer.

The Business Model Canvas		Designed for: Bilforhandler i Roskilde	Designed by: Ian, Tobias, Ahmed, Nikolaj og Jonas	Dato: 16/022017 Iteration Text
Key Partners <ul style="list-style-type: none"> • Bilfabrikker • Andre forhandlere 	Key Activities <ul style="list-style-type: none"> • Salg af bilér • Reparationer • Service 	Value Propositions <ul style="list-style-type: none"> • Biler • Lave priser • Bredt sortiment • Mærkevarer 	Customer Relationships <ul style="list-style-type: none"> • Personlig service • Åbent Hus • Udstillinger • Nyhedsbreve • Mængderabat • Specielle ugentlig tilbud • Flexibilitet mth. åbningstider. 	Customer Segments <ul style="list-style-type: none"> • Familie • Gennemsnitlige lønmodtager
	Key Resources <ul style="list-style-type: none"> • Mekaniker • Sælger • Butik (showroom) • Værksted • Bogholder (til regnskab) 		Channels <ul style="list-style-type: none"> • Køb ved Personlig fremmøde • Mulighed for levering til adresse. 	
Cost Structure <ul style="list-style-type: none"> • "Husleje" • Reservedele • Personale (Løn) • Køb af biler 		Revenue Streams <ul style="list-style-type: none"> • Salg af biler (nye samt brugte) • Udlejning • Service • Reparation 		

IT-strategi

Vi tager nu udgangspunkt i teksten: The Car Dealership.

IT-strategien er udarbejdet på baggrund af tekstens kriterie.

Derudover bruger vi i arkitekturmodellen MVVM og programmeringssproget er C#.

Beskrivelse

I IT-strategien laves der nogle forslag til forbedringer i virksomheden på baggrund af forretningsanalysen.

Mr. Truckers nye systems skal kunne følgende:

Record customer information

- Når man skal leje en bil, skal man oprettes i et program, der kan gemme oplysninger og informationer om kunden.

Der antages at mr. Trucker har brugt udprintet udlejningskontrakter før i tiden, og kan nu effektivisere hans udlejning proces. Nu er hans chance for at smide en kontakt væk, væsentlig mindsket.

Create Invoices

- Oprettes via informationer fra "customer information" og sender en faktura ud på mail, gemmes også i vores database.

Record information about new and used cars

- Ved at gemme informationer omkring bilerne, kan han lave statistikker over hvilke biler der fx sælger bedst, efterspurgt, gode køb og dårlige og selvfølgelig, hvilke biler der er på lager og som kan sælges.

Calculate salesperson commission

- Salgsperson indberetter i systemet hvilket bil han har solgt ("bilnummer") ud fra salget, beregner programmet selv kommissionen og vedlægger det til den pågældende sælgers løn.

Inception phase

Purpose

Grunden til at vi laver en Inception phase, er så vi kan få et overblik over forretnings nuværende status samt et klart billede af hvordan markedet er pt (SWOT, Porter 5 forces m.m.)

Derudover benytter vi os af en række værktøjer/modeller: Use-cases, risikoanalyse og Feasibility.

Alt ovenstående er med henblik på at finde ud af om det kan betale sig, tidsmæssigt, profitable m.m. (go or not go)

Use-case diagram

I use-Case diagrammet, kigger vi på de Use-cases der er indenfor vores scope og hvordan de interagerer med vores primære aktører, sekundære aktører og offstages.

Elicit stakeholders requests

The stakeholders of the car dealer System are: chefen, sælgerne, kunder, fabrikanter, samarbejdspartner, bilvask.

Vores primære aktører er

Sælgeren og chefen, eftersom det er dem der skal bruge systemet i sidste ende.

- Sælgerens formål med systemet er: bestille biler hjem, registrer kunde og indberet salg.
- Chefens formål med systemet er: Kunde redigering og invoices.

Vores sekundære aktører er

Leverandørerne til use-casen "Bestille biler hjem" da vi ikke kan bestille biler hjem, hvis vi ikke har en leverandør og kunderne da vi ikke kan oprette en ny kunde og lave invoices, hvis vi ikke har kundes information.

- Den sekundære aktør gør så den primære aktør kan opnå sit mål (det er dog ikke alle use-cases der er en sekundær aktør – Se eks Sælger å indberet salg.)

Offstage aktører er: Investorer, banken, bilvask etc.

Use-Cases

- Opret bil
- Registrere kunder
- Indberet salg
- Opret sægler
- Kunder redigering
- Invoices

Se også bilag 1 (create Customer) & 2 (create salesperson)

Use-Case tekst

Use Case	Create new car
ID	UC_01
Summary	Når vores sælger skal oprette en ny bil i systemet.
Primær Aktør	Sælger
Sekundær Aktør	
Generalization	
Include	
Extend	
Pre-betingelser	System, bil informationer m.m.
Trigger	Der er købt en ny bil der skal sælges, så derfor skal den oprettes i vores system.
Normal flow	<ol style="list-style-type: none"> 1) Bruger: Tryk "opret ny bil" Systemet: - Opretter skabelon til en ny oprettelse af bil. 2) Bruger: Udfylder bilens informationer (License Plate, category, price, color, brand, model, kilometers) Systemet: - Gemmer informationerne som katalog 3) Bruger: Trykker "Afslut" Systemet: Returnere os til forsiden

Alternate flow/exceptions	<p>2.a Manglede udfyldelse af information på bilen (kategorier, pris, årgang, model, mærke og antal km) – dialogboks med advarsel "manglende informationer"</p> <p>2.b Ved indtastning af forkert bilmærke, fremhæves teksten og en tekstboks popper op og fortæller "ugyldigt bilmærke"</p> <p>2.c Ved indtastning af negativ eller slet ingen pris → Trykkes der gem vil der komme en dialogboks med advarslen "ugyldig pris"</p>
Post betingelser	Et nyt bil-"objekt" er oprettet i vores katalog
Author	Nikolaj, Ian, Ahmed, Tobias og Jonas

Prioriteret Liste af FURPS

1) Functionality (funktionelle krav)

- Er det vigtigste da det er her vi beskriver de funktionelle ting i vores system

2) Performance (non-funktionelle krav)

- Det er vigtigt at vi har en hurtigt performance, så det ikke tager flere timer at oprette en annonce, samt redigere i en.

3) Supportability(non-funktionelle krav)

- Hvis det skal ændres noget, er det vigtigt at det kan gøres relativt hurtigt.

4) Reliability (non-funktionelle krav)

- Da vi allerede har beskrevet de potentielle fejl i vores system – hvilket medfører en lav fejlfrekvens, som gør at reliability ikke er prioriteret

5) Usability (non-funktionelle krav)

- Eftersom det er salgs personerne og administrationen der skal bruge systemet, samt at systemet ikke er særlig komplekst, så er usability det mindst prioriteret punkt på listen.

Risikoanalyse

Her kigger vi på de forskellige risici der kan forekomme under projektet, såsom sygdom, fravær m.m.

ID navn	Risiko navn	Beskrivelse	Påvirkning	Sandsynlighed	Ansvarlig	Next step
UC_01	Sygdom	Hvis en medarbejder er syg og fraværende	Den resterende del af teamet vil ikke have indsigt i den pågældendes arbejde	M	Teamlederen	Konstante briefings om hvor langt man var nået og hvad man er i gang med
UC_02	Systemfejl	Hvis nuværende arbejde på anden vis mistet	Hele projektet kan risikere at gå ned	L	Alle i teamet	Jævne backups
UC_03	Misforståelser	Misforståelser mellem teamet og kunden	Man kan evt. udvikle et produkt kunden ikke ønskede	M	Teamlederen	Konstant kontakt mellem kunde og team

Introduction – Shared vision

En oversigtsplan som gennemgås med kunden, som så kan give feedback og godkende de løsninger vi er kommet op med og komme med evt. rettelser.

Positioning

Problem Statement

The problem of	<p>Mr. Trucker sender faktura ud via brev og udregner salgskommission hver 14. dag i hånden.</p> <p>Sælgerne laver salgskontrakter på papir, og gemmer kundeinformationer i ringbind.</p> <p>Når sælgeren skal se om der er en bil hjemme, går han ud og kigger på parkeringspladsen.</p>
affects	<p>Chefen</p> <p>Sælgere</p>
the impact of which is	Da alt gøres manuelt, tager det længere tid.
a successful solution would be	Mr. Trucker samt sælgere, vil komme til at bruge mindre tid på manuelt arbejde, da deres manuelle processen er blevet erstattet af IT-systemet.

Product Position Statement

For	Mr. Truckers bilforretning
Who	Han har behov for et nyt it system
The (product name)	IT-System
That	Firmaets ansatte bruger mindre tid på manuelt arbejde
Unlike	Køb et allerede eksisterende system hos en anden konkurrent
Our product	Vores system bliver skræddersyet til firmaets behov

--	--

Stakeholder Summary

Name	Description	Responsibilities
Mr. Trucker	Chef for firmaet	Han skal sørge for at systemet holder nemt det ved lige
Sælgere	Sælger biler for firmaet	Oprette biler i systemet, lave faktura og servicere kunderne

User Environment

Overordnet set bruger firmaet meget tid på manuelt arbejde i form af kundehåndtering m.m.

F.eks. Når en kunde kommer ind for at få service, tager det lang tid at finde kundens information i ringbindet, som senere også skal bruges til fakturaen

Product Overview

Needs and Features

Need	Priority	Features	Planned Release
Oprette ny bil med skabelon	Høj	Man kan trykke opret ny bil, hvor der kommer en skabelon ud man kan udfylde	30/4/17

Gemme informationer om nye og gamle biler	Høj	Gemme informationer i systemet (typer, pris, model m.m.)	30/4/17
Udregne salgskommission	Middel	Systemet udregner automatisk salgskommissionen på baggrund af indberettet salg af sælger	30/4/17
Lave faktura	Høj	Udsteder faktura elektronisk (e-mail)	30/4/17

Other Product Requirements

Her beskriver vi hvilke non-funktionelle krav der er til systemet.

- Brugervenligheden skal være høj.
- GUI skal være enkelt.
- Flot design

Requirement	Priority	Planned Release
Brugervenlighed	Høj	
GUI	høj	
Flot design	Lav	

Feasibility analyse

Legal feasibility

Da vores system ikke overskrider nogle love, er det lovligt at lave.

Schedule feasibility

Da vi selv har bestemt vores deadline, har vi en bedre fornemmelse af hvor meget vi skal nå og hvad vi skal lave i de forskellige tidsrum. Det er derfor muligt at nå at færdiggøre forskellige iterationer indenfor forskellige deadlines.

Cultural feasibility

Lokalt og generelt har vores IT-system ikke den store påvirkning. I forhold til firmaet påvirker det arbejdspladsen idet der kan spares ressourcer på at mindske manuelt arbejde.

Resource feasibility

Resource er det ikke dyrt, da selve programmet bliver lavet i et program på computeren. Der skal ikke betales for større objekter.

Technical feasibility

Vi ansætter selv vores team alt efter, hvad vi mangler. Dvs. mangler vi programmør, testere eller andet, så sørger vi selv for at de enkelte lever op til de tekniske krav.

Economical feasibility

Da vi kan lave programmet for ca. 10.000 kan vi sagtens sælge det for en million kroner. Det er et program der er efterspurgt af mange kunder og derfor kan vi sætte en høj pris.

Elaboration

Beskrivelse af Elaboration phase

Man bruger Elaboration fasen til at få en bedre forståelse og flere detaljer i forhold til de krav kunden har sat.

I Elaborations phase bruger vi tid på at få vores fundament til applikationen op at køre. Det vil sige at vi skaber en baseline for vores arkitektur som i dette tilfælde er brugen af MVVM og Per Laursen's Frameworket "The MVVMStarter project".

Vi udvikler altså de kritiske krav først, som giver størst værdi til forretningen. Udover det skal vi planlægge forskellige iterationer med så præcise datoer som muligt, så vi kan give kunden mulighed for at komme med rettelser og nye ideer.

Use cases

Beskrivelse af use-case tekst

En Use-case tekst, er en beskrivelse af de use-cases der indgår i projektet, som bl.a. visualisere hvorfor man har lyst til at udfører handlingen (trigger), hvordan et normalt flow forløber igennem systemet (normal flow) m.m.

Vi har gennemgået dem med mr. Trucker og han har kommet med nogle indvendinger, som har gjort at vi har redigeret vores use-case tekst.

Opsummering

Use cases der laves use case tekst for er følgende:

- Opret bil
- *Opret kunde (se bilag 1)*
- *Opret sælger (se bilag 2)*
- Opret salg
- Opdatere og slette information

Opret salg

Use Case	Create sale
ID	UC_04
Summary	Når vores sælger har solgt en bil og derfor skal oprette et salg i systemet
Primær Aktør	Sælger
Sekundær Aktør	
Generalization	
Include	
Extend	
Pre-betingelser	Systemet
Trigger	En sælger har solgt en bil og skal derfor oprette et salg
Normal flow	<ol style="list-style-type: none"> 1. Bruger: Man vælger den bil der skal sælges i en dropdown-menu. Systemet: Returnere den valgte bil 2. Bruger: Udfylder informationer (KundeID og salespersonID) med dropdown og confirmer salget. Systemet: Informationerne bliver gemt i et salgskatalog og kommissionen bliver udregnet.
Alternate flow/exceptions	2.a Manglede udfyldelse af informationer i salg dialogboksen (sælgerID eller KundeID) – dialogboks med advarsel "manglende informationer"
Post betingelser	Et nyt salgs-object er oprettet i systemet
Author	Nikolaj, Ian, Ahmed, Tobias og Jonas
Revision og date.	Date 24/04 2017 kl 11:00

Opdatere og slette information

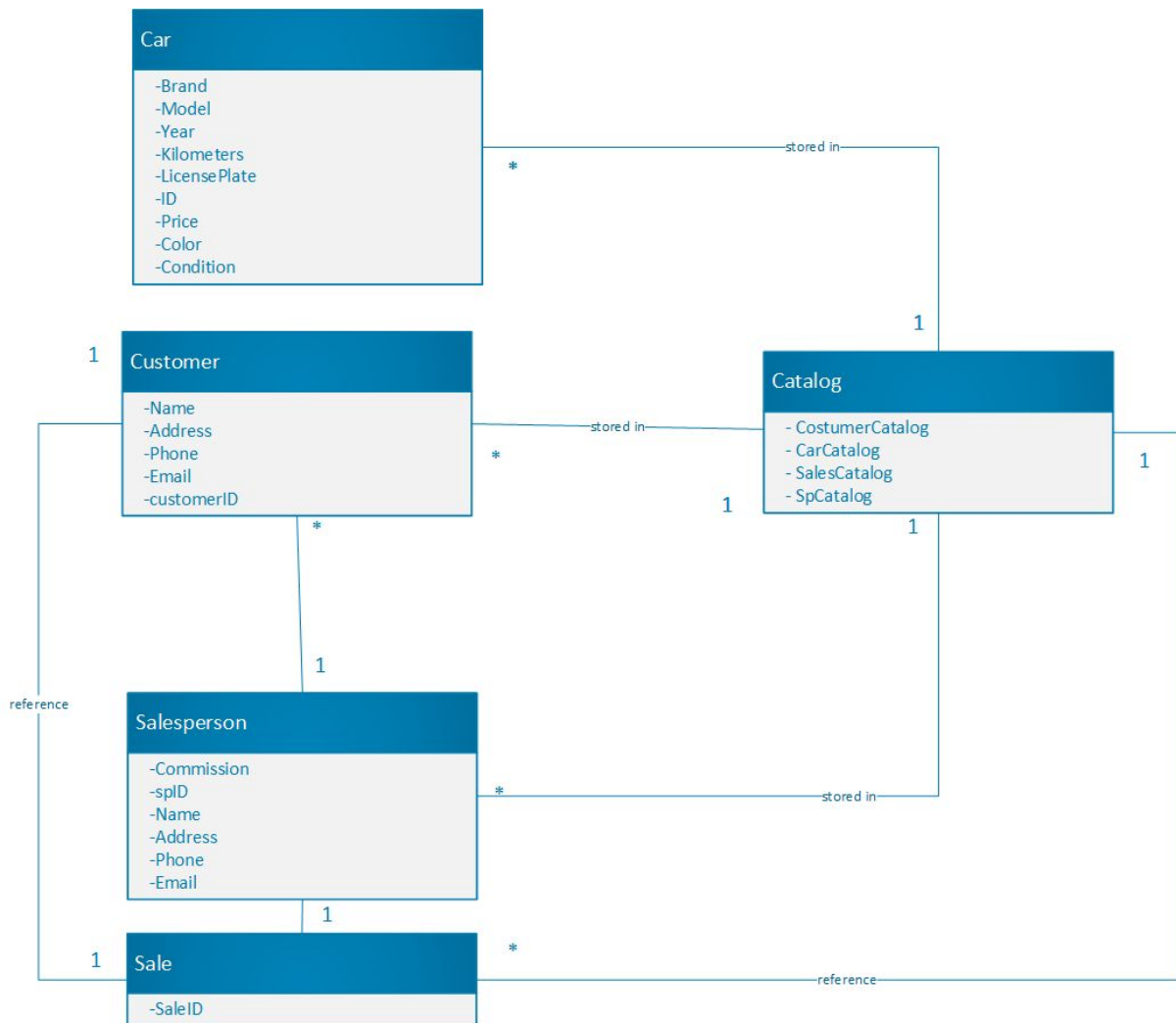
Use Case	Opdatere og slette information
ID	UC_05
Summary	Hvis sælger skal opdatere bil informationer kan det gøres ved brug af "update" knappen. Hvis en kunde ikke længere ønsker at være i systemet længere og derfor ønsker at få slettet sine informationer kan det gøres vha. "delete" knappen.
Primær Aktør	Sælger
Sekundær Aktør	
Generalization	
Include	
Extend	
Pre-betingelser	Systemet samt færdig oprettet kunder og biler
Trigger	Ændring og sletning af information
Normal flow	<ol style="list-style-type: none"> Bruger: Primær aktør vælger knappen "update" Systemet: Returnere en side med udfyldt information Bruger: Indtaster ny information og trykker "update". Systemet: Ny information bliver gemt.
Alternate flow/exceptions	
Post betingelser	Der er slettet og rettet information hos kunden og sælgeren
Author	Nikolaj, Ian, Ahmed, Tobias og Jonas
Revision og date.	Date 24/04 2017 kl 11:00

Domain model

Beskrivelse

En domain model illustrerer værdifulde **conceptual objects**² i **problem domain**³. (problem domain er vores use cases, opret bil, opret sælger og opret kunde. Vi ønsker altså at kigge på de problemer der kan opstå inden for dette scope).

Individuelle domain modeller for henholdsvis bil, salgsperson, customer og salg kan ses i bilag 3.



² Conceptual objects: Betyder at de objekter der arbejdes med, er virkeligheds relateret, fx en bog, en bil m.m.

³ Problem Domain: Det scope man prøver at løse et problem inden for.

System Sequence Diagram - SSD

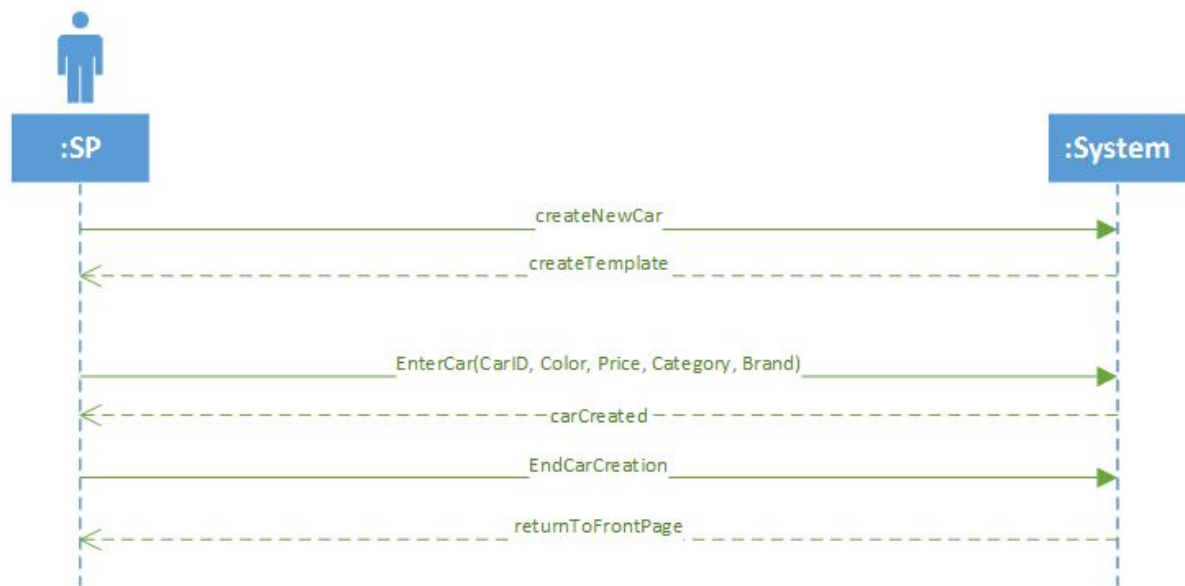
Beskrivelse

Et System Sequence Diagram er et diagram der viser et scenarie for en use-case. SSD'en viser events og hvilke beskeder (metoder) vores aktør sender til systemet, og hvilke beskeder (metoder og properties) vores system returnere. Man viser også hvis der er evt. loops.

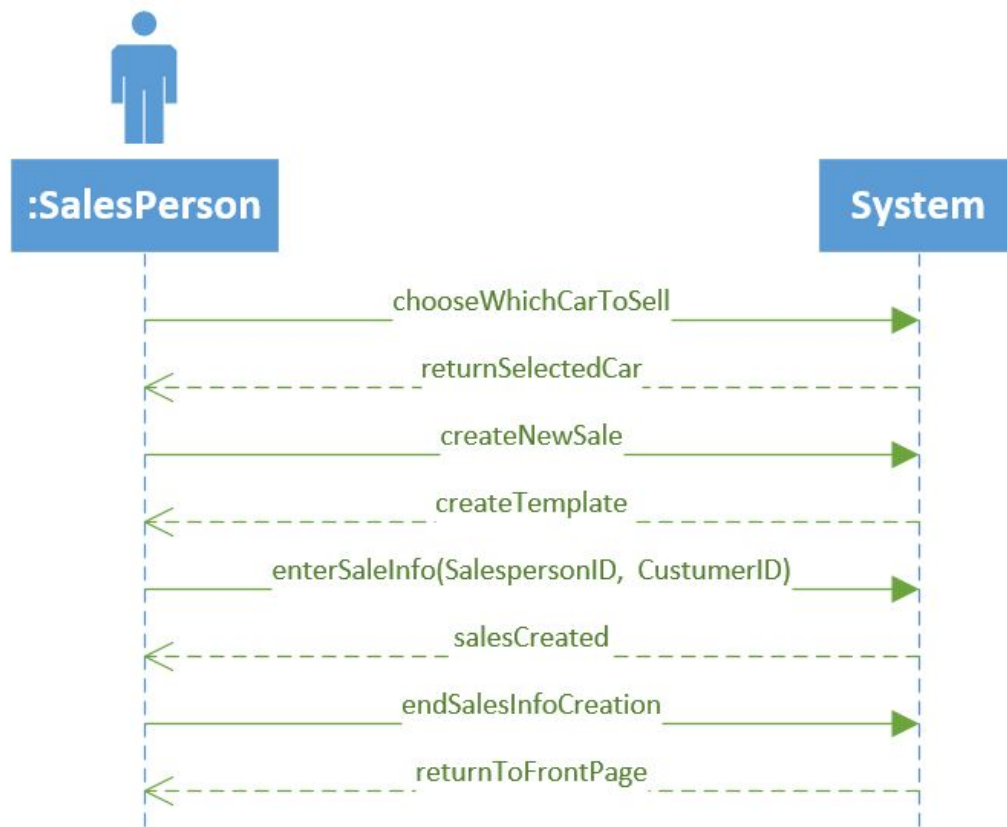
- Med andre ord, vi gør vores use-case visuel.

Se også SSD'er for customer og salesperson i bilag 4.

SSD: Car



SSD: Sale



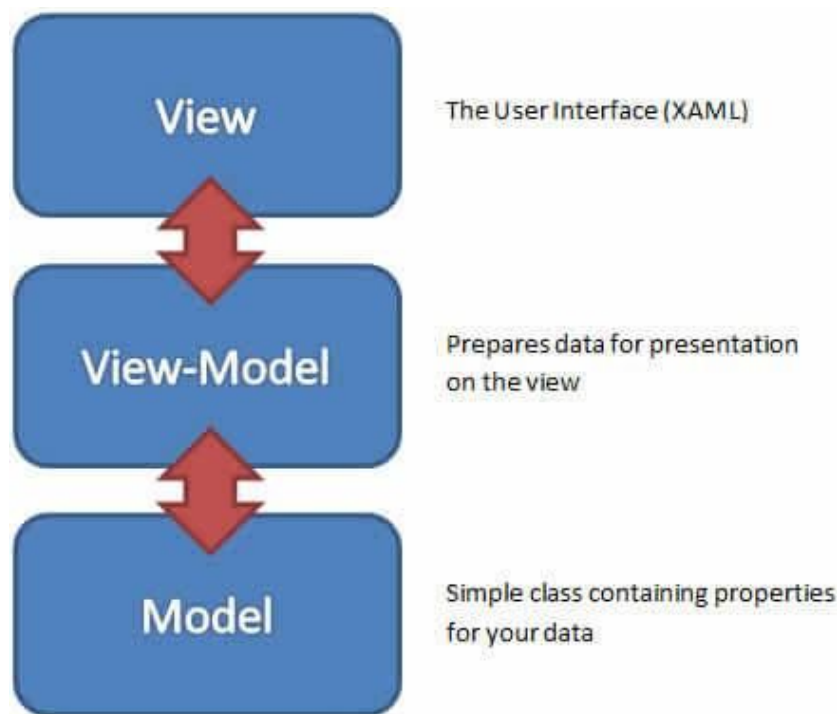
Arktitektur

Beskrivelse

Der findes forskellige arkitekturer man bruger indenfor softwareudvikling. Vi har valgt at arbejde med MVVM.

MVVM – Model-View-View Model

Pointen ved at bruge MVVM er at man bruger en layer arkitektur. Man holder de forskellige lag adskilt. Presentation Layer (View), Logic Layer (ViewModel) og data layer(Model). Fordi vores View er separeret fra vores Viewmodel kan vi sørge for at alt logikken i programmet ligger i vores Viewmodel. Det gør at vi nemt kan udvikle til flere platforme ved hjælp af ingen eller meget lidt ekstra kodning. Et eksempel kunne være at en Windows applikation ville kræve meget lidt for at skulle blive udgivet til Windows Mobile eller Silverlight.



Framework

Vi har fået udleveret et framework der skulle danne ramme om vores projekt, samt at give os nogle helt basale byggesten såsom base-classer, validering, skabeloner, filsystemer, funktionelt GUI udkast m.m.

- Alt ovenstående har vi selvfølgelig haft fuld råderet over, med hensyn til modificering af enkle ting.

Vores View-models er lavet ud fra skabeloner, så når vi senere skal kigger på vores design class diagram, undlader vi disse samt views.

Codebehind

Eftersom vi arbejder med MVVM arkitekturen, er det som regel ikke optimalt at skrive code behind medmindre det er højst nødvendigt.

For at gøre vores program endnu mere brugervenligt valgte vi at lave vores create, update, delete i en Combobox. Eftersom Comboboxen ikke har en XAML Property som sender en command, var vi nødt til at skrive en lille smule code-behind.

Dette gør at comboboxen sender en command, når man vælger en string som er magen til den string vi spørger efter i vores kode.

Operation Contracts

Beskrivelse

Man bruger operation contracts til at forklare at der sker en ændring i systemet. Det vil definere hvad, hver operation gør. Operations finder man i sit SSD, det kunne f.eks være createNewCar.

Se også i bilag 5, operation contracts for Customer og Salesperson.

Operation Contract - Car

Operation :

- createNewCar (button)
- insertInformation(brand, model, year, color, price, condition, License plate) (method)
- endSalesperson (exit)

Cross references Opret ny car.

Preconditions Der er en ny car der skal oprettes og gemmes i systemet.

Postconditions The state of objects in the Domain model after completion, 3 steps.

- **Instances there were created**
 - Instance of a car were created
- **Association there were formed or broken**
 - The car is saved in a catalog, so we have a association between our car instance and our catalog.
- **Attributes who changed is**
 - brand, model, year, color, price, condition, License plate – has now been set to a specific value.

Operation Contract - Sale

Operation

- createNewSale (button)
- insertSalesID (SalespersonID, CustomerID) (method)
- endSale (exit)

Cross references Opret nyt salg.

Preconditions Der er en bil, som skal sælges og oprettes samt gemmes i systemet.

Postconditions The state of objects in the Domain model after completion, 3 steps.

- **Instances there were created**
 - An instance of sale were created.
- **Association there were formed or broken**
 - The sale is saved in a catalog, so we have an association between our sale instance and our catalog.
- **Attributes who changed is**
 - SalesID – has now been set to a specific value.

GUI & Prototype

Beskrivelse

Vi laver Prototyper som ligner det færdige produkt, så vi på en visuel måde kan se hvordan det færdige resultat vil udforme sig.

Se også bilag 6 for customer, salesperson og load

Program brugt til fremstilling

- Axure RP 8

Prototype type

- **High fidelity**

Prototype typer

High fidelity

- Prototyper der ligner det færdige produkt, laves i live programmer, hvor det er click and drag – der er ingen implementering. (evt. programmer: Axure)

Low fidelity

- Skitser af det færdige produkt, hvor der stadigvæk mangler nogle detaljer.

Nedenstående ses en prototype af henholdsvis Car og Sale view fra vores MVVM. Andre screenshots af Customer, Salesperson og load ses i bilag 8.

The image displays two screenshots of a software application, likely a car dealership management system, showing different views of a data model.

Top Screenshot (Car View):

- Header:** "Touren's Car dealership" (top left), navigation tabs: "Customer", "Car" (selected), "Salesperson", "Sale", "Load", "Save", "Quit" (top right).
- Form Fields:**
 - Choose an option: Create ~
 - Brand: (Brand)
 - ID: 1
 - Licenseplate: (License)
 - Model: (Model)
 - Year: 0
 - Kilometers: 0
 - Price: 0
 - Color: (Color)
 - Condition: (condition)
- Buttons:** Create, Update, Delete (bottom left).
- Content Area:** A large light blue rectangular area on the right side of the form.

Bottom Screenshot (Sale View):

- Header:** "Touren's Car dealership" (top left), navigation tabs: "Customer", "Car", "Salesperson", "Sale" (selected), "Load", "Save", "Quit" (top right).
- Form Fields:**
 - Choose an option: Create ~
 - Date: January 1 1917
 - Car: (License) (Model)
 - Customer: Bob Jones CustomerID
 - Salesperson: John Doe
- Buttons:** Create, Update, Delete (bottom left).
- Content Area:** A large light blue rectangular area on the right side of the form.

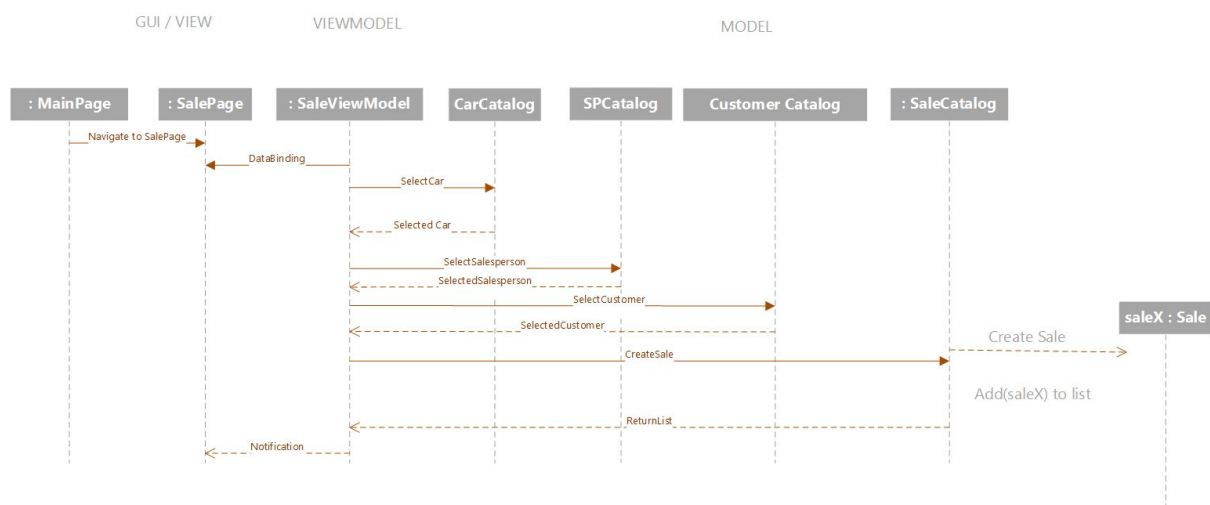
Design Sequence Diagram

Beskrivelse

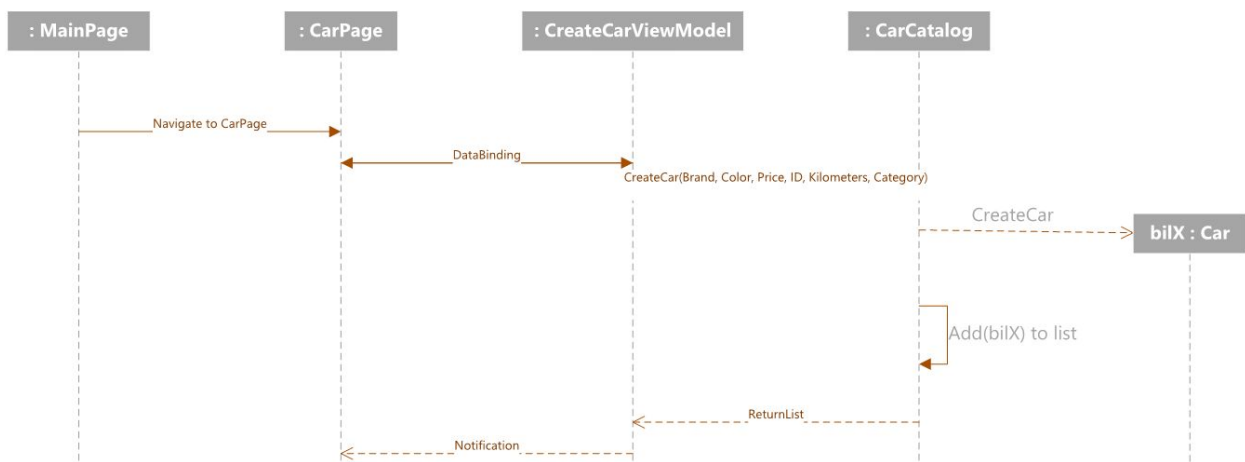
Vi har i ovenstående kapitel om SSD kigget på hvilke events og hvilke beskeder vores primære aktør sender til systemet, og hvilke beskeder vores system returnere. I Design Sequence diagram "åbner" vi den såkaldte "black box" og ser hvordan de forskellige objekter i systemet arbejder sammen med hinanden

Se også i bilag 7, operation contracts for Customer og Salesperson.

Design Sequence Diagram for Sale



Design Sequence Diagram for Car



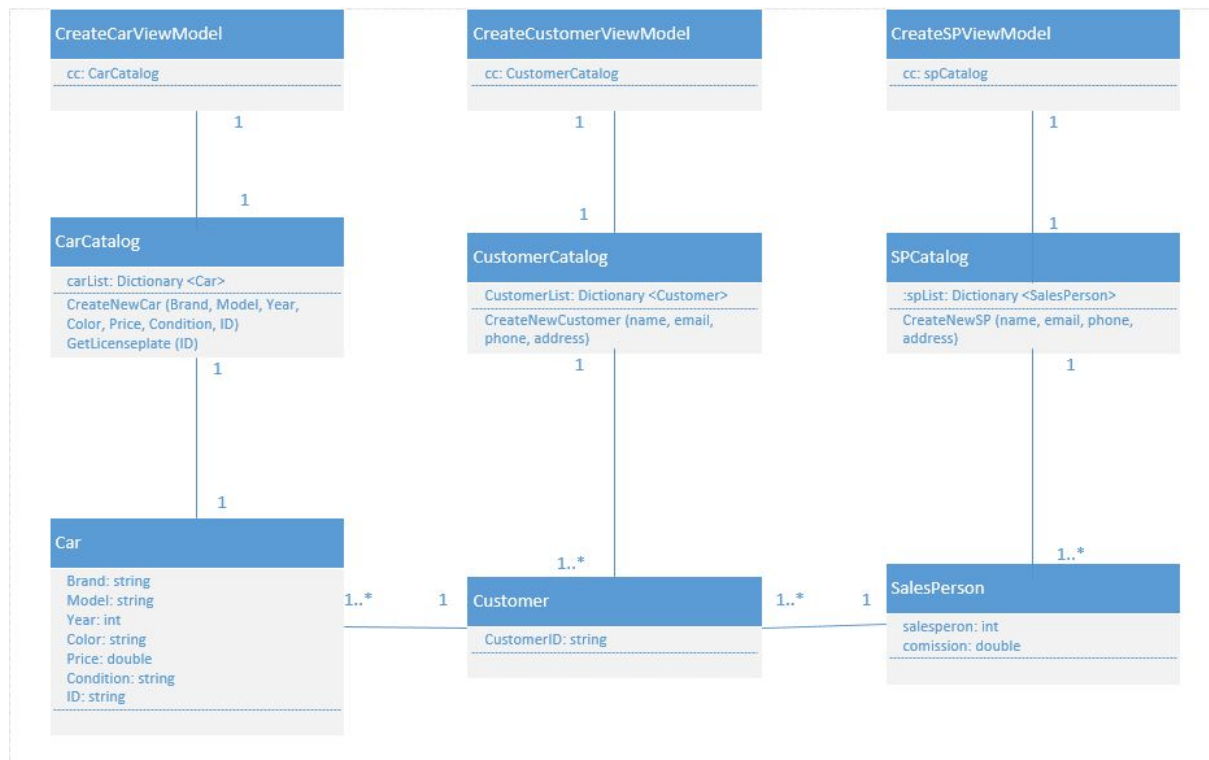
Design Class Diagram

Beskrivelse

Når man skal lave et "class design diagram" kigger man på "Design Sequence Diagram" her ser man hvilke klasser der sender hvilke metoder og hvilke klasser der har referencer til hinanden.

What's included in a "design class diagram" object?

- Name of the class
- List of attributes
- Methods the class contains



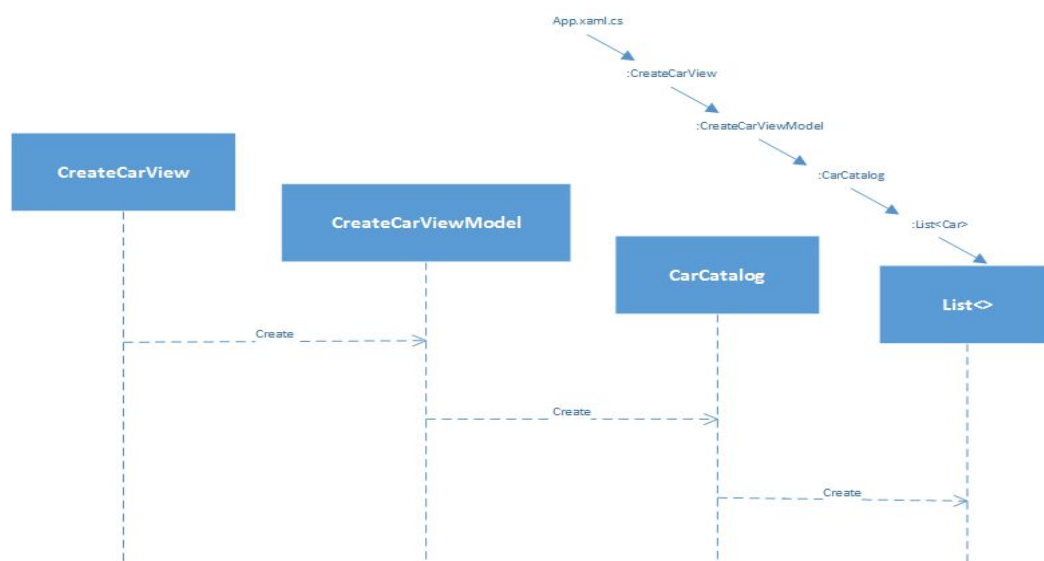
Start Up Design Sequence Diagram

Beskrivelse

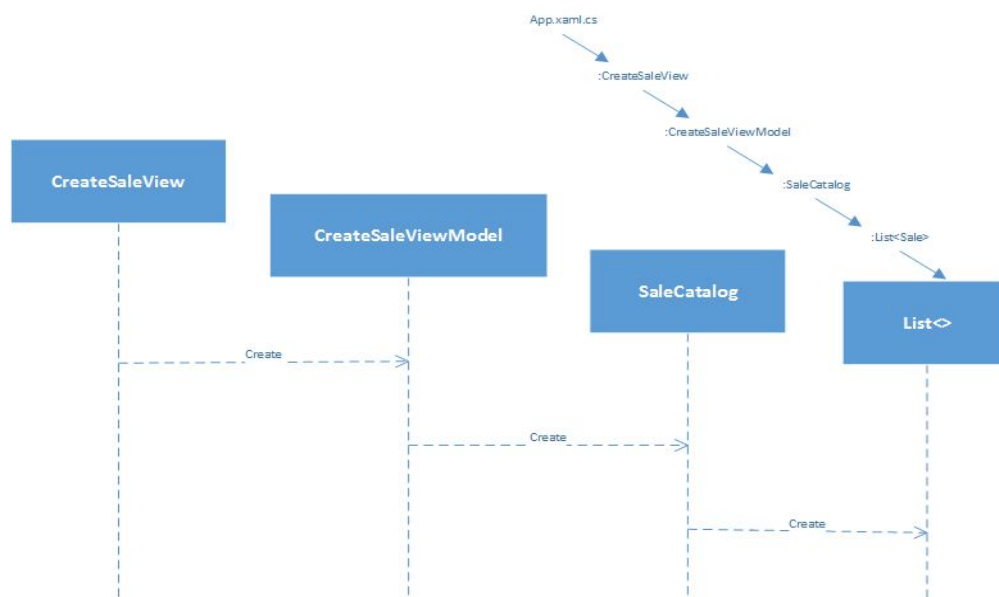
Man bruger et Start Up Design Sequence Diagram, til at vise hvad der kommer til at ske når applikationen starter op. Her viser vi at når CreateCarViewet bliver oprettet, laver den en Viewmodel, som så laver et katalog, som så laver en liste.

Se også i bilag 8, operation contracts for Customer og Salesperson.

Start Up Design Sequence Diagram - Car



Start Up Design Sequence Diagram - Sale



Evaluering af det færdige produkt

Formelt

Det færdige projekt lever op til alle de krav vi har fået fra vores kunde, derudover er der blevet finpudset lidt på GUI'ens udseende som gør vores program mere visuelt overskueligt.

Generelt

Vi har gennemarbejdet Inception fase og elaborations fase, så havde der været længere tid til rådighed, ville vi gå videre til construction, her ville vi kigge mere på:

- Search mellem bilerne
- Mere validering af de værdier man indtaster under oprettelserne (kunde, sælger og bil)
- "Bedre" menu (skulle lave som ren dropdown menu)
- Implementere et bilvask tilbud som en til og fra funktion.
- Hvordan skal der tages imod betaling? (kort, kontant, mobilpay)
- Hvad skal der ske med en evt. faktura?
- Finpudse vores nuværende system: se om der er nogle enklere måde at lave det vi har lavet på, implementere brugervenlige features (search, strukturering af list views, Billede af medarbejdere vises i beskrivelsen af en medarbejder m.m.)
- Tage stilling til hvad der skal ske når en bil fx er blevet solgt: skal der sendes en faktura, er det kontant betaling eller skal der integreres noget online betaling?

Dog har vi på nuværende tidspunkt ændret en del i frameworkets GUI, samt lavet små detaljer såsom: En anden startup baggrund, modificeret knapper, identiske sidder (mht højde, længde, skriftstørrelse) og validering af telefonnumre under oprettelsen.

Evaluering af projektet

● Diskussion af Inception fase længde

Vores Inception fase er meget længere end hvad den ville være i den virkelige verden - grunden er selvfølgelig at vi lige skulle stifte bekendtskab til alle de forskellige metoder og fremgangsmåder der er indenfor UP, samt at se det på andre måde at gøre det på (waterfall method, agile m.m.)

Evaluering af gruppearbejdet

Pros	Cons
<ul style="list-style-type: none">• Overholdelse af deadlines• God arbejdsfordeling (2 og 2, samt individuelt arbejde derhjemme)• Ved sygemelding er risikoanalysen overhold, mht skriv til resten af gruppe at man er syg.• Alle gruppemedlemmer er altid klar til at forklare problemstillinger med egne ord.• Gode til at diskutere problemstillinger• Stifte bekendtskab med GitHub• stemning i gruppe er i top	<ul style="list-style-type: none">• Opgaverne der skal uddelegeres er ikke altid det nemmeste at få til at gå op med antallet af person i gruppen.• Meget sygdom

Bilag

Bilag 1: Use case create customer

Use Case	Create Customer
ID	UC_02
Summary	Når vores sælger skal oprette en ny kunde i systemet.
Primær Aktør	Sælger
Sekundær Aktør	
Generalization	
Include	
Extend	
Pre-betingelser	Systemet og kundeinformationer
Trigger	Der er en kunde der skal oprettes i systemet, så kunden kan foretage køb.
Normal flow	<ol style="list-style-type: none"> 1. Bruger: Tryk "opret ny kunde" Systemet: - Opretter skabelon til en ny oprettelse af kunden. 2. Bruger: Udfylder kundens informationer (Navn, telefonnummer, adresse og e-mail). Systemet: - Gemmer informationerne som katalog
Alternate flow/exceptions	<p>2.a Manglede udfyldelse af information på sælgeren (<i>Navn, adresse, e-mail, telefonnummer, kundeID</i>) – dialogboks med advarsel "manglende informationer"</p> <p>2.b Ved indtastning af ugyldigt nummer, manglende @, ugyldig adresse. Dialogboks "ugyldige information, se * "</p>
Post betingelser	Et ny kunde-"objekt" er oprettet i systemet
Author	Nikolaj, Ian, Ahmed, Tobias og Jonas
Revision og date.	Date 15/03 2017 kl 16:00

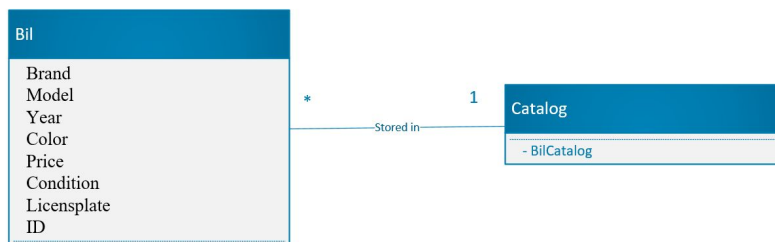
Bilag 2: Use case create salesperson

Use Case	Create salesperson
ID	UC_03
Summary	Når chefen skal oprette en ny sælger.
Primær Aktør	Chefen
Sekundær Aktør	Salesperson
Generalization	
Include	
Extend	
Pre-betingelser	Systemet og salespersonens informationer
Trigger	Der er blevet anset en ny sælger, der skal oprettes i vores system, så chefen kan holde øje med hvem der laver hvilke salg.
Normal flow	<ol style="list-style-type: none"> 1) primær aktør: Chefen skal oprette en "ny ansat" Systemet: Åbner en skabelon i en ekstern fane, til oprettelse af ny salgsperson. 2) primær aktør: Udfylder oplysninger (navn, adresse, email telefonnummer, spID) Sekundær aktør: Giver de nødvendige informationer Systemet: Systemet gemmer oplysninger og giver en confirm "en ny sælger er oprettet" dialogboks. 3) primær aktør: Udfylder oplysninger på en ny sælger mere (navn, adresse, e-mail telefonnummer m.m.) Sekundær aktør: Giver de nødvendige informationer Systemet: Systemet gemmer oplysninger og giver en confirm "en ny sælger er oprettet" dialogboks. 4) primær aktør: Er færdig med at oprette sælgere og afslutter systemet Systemet: Systemet lukker den eksterne fane og går tilbage til frontpagen.

Alternate flow/exceptions	<p>2.a / 3.a Manglede udfyldelse af information på kunden (Navn, adresse, email, telefonnummer) – dialogboks med advarsel "manglende informationer"</p> <p>2.b / 3.a Ved indtastning af ugyldigt nummer, manglende @, ugyldig adresse. Dialogboks "ugyldige information, se * "</p>
Post betingelser	Et nyt sælger-"objekt" er oprettet og gemt i et catalog.
Author	Ian, Ahmed, Jonas, Nikolaj og Tobias
Revision og date.	Date: 15/03-2017, kl 17:20

Bilag 3:

Domain model for bil



Domain model for person



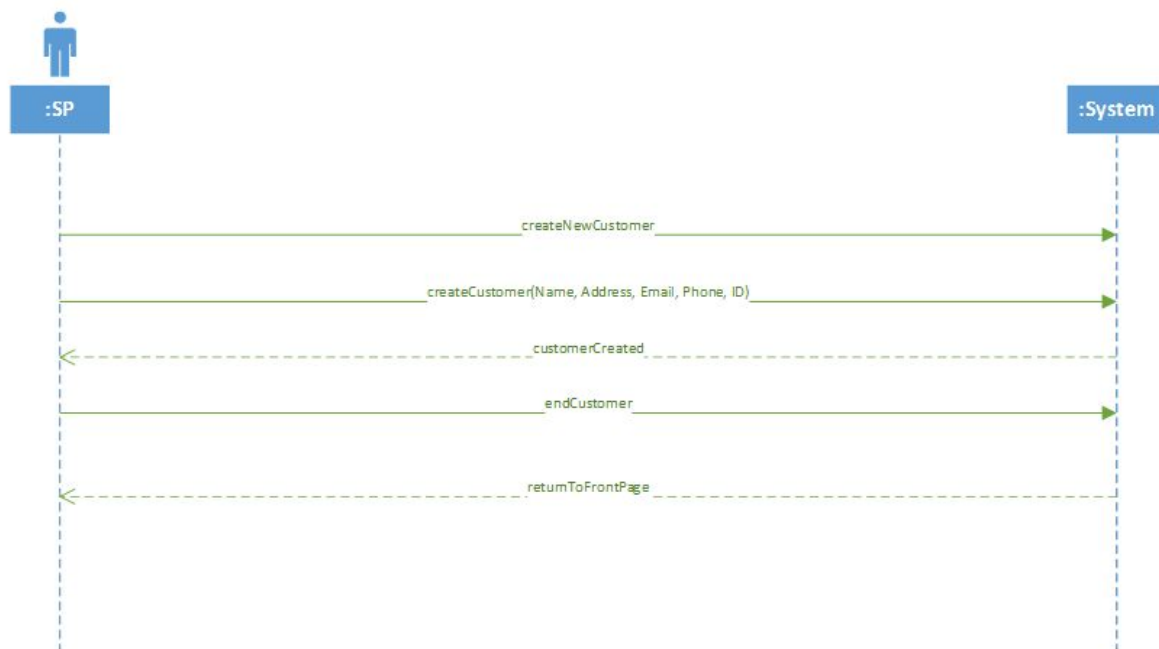
Domain model for customer



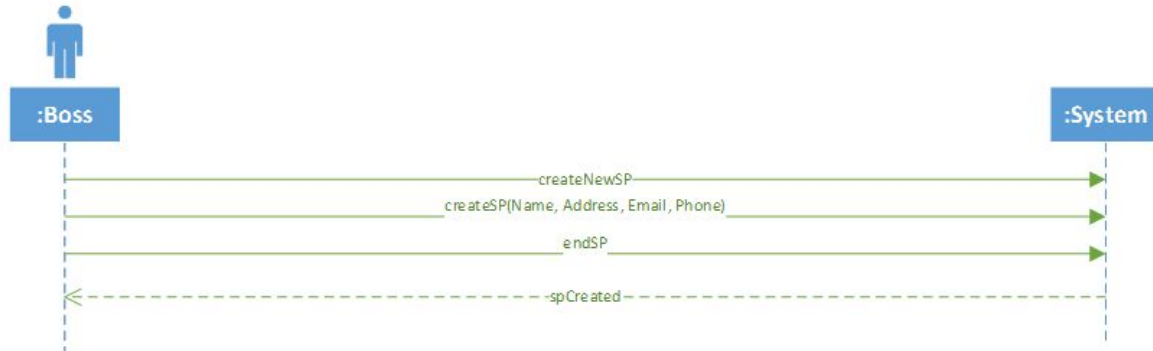
Domain model for sale

Bilag 4

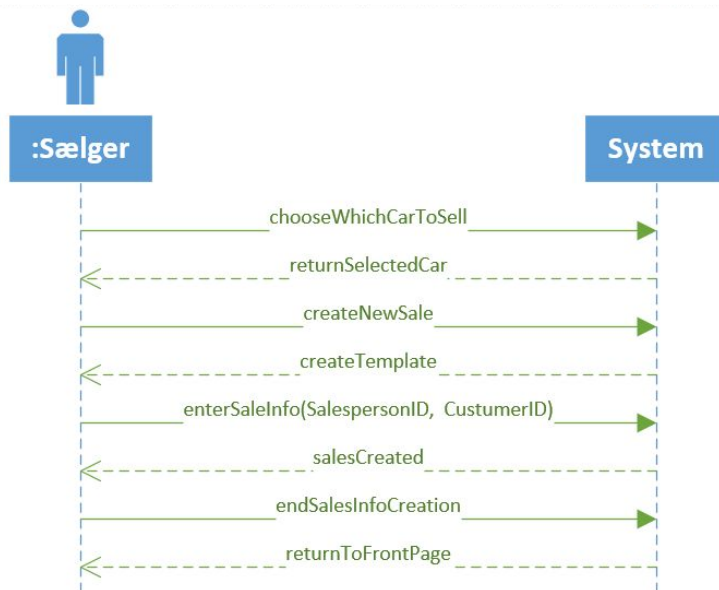
SSD for Customer



SSD for Salesperson



SSD for sales



Bilag 5

Operation Contract - Salesperson

Operation

- createNewSalesPerson (button)
- insertInformation(Navn, Adresse, Telefonnummer, Email, ID) (method)
- endSalesperson (exit)

Cross references Opret ny salesperson.

Preconditions Der er en ny salesperson der skal oprettes og gemmes i systemet.

Postconditions The state of objects in the Domain model after completion, 3 steps.

- **Instances there were created**
 - Instance of a salesperson were created
- **Association there were formed or broken**
 - the salesperson is saved in a catalog, so we have a association between our salesperson instance and our catalog.
- **Attributes who changed is**
 - Navn, Adresse, Telefonnummer, Email, ID – has now been set to a specific value.

Operation Contract - Customer

Operation

- createNewCustomer (button)
- insertInformation(Navn, Adresse, Telefonnummer, E-mail, ID) (method)
- endCustomer (exit)

Cross references Opret ny customer.

Preconditions Der er en ny customer der skal oprettes og gemmes i systemet.

Postconditions The state of objects in the Domain model after completion, 3 steps.

- *Instances there were created*
 - Instance of a customer were created
- *Association there were formed or broken*
 - The Customer is saved in a catalog, so we have a association between our customer instance and our catalog.
- *Attributes who changed is*
 - Navn, Adresse, Telefonnummer, Email and ID – has now been set to a specific value.

Bilag 6

Prototyper af Load, Customer og Salesperson

Trucker's Dealership

Choose an option:

Name	<input type="text" value="John Doe"/>
Address	<input type="text" value="Elisagårdsvej 5, 4000 Roskilde"/>
E-mail	<input type="text" value="Example@Email.com"/>
Phone	<input type="text" value="23954072"/>
SalespersonID	<input type="text" value="0"/>
Commission	<input type="text" value="0"/>

Customer Car **Salesperson** Sale Load Save Quit

John Doe

Trucker's Dealership

Choose an option:

Name	<input type="text" value="Bob Jones"/>
Address	<input type="text" value="kildevej 69, 4000 Roskilde"/>
E-mail	<input type="text" value="Example@Email.com"/>
Phone	<input type="text" value="69696969"/>
Costumer ID	<input type="text" value="CustomerID"/>

Customer Car **Salesperson** Sale Load Save Quit

Bob Jones CustomerID

Trucker's Dealership

Customer Car Salesperson Sale **Load** Save Quit

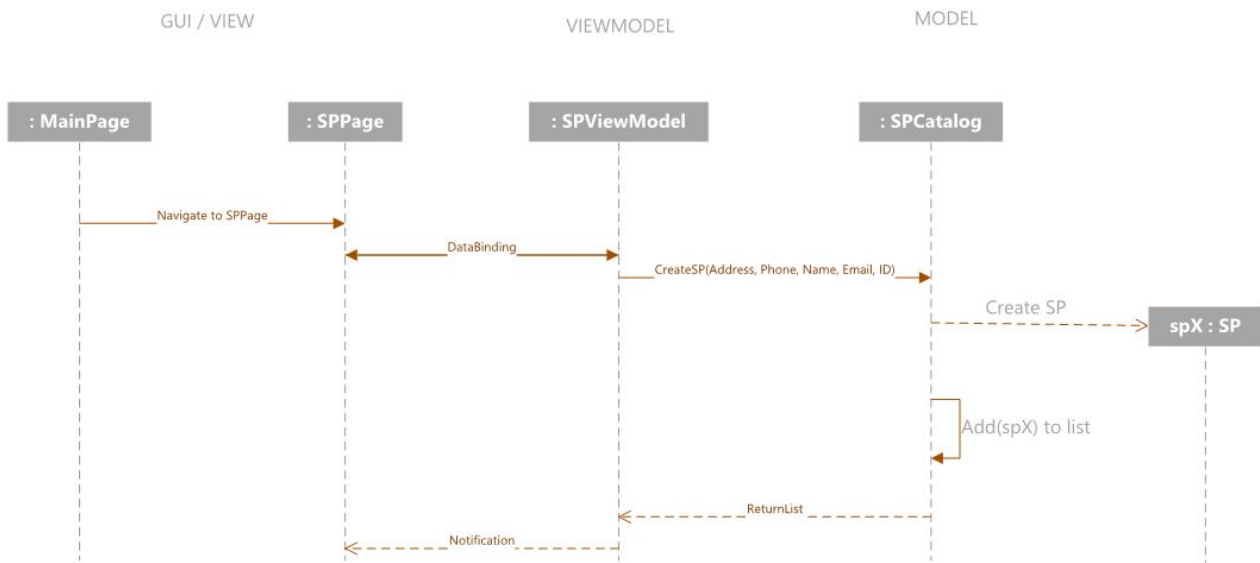
Trucker's Dealership



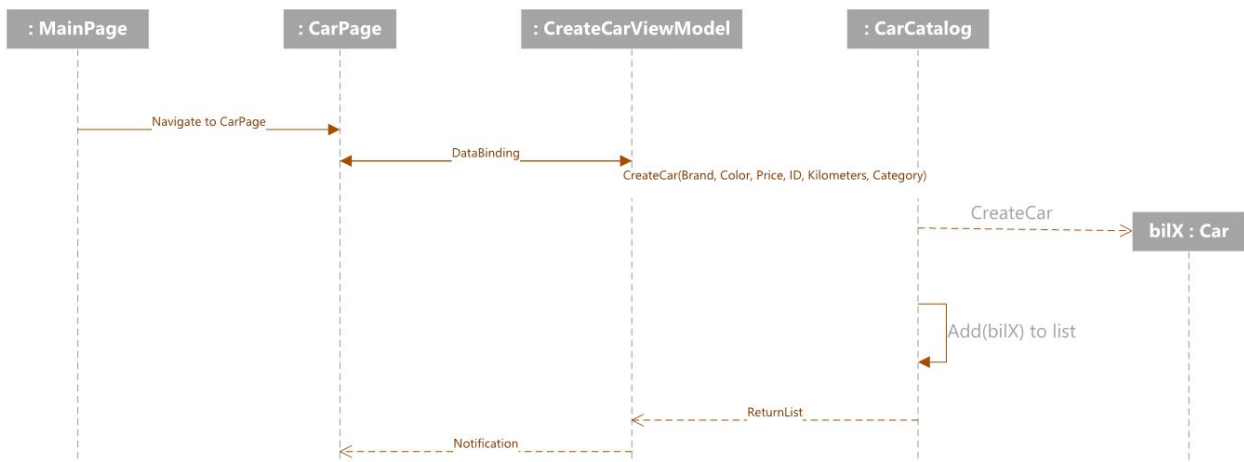
Bilag 7

Design sequence diagram

Design Sequence Diagram for Salesperson

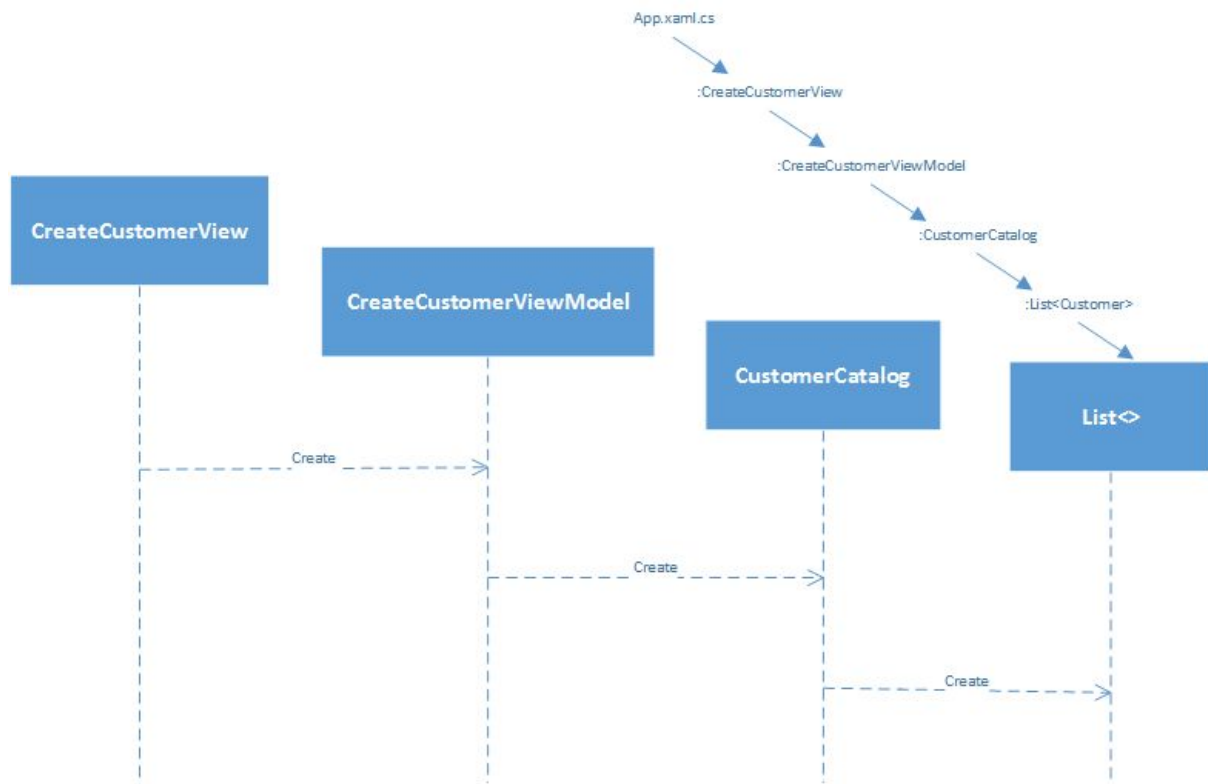


Design Sequence Diagram for Customer

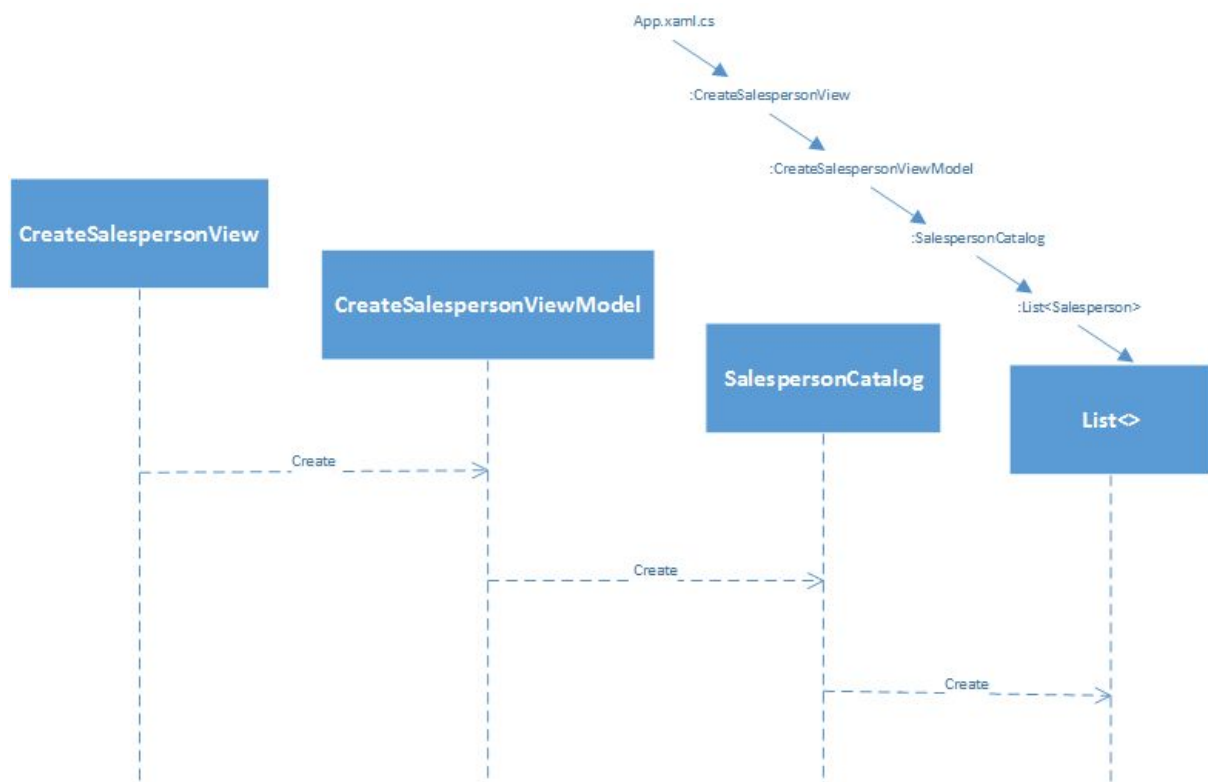


Bilag 8

Start Up App Design Sequence Diagram for Customer



Start Up App Design Sequence Diagram for Salesperson



Bilag 9

Dictionary

Name	Type	Description
Kommission	Forretningsregel	Sælgernes kommission bliver regnet efter total salg
Gem kundeinformation	Forretningsregel	Gemmer information om kunden (navn, adresse)
Create invoices	System regel	Opretter en faktura
Ny annonce	Objekt	Laver en ny annonce så bilen kan blive solgt
Bil	Objekt	Bilen som kan blive købt af kunden
C#	Programmeringssprog	Det programmeringssprog vi har valgt at bruge
Github	Programmerings Cloud	Et sted hvor hele programmet kan blive bygget i små bider
Iteration	Projekt beskrivende	Færdiggøre små bider af projektet
Sale	Objekt	Når der bliver solgt en bil til en kunde.
Codebehind	Programmeringssprog	kode der bliver lavet i GUI'en
Salesperson	Objekt	En sælger der arbejde i virksomheden