

IT-løsning til Tolkeservice

Omar Khalidan - [xx-xx-xx]

Morten Trolle Nielsen - [14-06-79]

Instruktor: Andreas Frisch

Projekt i systemudvikling 2014 (ProjDat2014)

16. april 2014

Indhold

1	Abstract	3
2	Problemformulering	4
3	Anvendelsesområde	5
4	Problemområde	7
5	Kravspecifikation	9
5.1	Funktionelle krav	9
5.2	Non-funktionelle krav	9
6	Softwarearkitektur	9
7	Use-cases	9
8	Projektplan	9
9	No silver bullit	9
10	den anden artikel	9

1 Abstract

Dette er anden delrapport i faget Systemudvikling. Vores udgangspunkt er første delrapport, som vi her udvikler og udbygger. Vi vil foresætte arbejdet med og præsentationen af den IT-løsning, som vi har udviklet til kunden; tolkeservicen K. Translation. Specielt vil vi fokusere på modelleringen af systemet. Vi vil præsentere anvendelsesområdet og problemområdet; både med tekst og UML-diagrammer. Det skal resultere i en bedre og umiddelbar forståelse af vores påtænkte it-løsning. Derefter vil vi også med afsæt i første delrapports kravspecificering videreudvikle denne og se på de funktionelle og nonfunktionelle krav. Vi vil tage modelleringen et skridt videre ved hjælp af use-cases, så vi kan understøtte vores modellering med konkrete eksempler på brug af systemet.

Vi vil afslutte denne anden delrapport med to korte referater og efterfølgende analyse og perspektivering af de to artikeler, der er et krav til rapporten.

Vi er selv førsteårs datalogistuderende ved Københavns Universitet, og det forudsættes, at læseren befinder sig på samme læringsniveau eller højere, idet der undervejs vil forekomme adskillige fagspecifikke termer.

2 Problemformulering

”Hvordan konstruerer man et bookingsystem til en tolkevirksomhed, som formindsker tidsspilde og fejl ? Og hvordan kan dette gøres på en brugervenlig måde, så virksomhedens samarbejdspartnere enkelt og simpelt kan tilgå systemet uden for meget tidsspilde ?”

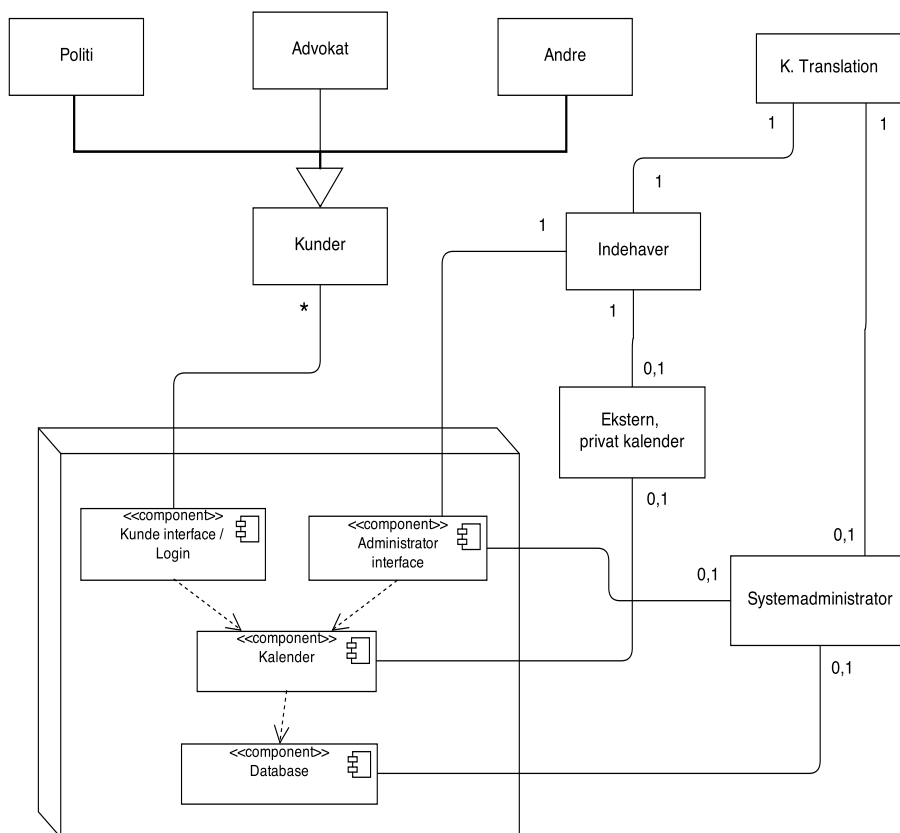
Tolkeservicen K. Translation er et lille, enmands tolkebureau med speciale i arabiske sprog. Indehaveren af bureauet er , og hun er som sagt både leder og eneste medarbejder i foretagendet. Hendes kunder er offentlige myndigheder som politiet og anklagemyndigheden og private aktører som advokatbureauer og forskellige virksomheder. Disse instanser kontakter K. Translation, når de har behov for en tolk til konkrete sager eller ved andet forefaldende tolkearbejde, og der laves en aftale. Det kan både være aftaler to eller tre uger ude i fremtiden, eller det kan være aftaler med kortere frist ved mere presserende sager.

K. Translation (KT) har i længere tid ønsket et it-system, som kan afhjælpe lidt af den daglige travle arbejdsbyrde. Derfor tog ejeren af KT kontakt til os, og anmodede om et møde omkring dette problem. Det første møde fandt sted fredag den 21. marts, og vi blev enige om at begynde et samarbejde, der skal resultere i et færdigt kalender- og bookingssystem til KT og KT's kunder. Kunden præsenterede os for en liste af de funktioner, som hun ønsker, at det nye system skal kunne levere. For det første ønskes der en hjemmeside, hvor KT's kunder kan logge på systemet med deres EAN-nummer. Da vi bl.a. snakker om offentlige myndigheder som politi og anklagemyndighed, skal login systemet fremstå professionelt og sikkert. Der skal være en tilhørende database indeholdende alle KT's kunder, og det skal være muligt løbende at tilføje kunder til databasen.

Når KT's kunde har logget på systemet, skal vedkommende præsenteres for en kalender opslået på dags dato med mulighed for visning af de næste to til tre uger. Det er KT's holdning og erfaring, at der ikke forekommer aftaler længere end tre uger ude i fremtiden; derfor denne begrænsning. KT's kunde skal i kalenderen kunne se alle de tidspunkter, hvorpå det er muligt at booke en aftale med K. Translation. Disse ledige tidspunkter kan evt. markeres med grønt. Allerede indgåede aftaler markeres med rødt for at tydeliggøre denne forskel. Når KT's kunde har besluttet sig for en dato og et tidspunkt, taster vedkommende dette ind i systemet, og aftalen gemmes i databasen og vises herefter i kalenderen. Det er vigtig for KT's ejer, at hendes samarbejdspartnere ikke skal bruge alt for lang tid på at booke en aftale, så enkelhed og intuitivitet skal prioriteres højt.

3 Anvendelsesområde

Vi vil i de næste afsnit kigge på to centrale dele af enhver modellering; nemlig anvendelses- og problemområdet. Vi begynder her med en analyse af anvendelsesområdet, og for overskuelighedens skyld præsenterer vi det først ved hjælp af et UML-diagram. Se figur 1.



Figur 1: UML-diagram over anvendelsesområdet

Det skal dog præciseres, at det store kvadrat i nederste venstre hjørne i figur 1 ikke hører til anvendelsesområdet. Det er et high-level billede af problemområdet, som her er medtaget for at binde de to områder sammen, og som behandles indgående i næste afsnit.

Umiddelbart er anvendelsesområdet meget begrænset. Vi har fundet følgende klasser, der skal modelleres i anvendelsesområdet:

- K. Translation med indehaver (som samtidig er eneste medarbejder)
- Systemadministrator hos K. Translation
- K. Translations kunder som politi, advokater, etc.
- Eksternt, privat kalendersystem

K. Translations indehaver, som samtidig er den eneste medarbejder, skal som den første modelleres i systemet. Hun skal bruge kalenderen og bookingssystemet i sit daglige arbejde, og hendes ønske er, at kunne kombinere hendes private kalender med vores it-løsning, så hun har alle sine aftaler samlet et sted. Derfor har vi modelleret hende som indehaver af K. Translation og bruger af kalenderen, men vi har samtidig udvidet modellen med en ekstern kalenderklasse mellem indehaveren og systemet, hvor hendes private aftaler ligger, og hvor den eksterne kalender kan synkronisere med vores bookings- samt kalendersystem og overføre de private aftaler. Det skal dog siges, at vi på nuværende tidspunkt er meget usikre på denne funktion, og at vi potentielt må fortælle K. Translation, at dette ligger uden for vores formåen, så vi er påpasselige med at stille hende for meget i udsigt. Derfor har den også i UML-diagrammet fået multiplicitet 0, 1, da vi højst regner med 1 ekstern kalender, der skal integreres. Det kunne godt gå hen og blive en stor og kompliceret opgave, så derfor får den også i første omgang en lavere prioritet end andre mere tilgængelige ønsker. Se evt. afsnittet om projektplan.

Vi har i UML diagrammet givet indehaveren af KT multiplicitet 1. Man kunne sagtens forestille sig flere ejere af bureauet, men vi har taget udgangspunkt i den nuværende situation, og der er ikke umiddelbart udsigt til nogen ændringer. Selvom indehaveren ikke besidder it-kundsskaber ud over det almindelige, har vi i erkendelse af, at hun også er eneste medarbejder, i UML-diagrammet givet hende adgang til administrationsinterfacet, da hun sikkert vil komme i en situation, hvor hun selv er nødt til at tilgå systemet med alle rettigheder for at ændre eller oprette nye kunder.

Hvis vi skal foresætte modelleringen af anvendelsesområdet, så er der en Systemadministratorklasse associeret til K. Translation. Indehaveren af KT er ganske almindelig it-bruger, og her snakker vi om mail, netbank, facebook, etc, men er ikke it-kyndig udover dette. Derfor er vi nødt til at modellere en systemadministrator, der har overblik over systemet, og som kan yde support, hvis der opstår problemer. Hvis det færdige system ikke kommer til at ligge på en intern server hos KT, men ender med at blive hosted af en ekstern udbyder af serverplads, så vil nødvendigheden af denne klasse mindskes betragteligt, men der vil altid være et

vist behov for central systemadministration hos KT i tilfælde af ændringer i kontaktinformation eller kalenderbrug med dertil hørende ændringer i kodebasen og efterfølgende nye upload til serveren. K. Translation må selv tage stilling til, hvor meget systemadministration der er nødvendig efter afleveringen og implementeringen af systemet, men vi vil naturligvis indgå i en dialog med hende omkring dette emne, når det bliver aktuelt.

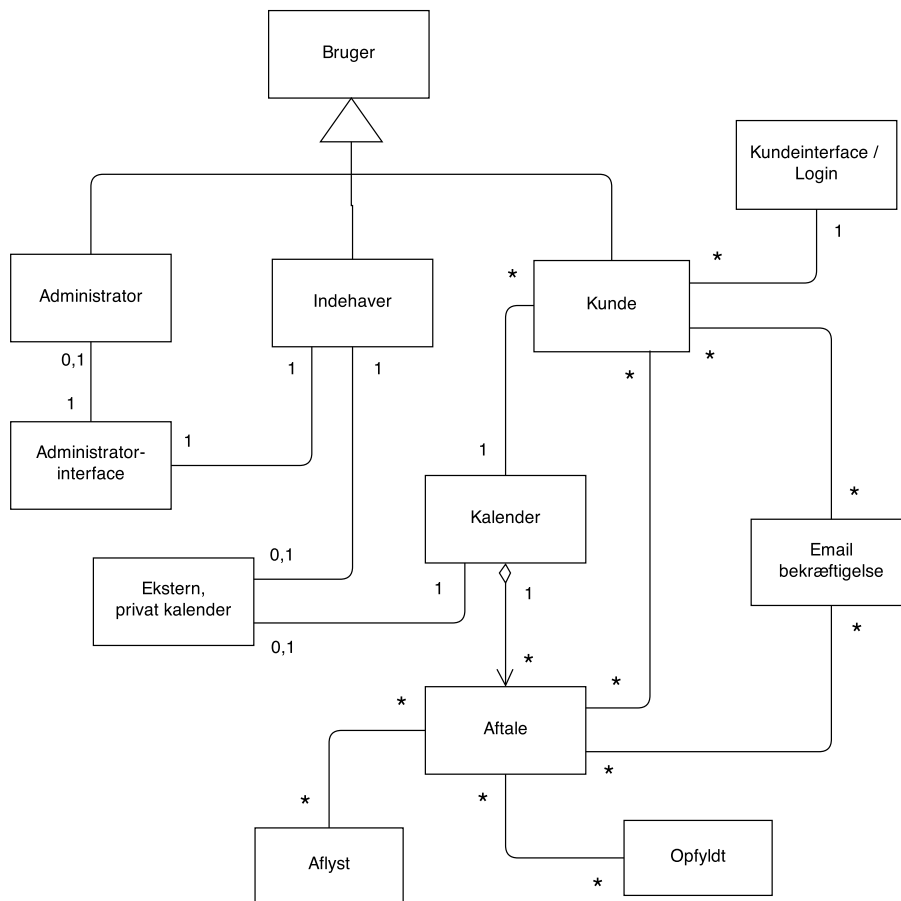
Den sidste klasse, der skal modelleres i anvendelsesområdet, er KT's kunder og samarbejdspartnere. Denne klasse er omdrejningspunktet for hele systemet, da en af hovedpræmisserne for vores it-løsning er, at KT's kunder på en hurtig og overskuelig måde kan booke en aftale med tolkeservicen. Kundegrundlaget er offentlige myndigheder som politi, ministerier og hospitaler samt private aktører som advokater og andre samarbejdspartnere. Når de møder hjemmesiden, skal de kunne tilgå kalenderen via et loginsystem med deres EAN-nummer og derfra føres videre til selve kalenderen, hvor der kan bookes en aftale. Vi må forvente, at kundesegmentet har meget svingende it-kundsskaber, men at de i kraft af deres daglige arbejde er vant til at arbejde med diverse login- og kalendersystemer, der findes overalt i f.eks. den offentlige sektor.

4 Problemområde

Efter at vi har modelleret tilgangen til it-systemet gennem anvendelsesområdet, er turen nu kommet til systemets hjerte: problemområdet. Det er her, vi efter modelleringen skal implementere de løsninger, vi finder på KT's ønsker. For at binde modelleringen sammen går alle klasserne fra anvendelsesområdet igen i problemområdet. Vi præsenterer igen for overskuelighedens skyld først et UML-diagram over området. Se figur 2. Dernæst kommer der en kort opsummerende liste med klasserne efterfulgt af et længere forklarende afsnit.

Her følger en liste af de klasser vi har identificeret i problemområdet:

- Bruger: Kunde, administrator, indehaver. Går igen fra anvendelsesområdet
- To interfaces. Et til kunderne og et til administratoren
- Kalenderen, som kunderne kan bruge, når de skal finde en ledig tid, og som KT's indehaver kan bruge i sit daglige, travle arbejde
- Aftale. Kunderne booker en aftale i kalenderen.



Figur 2: UML-diagram over problemområdet

- En aftale kan både være aflyst og opfyldt, hvis den ikke længere er aktuel
- Når kunden har booket en aftale, sendes der en bekræftende email tilbage til kunden med det aftalte tidspunkt
- KT's indehaver har et ønske om at kunne synkronisere hendes eksterne, private kalender med vores it-løsning

5 Kravspecifikation

5.1 Funktionelle krav

5.2 Non-funktionelle krav

6 Softwarearkitektur

7 Use-cases

8 Projektplan

9 No silver bullit

10 den anden artikel