AP*Engros*

Udarbejdet af gruppen-Foxtrot

*Elena, Christian, Kasper & Anes*

Erhvervsakademi Dania Skive

*Datamatiker – 1. Semester eksamen 2018*

Undervise & Vejledere:

*Ove Thomsen & Claus Nørregaard*

Indhold

[Indledning 3](#_Toc515529007)

[Problemstilling 4](#_Toc515529008)

[Problemformulering 5](#_Toc515529009)

[Problemanalyse 7](#_Toc515529010)

[Brief Use case 8](#_Toc515529011)

[Fully Dressed Use Case 11](#_Toc515529012)

[Unified Process 12](#_Toc515529013)

[Class Diagram 13](#_Toc515529014)

[Database 15](#_Toc515529015)

[Entity Relationship Diagram 15](#_Toc515529016)

[Mapping Diagram 16](#_Toc515529017)

[Domain Model 17](#_Toc515529018)

[17](#_Toc515529019)

[Konklusion 18](#_Toc515529020)

[Logbog 19](#_Toc515529021)

[Bibliografi 20](#_Toc515529022)

[Bilag 21](#_Toc515529023)

## Indledning

Denne rapport er udarbejdet gennem et 6 ugers eksamensprojekt på Skive Erhvervsakademi Dania 2018.

Eftersom den danske økonomi er kommet i opsving, har det fået flere virksomheder til at søge efter nye indtjeningsmuligheder. Virksomheder har rette fokus mod nye markeder, innovation og forfinelse af produkter-og services på eksisterende markeder. Flere virksomheder ser IT og teknologi som en fremtidige forretning-og konkurrencemæssigt fordel.

Virksomheden AP Engros ønsker at følge med i det økonomiske opsving og bestemt dem for at være en konkurrencedygtig leverandør af VVS-og Sanitetsprodukter til små og mellem store VVS-installatører, badeværelses-og køkkenbutikker.

## Problemstilling

Virksomheden AP Engros er en konventionel handelsvirksomhed der til dagligt og de sidste 20 år har beskæftiget sig med salg-og service af energisystem, solceller, varmevekslere mv.

Det økonomiske opsving i den danske økonomi har fået virksomheder som AP Engros til at søge efter nye indtjening- og investerings muligheder. Mange virksomheder ønsker at gøre sig konkurrencedygtige på nuværende og nye markeder. AP Engros ønsker at følge med opsvinget og satse på fremtidige vækstmuligheder som fremtidige leverandører af VVS og sanitetsprodukter. Virksomheden har rettet fokus mod 3 målgrupper der omfatter små og mellemstore VVS-installatører, badeværelses-og køkkenbutikker. Derved har virksomheden designet et produktkatalog som imødekommer deres 3 målgrupper,

AP Engros nye forretningsstrategi er baseret på anvendelse af IT og automation. Løsningen skal hjælpe virksomheden til at optimere kortlagte og optimerede processer, mindske medarbejderomkostninger mv.

Virksomhedens mål og visioner baseret på en IT-kundeplatform som er følgende:

*”Igennem vores skræddersyede IT-Kundeplatform, konkurrencedygtige priser og exceptionelle service, vil vi være den fortrukne leverandør af VVS og Sanitetsprodukter til Køkken-og badeværelsesbutikker, samt små og mellemstore VVS-installatører”*

Der ligger flere fordele i at virksomheden vælger at få implementeret en IT-kundeplatform, eftersom det kan presse deres omkostninger ned samt priser og gøre sig konkurrencedygtige på nuværende markeder og derved opnå en større kundegruppe. Med større kundegruppe og indtagelse af nye markeder, fremmes virksomhedens indtjeningsmuligheder og virksomheden opnår økonomisk vækst.

## Problemformulering

Opgaven er at design-og implementere en IT-platform, der skal håndtere køb, salg-og service opgaver og de processerne der mellem virksomheden AP Engros og deres kunder.

AP Engros har en række funktionalitetskrav som skal design-og implementeres, dog har virksomheden ikke den store erfaring med digitale løsninger men gode med traditionel handel.

De konkrete krav som virksomheden stiller, er at der ønskes en kundeadministration som skal håndtere virksomhedens kunder samt deres oplysninger. Brugeren af programmet skal have mulighed for at kunne oprette, slette og redigere i virksomhedens kundeoplysninger samt detaljer om hver enkelte kunde.

Det indbefatter navn, adresse, kontaktoplysninger, kontaktperson og kundes forventet årlige omsætning.

Dernæst skal der være en aftaleadministration der kan oprette, slette og redigere aftaler der er mellem enkelte kunder og AP Engros. En aftale har en løbetid på min. 6-24 måneder, derefter skal kunden skal anmode om ny aftale, for at få adgang til produktgruppe og særlige rabatter.

Virksomhedens produktkatalog indeholder ca. 30.000 produkter, hvilket gør det relevant at, der er implementeret en produktadministration med søgemuligheder, der kan finde produkter ved at søge på f.eks. Produktnavn, produktnummer eller beskrivelse. Der skal også være mulighed for, at AP Engros kan laves masseændringer i pris, på udvalgte produktgruppe i deres produktkatalog.

Der vil ugentligt blive sent produktfiler fra AP Engros leverandører. Filerne bliver gemt i virksomheden indbakke på deres Linux-serve system. Dertil ønskes det, at filerne bliver automatisk indlæst og registret, når virksomheden modtager dem fra leverandøren. Det er vigtigt at filerne ikke bliver slettet fra AP Engros indbakke efter indlæsning af nye.

En abonnementsadministration skal holde styr på hvilke aftale der blive indgået mellem AP Engros og deres kunder. De kunder som har et aktivt abonnement, sender AP Engros en CSV fil, der indeholder produkter kunden har indgået aftale om. Er aftale inaktiv eller udløbet vil kunden ikke modtage filer med produkter, før der anmodet om fornyelse af aftale. Alle filer som bliver sent ud til kunder med et aktivt abonnement, vil ligge i udbakken på AP Engros test-Linux server system.

Med et produktkatalog der indeholder 30.000 produkter, ønsker AP Engros at føre statistik over b.la. Oprettede aftale samt afviklede kontrakter. Der skal også føres statistik over top 10 produktgrupper og mulighed for at AP Engros selv kan vælge top x over produktgrupper distribueret ud til deres aktive kunder.

Alle handlinger der bliver fortaget, skal logges i systemet. Den skal holde styr på seneste inddateringer der er foretaget og antallet af data der er blevet inddateret. Systemfejl logges også i systemet, så man på senere tidspunkt kan udbedre de systemdele som var skyld i fejlen.

### Problemanalyse

Gruppen har gennem de sidste 6 uger arbejde med systemudvikling og implementering af en softwareløsning til AP Engros.

Problemformuleringen omhandlede en virksomhed, der havde et ønske til at få implementeret en IT-løsning til deres kommende salg-og service. AP Engros har ønsket at øge deres omsætning ved at indtage nye markeder, ved at blive dygtige leverandører at VVS- og Sanitetsprodukter til små og mellemstore virksomheden.

AP Engros har ingen erfaringer indenfor IT og derfor ønsker hjælp med hensyn på design og implementering, hvor de stiller krav til hvilke funktionalitet programmet skal kunne indeholde.

Vi har arbejdet på at designe et overskueligt bæredygtig software design, så det nemt for AP Engros at håndtere programmet til at afvikle deres behov. Vi i gruppen diskuteret hvilke elementer der var nødvendige at implementere og hvordan User-Interface skulle implementeres.

Der er anvendt en del systemudvikling inden vi i gruppe har fortaget os implementering af selve Code-behind og programmet.

MANGLER lidt uddyb…

### 

### Brief Use case

Vi har anvendt systemudviklingsværktøj ” Brief Use Case” til at beskrive de krav der er blevet stillet til udvikling af AP Engros software løsning. Hver af de nedenstående Brief Use Cases indeholder som regel scenarier der beskriver systemets funktionalitets ud fra de krav der er blevet stillet. Brugerne skal kunne via. af programmet kunne løse specifikke opgaver, som systemet skal håndtere.

#### Kundeadministration

#### Oprette kunde

Virksomheden er den essentielle bruger af softwareprogrammet. Derfor opretter AP Engros selv deres kunder med relevante kundeoplysninger, hvorefter de bliver gemt i virksomhedens kundedatabase system.

#### Slette kunde

Det er kun virksomheden der kan ændre i deres kundesegment, eftersom de er de eneste der har adgang til programmet og dermed deres database med oprettede kunder.

#### Redigere kunde

Hvis en af virksomhedens kunder f.eks. har fået ny adresse, telefonnummer osv. Indhentes kundes nuværende oplysninger i programmet og ændres. Programmet gemmer og opdatere den pågældende kundes oplysninger til databasen.

#### Aftaleadministration

#### Opret aftale

En oprette aftale er mellem kunden og virksomheden. Kunde bindes i selvbestemt periode over 6-24 måneder. Derved får kunden tildelt rabat på den produktgruppe deres aftale binder over. Et abonnement oprette automatisk når der oprettes en aftale.

#### Slette aftale

Aftaler som er oprettet kan kun slette af virksomheden. Kunden har mulighed for at anmode om sletning af aftale efter 6 måneder eller den bindende periode.

#### Redigere aftale

Hvis der er sket en fejl under inddateringer af oplysninger, har virksomheden mulighed for at redigere oplysninger.

#### Varekatalogadministration

#### Oprette varekatalog

Der skal være mulighed for at virksomheden AP Engros kan oprette nye produkter i deres sortiment inden for VVS- og Sanitetsprodukter.

#### Slette varekatalog

Produkter som f.eks. er udgået eller ikke længere sælge af virksomheden, skal de kunne fjernes fra deres database.

#### Redigere varekatalog

Virksomheden bør ændre produktbeskrivelse og andre relevante informationer, hvis der er sket ændringer i deres varekatalog.

#### Liste produkter

Eftersom der næsten er 30.000 produkter hos virksomheden AP Engros, skal der være mulighed for at kunne søge på forskellige produktgrupper. Man vil kunne søge produkter via. ID, Navn, Kategori, Beskrivelse og Pris.

#### Abonnement administration

#### Aktiv abonnement

En kunde der har en aftale med virksomheden, er samtidig båndet af et abonnement. Det vil sige at kunden har et ”Aktivt abonnement” og vil få tilsendt produktfiler ugentlig fra AP Engros SMTP-Server.

#### Inaktiv abonnement

Er aftalen mellem kunden og virksomheden udløbet eller der slet ikke er oprettede nogen aftale, vil kunden ikke på noget tidspunkt modtage produktfiler fra virksomheden.

#### Statistik

#### Visning

Der skal føres statistik overfor afviklede kontrakter samt visninger af de udvalgte produktgrupper hos virksomheden. AP Engros skal selv kunne være top x over produktgrupper og aftale der er oprettet mellem virksomheden og kunderne.

#### Log

#### Visning

Programmet skal logge handlinger og inddateringer, der bliver foretaget af virksomheden. Det vil sige, at sker der en fejl i en inddatering, kan virksomheden se hvilken inddateringer fejlede.

#### Fejlmeddele

Hvis der opstår et problem i programmet eller der ikke kan udføres en handling, bliver det fejlmeddelelsen gemt i loggen, så der på et senere tidspunkt kan udbedres for systemfejl og andre handlinger, der får programmet til at melde fejl.

### Fully Dressed Use Case

Vi har ud fra vores udarbejdet Brief Use Cases videre udarbejdet dem til *Fully Dressed Use Case*.

Det giver en uddybelse samt bedre indblik af, hvordan vores software løsning skal være konstrueret. Man benytter især disse Use Cases til at specificere, hvordan programmet skal interagere med et objekt i form af f.eks. En bruger.

En *Primary Actor* beskriver hvilket objekt der interager med programmet for at opnå en løsning. I vores tilfælde består *Primary Actor* af en brugeransvarlig. Det betyder at AP Engros er brugeransvarlige, hvilket betyder at, integration foregår mellem AP Engros og programmet. Den brugeransvarlig kan benytte sig af funktionerne: Oprette, slette og redigere i programmet for b.la. kunder, produktgrupper mm.

|  |  |
| --- | --- |
| **UC name** | **Opret kunde profil** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig der ønsker at etablere en profil og gemt i systemet. Læser, der ønsker et overblik over profil. |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Profilen er korrekt oprettet i systemet, gemt i database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Opret Profil” og profils menuen for opretning åbnes. 2. Brugeransvarlig indtaster :  * “User Navn”, ”Password”; * ”Navn”, ”CVR”, ”Adresse”, ”Kontaktperson”, ”Konto oplysninger”, ”Forventet årlige omsætning”;  1. Brugeransvarlig trykker GEM. 2. Systemet gemmer profilen i databasen, returnerer med en godkendelse og lukker profilens menuen. |
| **Extension** | 2a. Brugeransvarlig indtaster ingen data or data af forkert type: ”Navn” ,”Password”, ”Navn”, ”CVR”, ”Adresse”, ”Kontaktperson”, ”Konto oplysninger”, ”Forventet årlige omsætning”;  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier.  4a. Systemet kan ikke gemme profilen i databasen pga. fx ingen internetforbindelse, hukommelsesmangel eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker profilen menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

## Unified Process

**Inception**

Brief Use case

Modelling

Kravspecifikation

**Elaboration**

Uml diagrammer

Xaml opstætning

Full dressed use case

**Construction**

code-behind

implementation

testning

xaml

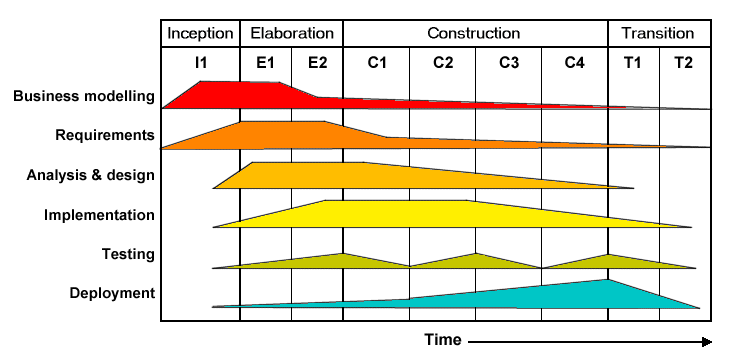
**Transition**

Implantation

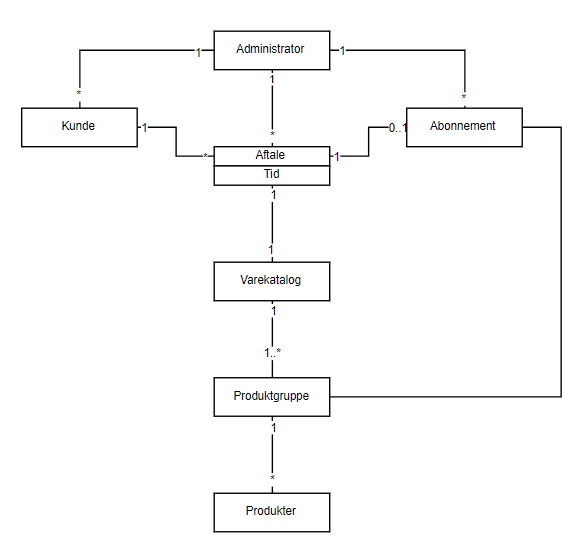
Testning

Kravspecifikation

Aflevering af program



### Domain Model



### Sekvens Diagram

Et sekvens diagram er med til vise det visuelle overblik over interaktioner der følger

mellem objekter der integrerer med hinanden. Man kan bruge sekvens diagrammer til at forudse, hvordan programmet ligeledes vil fungere, uden at have implementeret det.

Når vi skal udarbejde sekvens diagram ud fra vores Class Diagram, benytter vi forskellige symboler til at beskrive processerne og forholdene mellem objekterne i et sekvens diagram.

## Class Diagram

En af de mest benyttede UML diagrammer til at beskrive programmets software struktur, er et Class Diagram som er den fundamentale base for objektorienteret programmering.

Class Diagram er en visuel plan over, hvordan softwaren er struktureret. Det giver en visuel forståelse for, hvordan programmet er struktureret samt over hvilke ting der ligger bag Code-Behind.

Vores class diagram er bestående af 11 klasser som hver især har deres attributter som understøtter deres funktionalitet. Hvis vi ser nærmere på hvordan de enkelte klasser er forbundet med de andre, kan vi se en primær sammenhængende struktur.

En kunde oprettes med følgende oplysninger i systemets database og virksomheden laver en aftale med kundens og deres behov indenfor VVS og sanitetsprodukter. Når der er oprettet en aftale, er kunden som følgende bundet i en selvbestemt periode fra 6-24 måneder. Kunden vil modtage specialtilbud fra virksomheden med specifik produktgruppe som bliver sendt i CSV filformat via. Deres Linux server system.

Virksomheden har mulighed for at ændre i deres kunders oplysninger og aftale via. Den implementeret søgefunktion. Der vil også være mulighed for at fører statistik over de mest solgte aftaler mellem virksomheden og kunden. En log er implementeret til systemet som viser handlinger over forskellige inddateringer og fejlmeddelelser samt systemfejl.

## Class Diagram progress2.pngDatabase

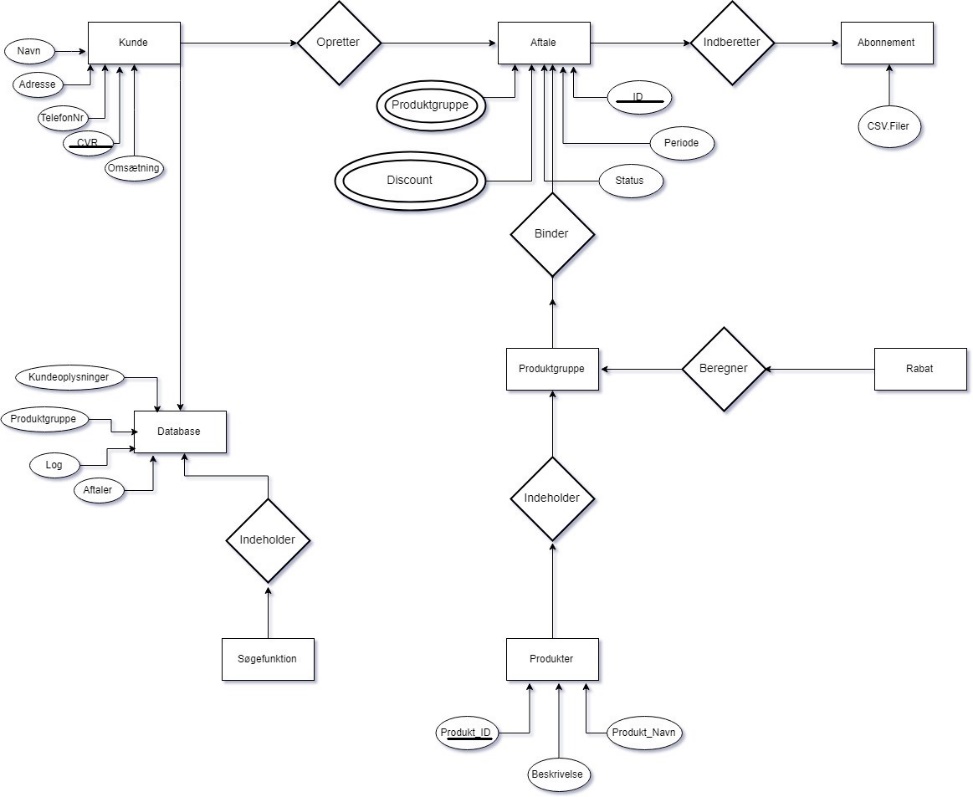
### Entity Relationship Diagram

På baggrund af vores Class diagram har vi udarbejdet dette nedstående E-R-D som viser den visuelle struktur over AP Engros database samt de relationer der er tilknyttet mellem entiteterne.

Vores E-R-D er bestående af 8 entiteter med 3 af dem som indeholder en *primary key*. Disse entiteter defineres som stærke, eftersom de indeholder en attribut der tildeler en unik værdi. Ser vi nærmere på diagrammet kan vi se 3 entiteter der indeholder primary keys.

Entiteten *Kunde* består af 5 attributter, hvor attributten CVR er understreget og defineret som *primay key*. Det vil sige at hver gang en kunde oprettes hos AP Engros, vil blive oprettet med deres eget unikke CVR-nummer.

Ser vi nærmere har vi den næste stærke entitets i *Aftale*. Her er ID defineret som *primary key*, eftersom der bliver tildelt et unik ID-nummer, når der oprettes en aftale mellem *Kunde* og AP Engros. Den sidste stærke entitet *Produkter* har vi defineret *Produkt\_ID* som *primary key*. Hver gang virksomheden AP Engros oprette nye produkter, tildeles et unikt autogeneret Produkt\_ID som sikre at der ikke forekommer problemer i databasen.

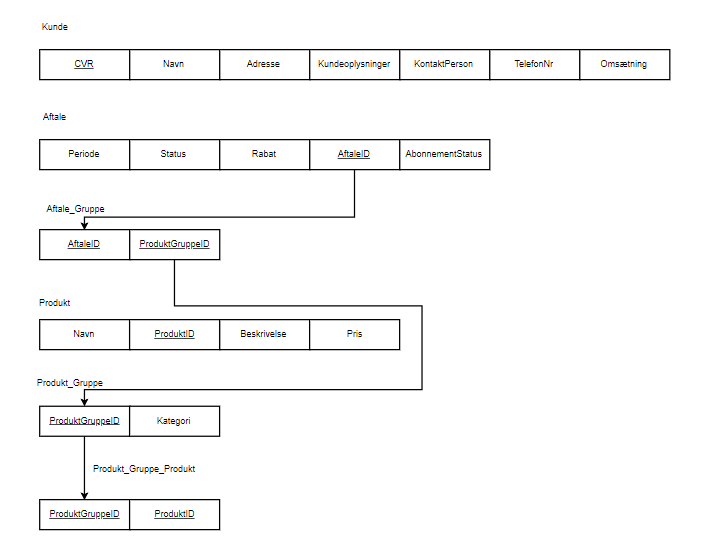


### Mapping Diagram

Vi har anvendt systemudvikling til at udarbejde dette Mapping Diagram. Det giver en visuel forståelse for, hvordan vores database er opbygget af de forskellige entiteter og attributter. Vi kan derfor ud fra dette diagram nemmere danne overblik over, hvilke entiteter der optræder i databasen og hvilke attributter der tilknyttet. Den viser også hvilke attributter optræder som *Primary-og Foreign Keys.*

Under entiteten *Aftale* er attributten *AftaleID* defineret som Primary key. Den optræder som *Foreign Key* i *Aftale\_Gruppe*.

Dernæst er optræder attributten *ProduktGruppeID* i entiteten *Produkt\_Gruppe* og *Produkt\_Gruppe\_Produkt.*



## Konklusion

Opnår de en større kundegruppe/markedsandel

Øger AP Engros omsætning, ved implementering af IT-løsning?

Fordele ulemper ved IT løsning kontra tidligere måde

## Logbog

Der er ført ugentlig logbog for gruppens arbejdsplan.

*Uge 1:*

Den første uge var der primært fokus på systemudviklingen.

I første omgang var det med henblik på strukturen af hele vores program: Opbygning, baseklasser, om det skulle være MVVM osv.

Desuden lavede vi en liste over de forskellige diagrammer som skulle laves og begyndte stille og roligt at få lavet dem.

Vi lagde ud med at lave vores domæne model som skulle give et indblik i hvordan vi havde tænkt os at bygge programmet op overfor kunden.

Vi begyndte på vores brief use cases og fully dressed use cases.

Vi begyndte derefter at kigge på vores sekvensdiagrammer og klasse diagram.

*Uge 2:*

I 2. uges forløb fik vi lavet det meste af vores systemudvikling færdigt - vi fik fuldendt vores sekvensdiagrammer, klasse diagram, ERD, Mapping samt unified processes.  
  
Herefter kunne vi begynde at stille vores solution op i visual studio.

(Det viste sig dog senere at vore sekvensdiagrammer blev ubrugelige og slet ikke havde noget som helst relation til måden vores program rent faktisk blev lavet)

*Uge 3:* I begyndelsen af 3. uge satte vi os alle sammen ned sammen og satte Model og ViewModel op i forhold til MVVM.  
Altså opbygning af de forskellige klasser indenfor model og viewmodel.  
Vi brugte lang tid på at lave full properties og begyndte stille og roligt på implantation af bindings i XAML koden.

Det var også på dette tidspunkt at vi besluttede os for at bruge ICommands i stedet for button\_clicks, da det arbejder langt bedre sammen med MVVM med henblik på databinding.  
  
Vi valgte herefter at benytte os af Headers eller tabs om man vil i MainWindow - Altså i stedet for at bruge et nyt vindue til hver ViewModel (Customer, Contract, Product, Statistic og Log) Så er det kun et vindue, hvor vi har indført forskellige tabs der kan skiftes imellem for at give en bedre overskuelighed for kunden.

Vi satte herefter skabelonerne op for hvordan de forskellige tabs skulle se ud, hvad de skulle indeholde i henhold til vores 1. uges design mønstre.

Og begyndte at få lavet de forskellige ICommands samt Lister og observablecollections som skulle henholdsvis udføre commands ved vores knapper eller vise informationer i vores lister/grids.

Vi fik også her oprettet database samt database klasse og fik forbundet de 2 med vores forskellige viewmodels og tabs, således det var muligt at kalde en metode til at tilføje kunder/aftaler/produkter til vores database.

*Uge 4:*

Vores 4. uge blev primært brugt på rettelser i diverse klasser eftersom vi fandt ud af tingene ikke nødvendigvis fungerede som vi gerne ville have dem til, eller de manglede nogle essentielle elementer.

*Uge 5:*

Efter denne uge havde vi fået styr på Customer og Product klasserne.  
Altså opretning, sletning samt redigering af kunder og produkter.  
Samt taget højde for evt. duplicates m.m.

Vi havde desuden fået XAML til at se ud som vi gerne ville og manglede nu kun at få de sidste ting indenfor Contract, Statistic og Log til at fungere - Men det skulle hurtigt vise sig at blive sværere end vi troede.

*Uge 6*

I starten af uge 6, blev Log færdig, således at den logfører alle inddateringer samt fejl i databasen med tidspunkt og meddelelse således det er overskueligt at kigge i loggen.

Vi manglede så kun Contract og statistics som vi på et hængende hår fik fuldendt i den sidste periode af uge 6.  
  
Det var så også desværre her det gik op for os at vi stadig ikke havde fået adgang til linux serveren og derfor intet havde mht. csv. Filerne og altså fungerende subscription så kunden ville modtage .csv filer med sine abonnerede produktgrupper.

# Bibliografi

<https://stackoverflow.com/>

<https://google.com/>

<https://www.w3schools.com/sql/default.asp>

<https://www.c-sharpcorner.com/>

<https://docs.microsoft.com/da-dk/dotnet/csharp/>

## Bilag

|  |  |
| --- | --- |
| **UC name** | **Opret kunde profil** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig der ønsker at etablere en profil og gemt i systemet. Læser, der ønsker et overblik over profil. |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Profilen er korrekt oprettet i systemet, gemt i database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Opret Profil” og profils menuen for opretning åbnes. 2. Brugeransvarlig indtaster :  * “User Navn”, ”Password”; * ”Navn”, ”CVR”, ”Adresse”, ”Kontaktperson”, ”Konto oplysninger”, ”Forventet årlige omsætning”;  1. Brugeransvarlig trykker GEM. 2. Systemet gemmer profilen i databasen, returnerer med en godkendelse og lukker profilens menuen. |
| **Extension** | 2a. Brugeransvarlig indtaster ingen data or data af forkert type: ”Navn” ,”Password”, ”Navn”, ”CVR”, ”Adresse”, ”Kontaktperson”, ”Konto oplysninger”, ”Forventet årlige omsætning”;  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier.  4a. Systemet kan ikke gemme profilen i databasen pga. fx ingen internetforbindelse, hukommelsesmangel eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker profilen menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

|  |  |
| --- | --- |
| **UC name** | **Redigere kunde profil** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig vælger redigere en kunde profil |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Profilen er korrekt redigeret i systemet, gemt i database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Vis” 2. Liste af kunde hentes fra databasen og printes ud på skærmen. 3. Brugeransvarlig vælger en kunde fra list 4. Menuen for redigering af profilen fuldes op 5. Brugeransvarlig vælger enhed som ønskes redigeret. 6. Brugeransvarlig redigerer de ønskede informationer og trykker ”gem”. 7. Systemet redigerer profilen fra databasen, returnerer med en godkendelse og lukker enhedsmenuen. |
| **Extension** | 2a. Systemet kan ikke hente alle enheder fra databasen pga. fx ingen internetforbindelse eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker profilen menuen.  3a. Brugeransvarlig vælger ingen enhed.  - Systemet returnerer en fejlbesked og beder brugeransvarlig om at vælge en enhed.  4a. Systemet kan ikke fuldes op informationer fra listen til profil menuen  - Systemet returnerer en fejlbesked og lukker enhedsmenuen.  6a. Brugeransvarlig indtaster data af forkert type, eller efterlader informationer ”tomme”.  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at indtaste korrekte værdier.  7a. Systemet kan ikke redigere profilen, da den i øjeblikket er tildelt af job eller er i køen.  - Systemet returnerer en fejlbesked og lukker enhedsmenuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

|  |  |
| --- | --- |
| **UC name** | **Slet kunde profil** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig vælger at slettet en kunde profil fra systemet. |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Brugeransvarlig er korrekt slettet fra systemet. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Vis”. 2. Brugeransvarlig vælge en kunde profil 3. Systemet slettet profilen fra databasen, returnerer med en godkendelse og lukker profilen menuen. |
| **Extension** | 2a. systemet kan ikke få forbindelse til databasen.  - System returnerer en fejlbesked.  2b. brugeransvarlig vælger ingen kunde profil  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt data/filformat, SQL Server |
| **Frequency of occurrence** | Middel |
| **Miscellaneous** |  |

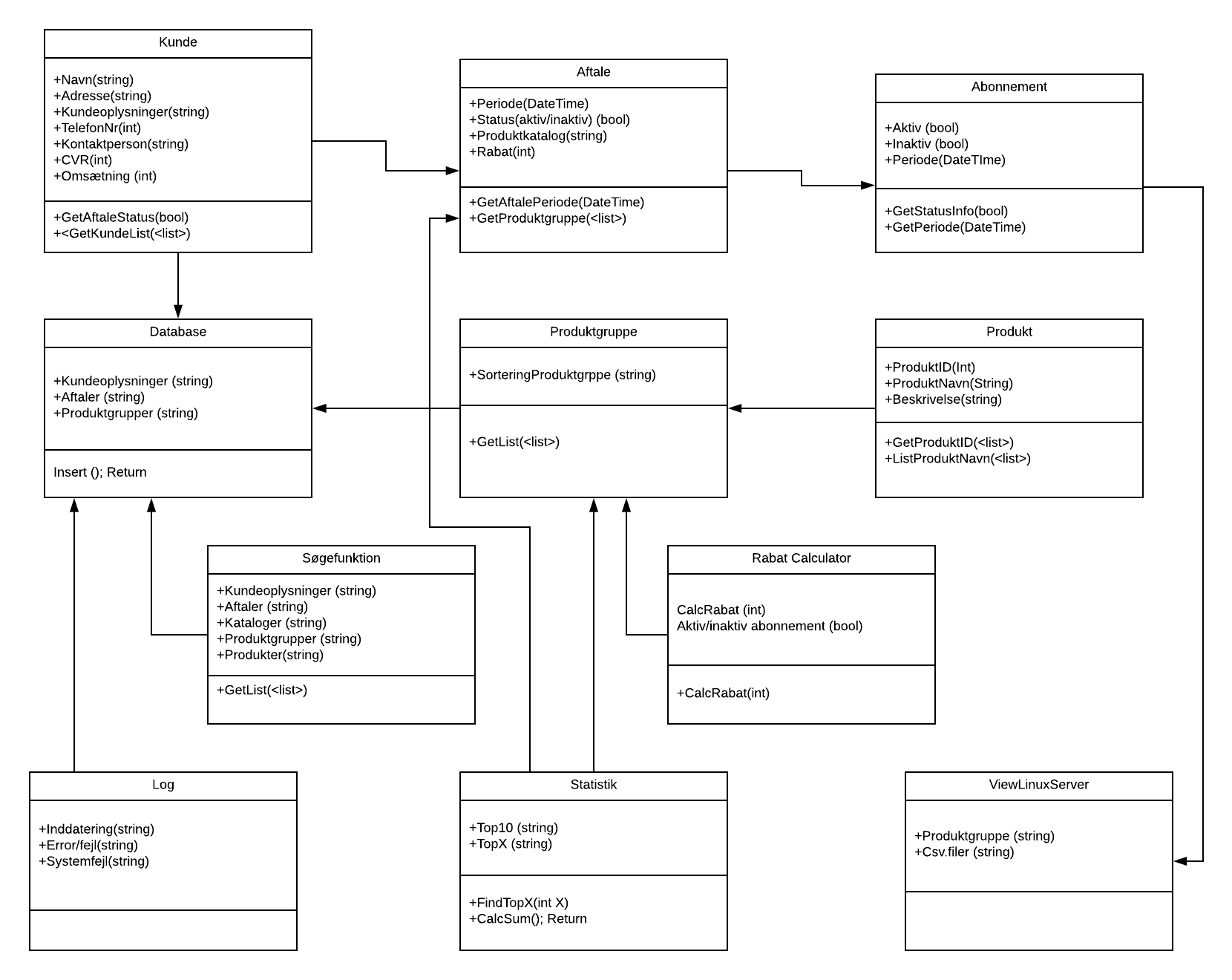
|  |  |
| --- | --- |
| **UC name** | **Vis kunde profil LISTE** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig der ønsker at vise en list med alle kunde profilen. |
| **Pre-conditions** | Brugeransvarlig ansvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Listen er korrekt hændet fra database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Vis ” 2. Liste af alle bruger profilen hentes fra databasen og printes ud på skærmen. |
| **Extension** | 2a. Systemet kan ikke hente alle profilerne fra databasen pga. fx ingen internetforbindelse eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

|  |  |
| --- | --- |
| **UC name** | **Opret aftale** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig der ønsker at etablere en aftale og gemt i systemet. |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Aftalen er korrekt oprettet i systemet, gemt i database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ” Aftale” og ny vinduer for opretning åbnes. 2. Brugeransvarlig indtaster :  * ”Tidsbestemt” ,  1. Brugeransvarlig vælge :  * ”Produktgruppe” og ”Abonnement” ;  1. Brugeransvarlig trykker OPRET . 2. Systemet gemmer profilen i databasen, returnerer med en godkendelse og lukker aftalens menuen. |
| **Extension** | 2a. Brugeransvarlig indtaster data af forkert type, eller efterlader informationer ”tomme”.  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier.  3.a Brugeransvarlig vælger ingen ”Produktgruppe”  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier.  5a. Systemet kan ikke gemme aftalen i databasen pga. fx ingen internetforbindelse, hukommelsesmangel eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker aftalens menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

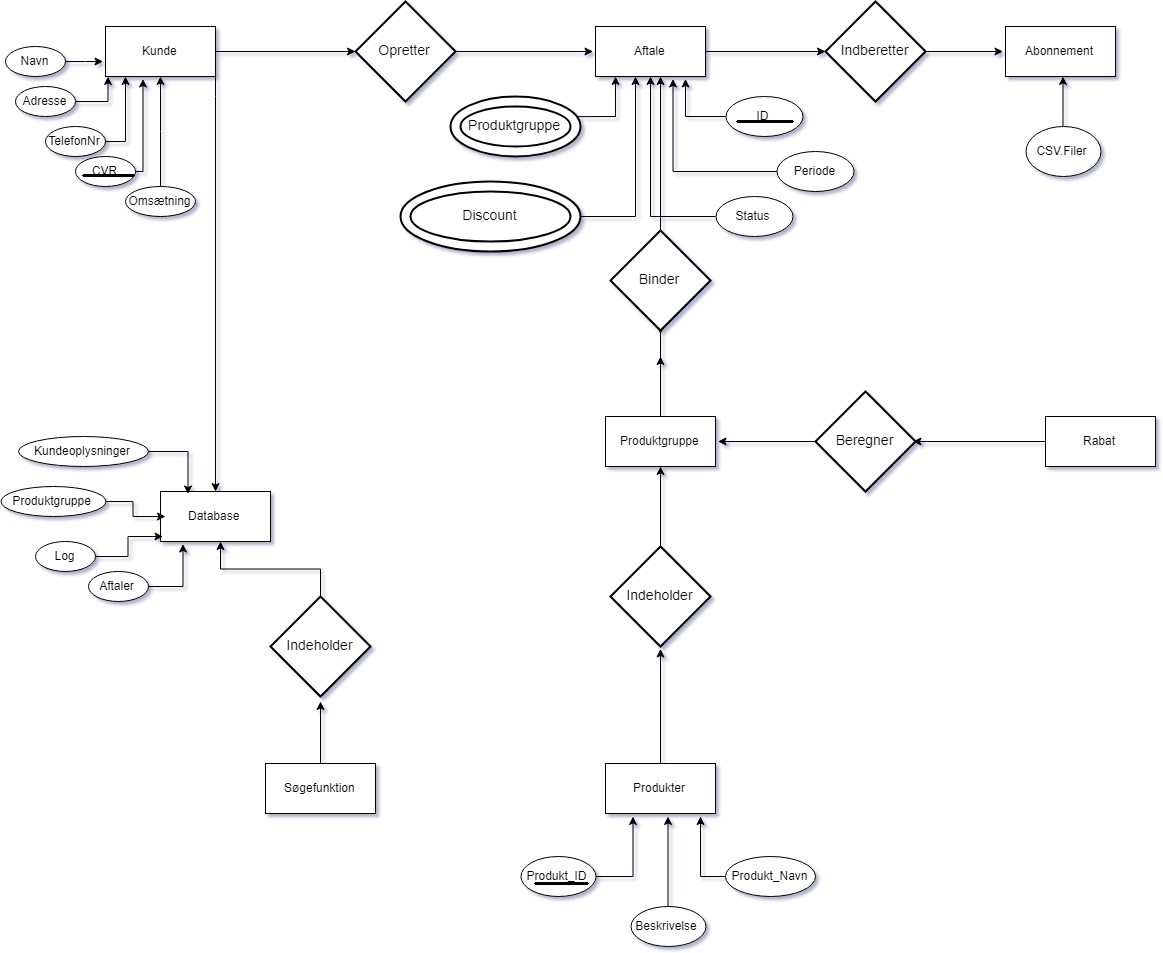
|  |  |
| --- | --- |
| **UC name** | **Redigere aftale** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig der ønsker at redigere en aftale og gemt i systemet. |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Aftalen er korrekt redigeret i systemet, gemt i database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Vis” 2. En liste af aftale hentes fra databasen og printes ud på skærmen. 3. Brugeransvarlig vælger en aftale fra list 4. Menuen for redigering af profilen fuldes op 5. Brugeransvarlig vælger enhed som ønskes redigeret. 6. Brugeransvarlig redigerer de ønskede informationer og trykker ”gem”.   Systemet redigerer profilen fra databasen, returnerer med en godkendelse og lukker enhedsmenuen. |
| **Extension** | 2a. systemet kan ikke få forbindelse til databasen  - System returnerer en fejlbesked.  2b.Brugeransvarlig indtaster ingen ”Tidsbestemt” eller  vælge ingen: ”Produktgruppe” og ”Abonnement” ;  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at indtast eller vælger nye værdier.  4a. Systemet kan ikke gemme aftalen i databasen pga. fx ingen internetforbindelse, hukommelsesmangel eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker aftalens menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

|  |  |
| --- | --- |
| **UC name** | **Slette aftale** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig ønsker at slette en aftale. |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Aftalen er korrekt slettet fra systemet. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ”Vis” 2. En liste af aftale hentes fra databasen og printes ud på skærmen. 3. Brugeransvarlig vælger en aftale fra list 4. Brugeransvarlig vælger aftale som ønskes slette. 5. Brugeransvarlig slette de ønskede informationer og trykker ”Slette”.   Systemet slette profilen fra databasen, returnerer med en godkendelse og lukker enhedsmenuen. |
| **Extension** | 2a. systemet kan ikke få forbindelse til databasen  - System returnerer en fejlbesked.  2b.Brugeransvarlig vælger ingen aftale;  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier.  4a. Systemet kan ikke slette aftalen fra databasen pga. fx ingen internetforbindelse, hukommelsesmangel eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker aftalens menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

|  |  |
| --- | --- |
| **UC name** | **Vis Varekatalog** |
| **Scope** | Manager |
| **Level** | Bruger |
| **Primary actor** | Brugeransvarlig |
| **Stakeholders & interests** | Brugeransvarlig ønsker at se varekatalog |
| **Pre-conditions** | Brugeransvarlig er autentificeret og autoriseret til brug af systemet. |
| **Success guarantee** | Produktkatalog er korrekt hændet fra database. |
| **Main success scenario** | ”Happy” scenariet :   1. Brugeransvarlig trykker på ” Produktkatalog” 2. En ny vinduer åben 3. Brugeransvarlig kan søg efter en Produkt Grupper 4. En liste med Produkt gruppe hentes fra databasen og printes ud på skærmen. |
| **Extension** | 3a. systemet kan ikke få forbindelse til databasen  - System returnerer en fejlbesked.  2b.Brugeransvarlig vælger ingen Produkt Grupper;  - Systemet returnerer med en fejlbesked og beder brugeransvarlig om at vælger nye værdier.  4a. Systemet kan ikke hente alle profilerne fra databasen pga. fx ingen internetforbindelse, hukommelsesmangel eller hvis databasen er nede.  - Systemet returnerer en fejlbesked og lukker aftalens menuen. |
| **Special requirements** | Informationssikkerhed, Brugervenlighed. |
| **Technology and data variation list** | Universelt dataformat, SQL Server |
| **Frequency of occurrence** | Middel |

Class Diagram

E-R-D Diagram



Domain Model

### 