

Rustvogns kalender



Indhold

Indledning	1
Forretningen	1
Forretningsmæssig baggrund	1
Forretningsmæssig problemstilling	1
Succeskriterier	2
Forretningsmodel lærred	2
Business case	2
Interessenter	3
Systemets brugere	3
Scenarie oversigt	3
1-scenariet	4
Formål	4
Forretningsmæssig løsningsbeskrivelse	4
IT-mæssig løsningsbeskrivelse	4
Key Performance Indicators (KPI)	5
Konklusion – Business case	6
Business artefakter	6
Forretningens use cases	6
Objekt modellering	6
Domænemodel	7
System sekvens diagram	8
System operations kontrakt	8
Software artefakter	9
Designklassediagram	9
Sekvens diagram	10
Business Process Model and Notation	11
Kode udsnit	12
Processen – Projektarbejde	14
Organisering af gruppearbejde	14
Etablering af kontakt med virksomhed	14

Uforudsete problemer	15
Scrum	15
Brugervejledning	16
Kvalitetssikring	17
Vurderingskriterier	17
Kodestandard	18
Gloser	18
Konklusion	19

Indledning

Bedemanden Peter Klæmmensen er en dansk enkeltmandsejede virksomhed som har deres hovedkontor lokaliseret i Odense SØ, men har også andre afdelinger på Odense C, Bogense og Tommerup. Virksomhedens betydningsfulde rolle er at være de pårørendes rådgiver og vejleder i et vigtigt og hårdt forløb, hvor den sidste afsked ikke kan gøres om. En af Virksomhedens opgaver er at sikre, at den afdødes eventuelle sidste ønsker bliver imødekommet og udført i et omfang, at det ikke strider imod de praktiske og lovmæssige normer. Virksomhedens indkomst kommer fra udlejning af rustvogne, salg af remedier til ceremonier, og gebyrer ved planlægning af højtideligheder.

Virksomheden har på nuværende tidspunkt kun en hjemmeside som de har adgang til der viser en oversigt over hvilke højtideligheder de har og andre relevante informationer der bliver skrevet ned i deres personlige kalender eller på papir, hvilket er meget tidskrævende og ikke tilgængeligt for de resterende ansatte.

Denne rapport beskriver udviklingen af et system til virksomhedsejer og Product Owner(PO), Peter Klæmmensen, af den enkeltmandsejede virksomhed Bedemanden Peter Klæmmensen, hvor vi har udarbejdet et kalendersystem, der skal skabe overblik over virksomhedens aftaler.

Forretningen

Forretningsmæssig baggrund

Bedemanden Peter Klæmmensen sælger samlede pakker til begravelses- og bisættelseshøjtideligheder.

De tager kontakt til de nødvendige personer til højtidelighederne, dette indebærer kontakt til præst, imam eller rabbiner, blomsterforretning og krematorie. Derudover sælger han de remedier der skal bruges til en begravelse.

Forretningsmæssig problemstilling

Firmaet har på nuværende tidspunkt ikke et samlet kalendersystem. De har adgang til en hjemmeside med en oversigt over hvilke højtideligheder de har, resten af informationerne holder de i deres personlige kalender eller på papir.

Vi har fået til opgave at lave et system, hvor virksomheden kan se hvilke højtideligheder, møder og eventuelt andre aftaler de har. Systemet skal selv booke en rustvogn, hvis aftalen kræver en rustvogn.

- Hvordan giver vi virksomheden en overskuelig oversigt over aftaler?

- Hvordan sparer vi tid for virksomheden?

Succeskriterier

- Programmet skal kunne læse i og skrive til en database.
- Programmet skal selv kunne se om den skal reservere en rustvogn, og gøre dette hvis nødvendigt.
- Programmet skal kræve medarbejderinitialer ved oprettelse af en aftale.
- Programmet skal kunne vise en oversigt over nuværende aftaler.

Forretningsmodel lærred

Partnere Nembedemand.dk	Aktiviteter Køres rustvogn, Arrangere begravelsesceremonier, Arrangere bisættelsesceremonier.	Værditilbud Begravelse- og bisættelseshøjtideligheder.	Kunderelation Dedikeret personlig betjening, Personlig og via telefon.	Kunder Afdødes nære familie
	Ressourcer Rustvogne, Rustvognschauffør, Klargøringspersonale, Knowhow ift. Begravelse- og bisættelseshøjtideligheder.		Kanaler Nembedemand.dk, Fysisk butik, Sociale medier.	
Omkostninger gravsten, tøj, benzin og slid på køretøjer Licens, løn, leje af lokaler, afgifter og skatter.		Indtægter Udleje af rustvogn (m. chauffør), Remedier til ceremoni (gravsten tøj, blomster), Gebyrer ved planlægning af højtideligheder.		

Business case

I dette afsnit vil vi komme ind på hvilket problem Bedemanden Peter Klæmmensen har, som de gerne vil have at vi løser. For at få en god idé om hvorvidt vores foreslåede løsning er optimal, vil vi lave en analyse af problemstillingen. Dertil vil vi sammenligne vores løsningsforslag med nogle alternative forslag, for at kunne vurdere hvilken løsning der er optimal for virksomheden ud fra et forretningsmæssigt synspunkt.

Interessenter

Peter Klæmmensen er ejer af virksomheden. Han agerer både Product Owner og bedemand.

Han har 2 fastansatte bedemænd som har til ansvar at udføre arbejdet forbundet med højtidelighederne.

Systemets brugere

- Product Owner
- Bedemændene

Scenarie oversigt

0-scenariet: nuværende oversigt over højtideligheder og ekstern kalender

Fordele:

Der skal ikke anskaffes dyrebare servere til opbevaring af data

Bedemændene skal ikke til at lære et nyt IT-system at kende.

Ulemper:

Der skal bruges to separate kalendere/oversigter for at kunne se de relevante informationer for en aftale (højtidelighed, møde, evt. andet)

Der skal bruges nogle mandetimer på at koordinere hvem der tager hvilken rustvogn

PO skal via telefon kontrollere hvem der har hvilke aftaler i løbet af dagen.

1-scenariet: samling af informationer i ét kalendersystem, som automatisk booker rustvogne.

Fordele:

Der skal ikke længere bruges flere forskellige systemer til at holde styr på aftaler.

Der vil ikke længere være tvivl om hvem der tager hvilken rustvogn.

Ulemper:

Der skal anskaffes en server til at holde dataene.

Der skal bruges nogle mandetimer på at sætte medarbejderne ind i det nye system.

1-scenariet

Det foreslås at der udarbejdes et samlet elektronisk kalendersystem, som indeholder al data for virksomhedens aftaler, for at mindske det merarbejde der ligger i at have to separate systemer. Derudover skal systemet automatisk gå in og reservere en rustvogn, hvis der er behov for dette.

Formål

Formålene med Kalendersystemet er:

- At skabe overblik over virksomheden aftaler
- At reservere rustvogne automatisk, hvis der er brug for det, så medarbejderne ikke skal bruge tid på at koordinere, hvem der tager hvilken rustvogn.

Forretningsmæssig løsningsbeskrivelse

Løsningen skal sikre at PO ikke skal bruge tid og energi på at finde ud af hvilke aftaler han selv, og de ansatte har, samt sikre at den enkelte ansatte ikke behøver at kigge i flere forskellige systemer for at finde informationerne vedrørende en aftale. Derudover skal den sikre at der ikke sker en dobbeltreservation af en rustvogn.

Den nuværende løsning, hvor PO og de ansatte bliver nødt til at holde en fysisk kalender, digital oversigt, samt manuel fysisk allokering af rustvogne, erstattes af et samlet kalendersystem, som henter den digitale oversigt over højtideligheder, lader brugeren redigere i højtidelighederne og viser en oversigt over højtidelighederne, sammen med øvrige aftaler.

Løsningen skal kunne anvendes af PO og de ansatte bedemænd. Der er intet i vejen for at systemet kan bruges af kontorassistenterne i virksomheden. Løsningen er også bygget sådan at der let kan registreres flere ansatte, til brug af systemet.

IT-mæssig løsningsbeskrivelse

Kalendersystemet udvikles i .NET C#, og skal kunne køre lokalt på arbejdspladsens pc'er.

Systemet designes og opbygges objektorienteret, så det kan udvides med yderligere funktionalitet og let kunne omstilles til at interagere med andre databaser.

Key Performance Indicators (KPI)

KPI	Tidsbesparelse
Hvorfor måle det?	For at se om det nye system reducerer antallet af mandetimer, som skal bruges på planlægning og koordinering.
Hvordan skal det måles?	Det gøres ved at sætte et estimat for hvor meget tid der bruges før implementering og sætte det op mod tiden brugt efter implementering.
Hvem skal være ansvarlig for at måle det?	Peter (Ejer/PO)
Forventet dato for måling	Så snart de ansatte er blevet omskolet til brug af systemet.
Forventet resultat	Vi regner med at brugerne kan spare en betydelig mængde tid, som ellers ville være brugt på fysisk koordinering, og opslag i flere forskellige systemer.
Måling	Der er endnu ingen måling.
Aktionsplan for, hvis resultatet er uden for det forventede mål.	Hvis det viser sig at systemet ikke har den forventede effekt, skal der kigges på om det kan optimeres.
Ansvarlige for actionsplan	PO skal efter en testperiode vurdere om tidsbesparelsen er nok værd økonomisk, til at dække omkostningerne ved anskaffelse og drift af systemet.

Konklusion – Business case

Som det ser ud lige nu, bruger virksomheden en del af deres tid på manuelt at skrive informationer ned i deres fysiske kalendere, derudover bruger de meget tid på at koordinere med og formidle deres arbejdsdag til hinanden, dette gøres enten via telefon eller face to face. Så hvis virksomheden har det økonomiske råderum til at anskaffe sig kalenderen, vil det på længere sigt kunne spare dem for en masse mandetimer.

Business artefakter

Forretningens use cases

I samarbejde med vores PO, har vi fundet frem til tre use cases for vores projekt.

Opret booking

Bedemanden har fået en forespørgsel om en begravelse og vil gerne se om der er nogle ledige rustvogne på det pågældende tidspunkt. Såfremt der er en ledig rustvogn, vil han gerne booke den.

Slet booking

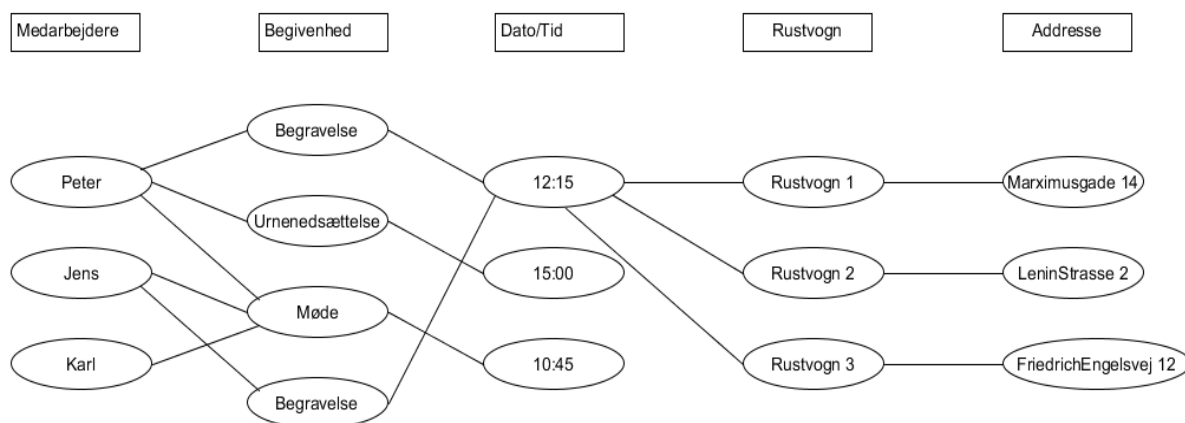
Bedemanden har lavet en fejlbooking og vil gerne slette en bestilling.

Rediger booking

Bedemanden har lavet en fejl i tidspunktet på en bestilling og vil gerne ændre dette.

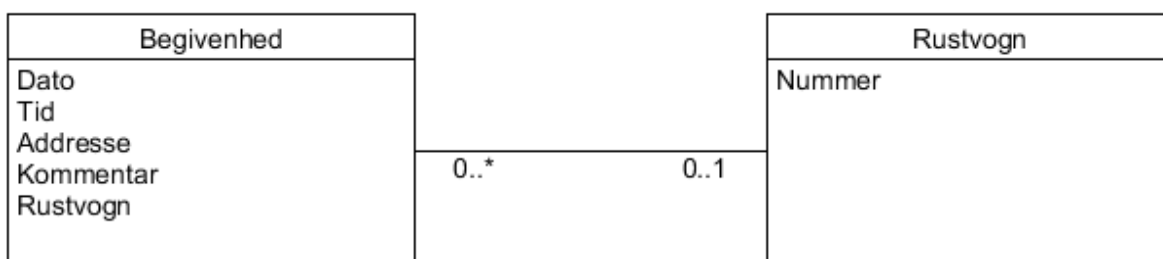
Objekt modellering

Vi har, efter nogle samtaler med PO, om hvad han gerne vil have systemet til at indeholde, lavet denne objektmodel som skal give os en oversigt over hvilke informationer bliver brugt i vores program.



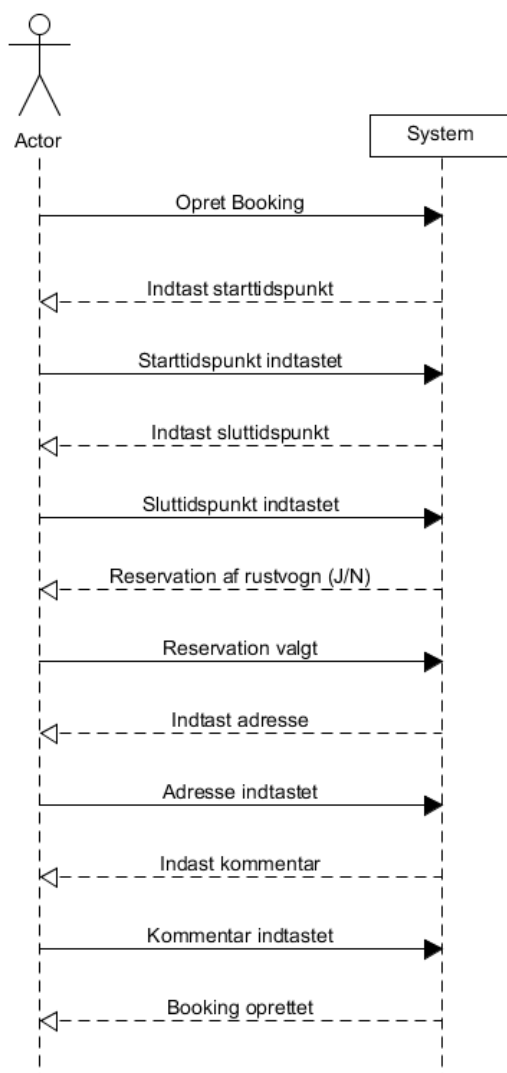
Domænemodel

Domænemodellen bygger på objektmodellen og reelle koncepter som eksisterer i virksomheden. Vi har, i vores løsning, to klasser: Begivenhed og Rustvogn, og vi kan ud fra dette se hvad de to klasser skal indeholde af information. Det kan ses i vores domæne model at mange af objekterne fra objektmodellen overgået til begivenhedsklassen som attributter.



System sekvens diagram

Vi har på baggrund af vores use case CreateEvent lavet et system sekvens diagram, for at illustrere interaktionen mellem brugeren af systemet og systemet.



System operations kontrakt

Operation: OpretBooking

Krydsreference: OpretBooking

Før betingelser: Ingen

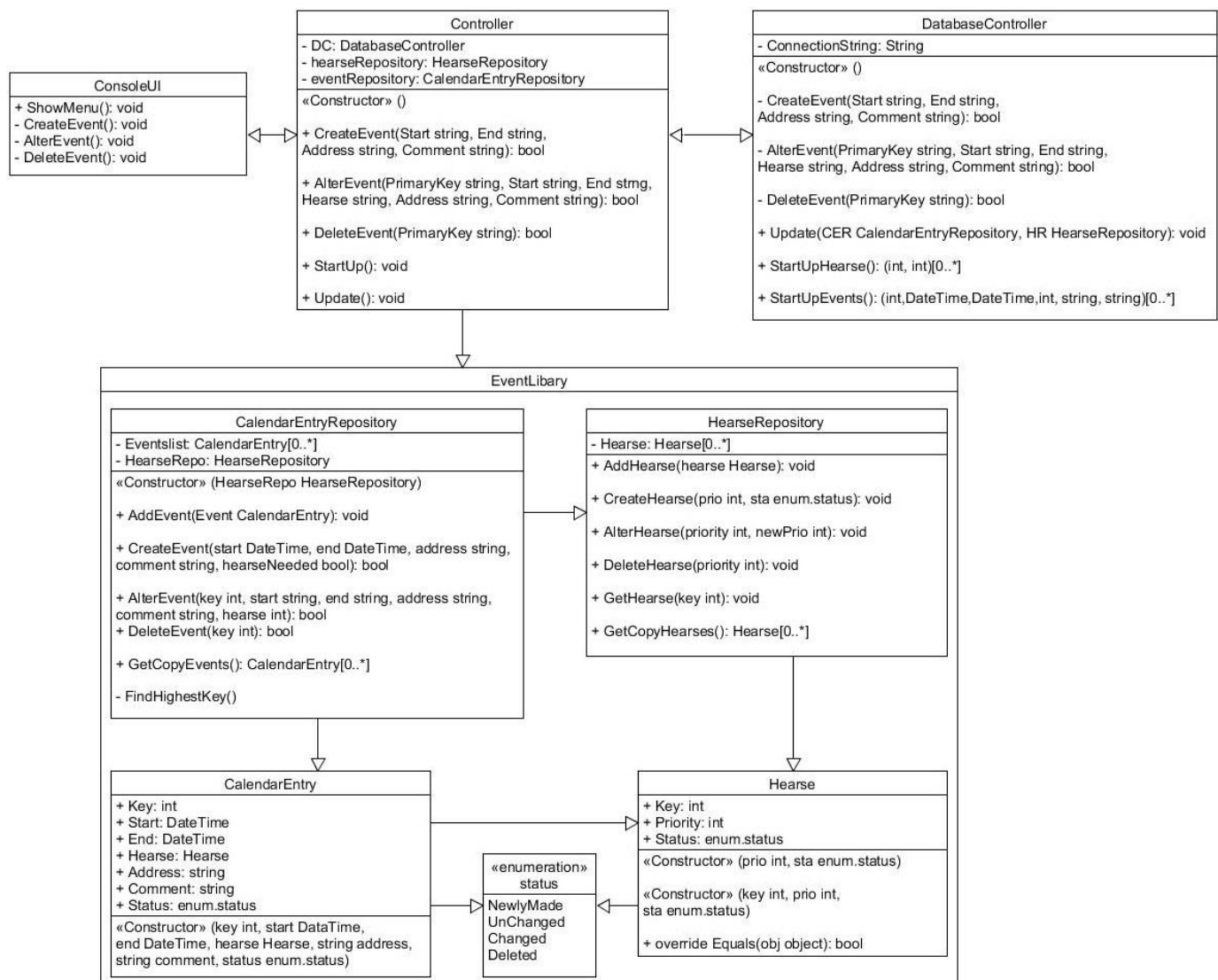
Efter betingelser: En booking er oprettet

Software artefakter

Designklassediagram

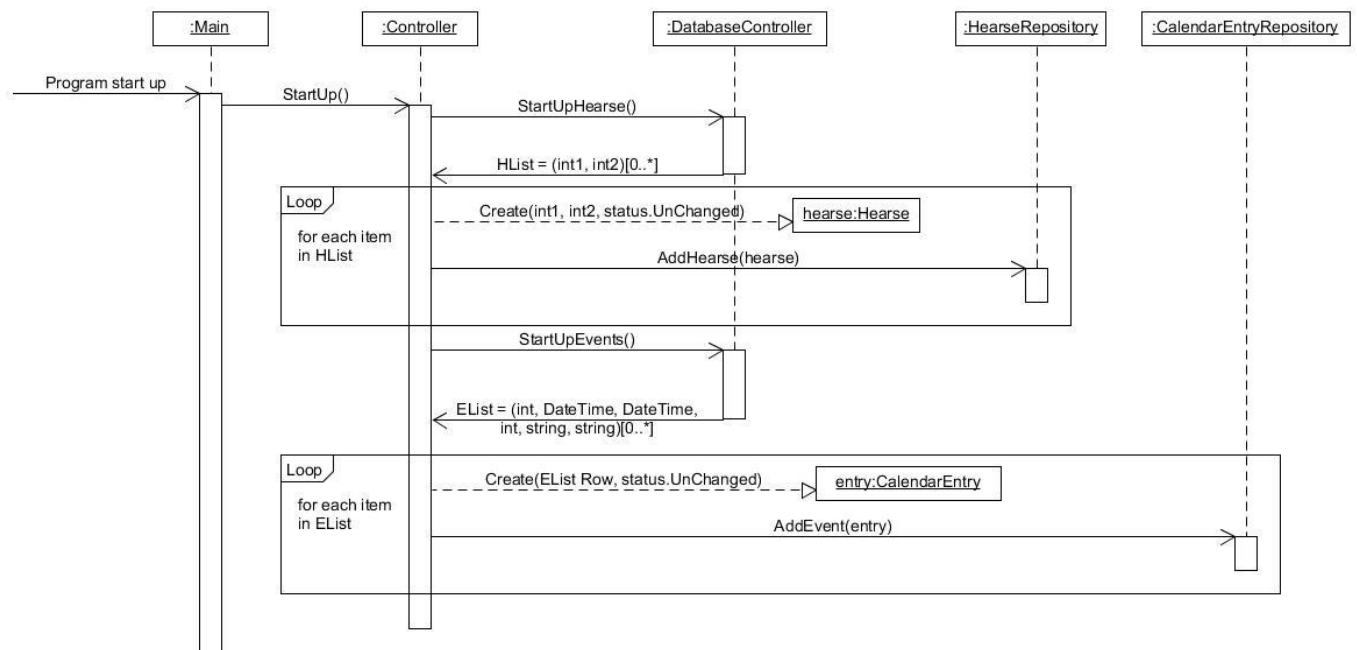
På baggrund af objekt- og domænemodellen har vi opsat et designklassediagram til at danne et overblik over programmets arkitektur samt med hvordan programmet agerer med de forskellige klasser.

Først og fremmest har vi en konsolapplikation, i denne model, navngivet *ConsoleUI*, som skal give kunden mulighed for at vælge hvilken metode, personen vil bruge. Dette sker i vores menu som ud fra hvilket punkt kunden vælger kalder den pågældende metode i vores *Controller* klasse som derefter videresender den information til de respektive klasser i *EventLibrary*

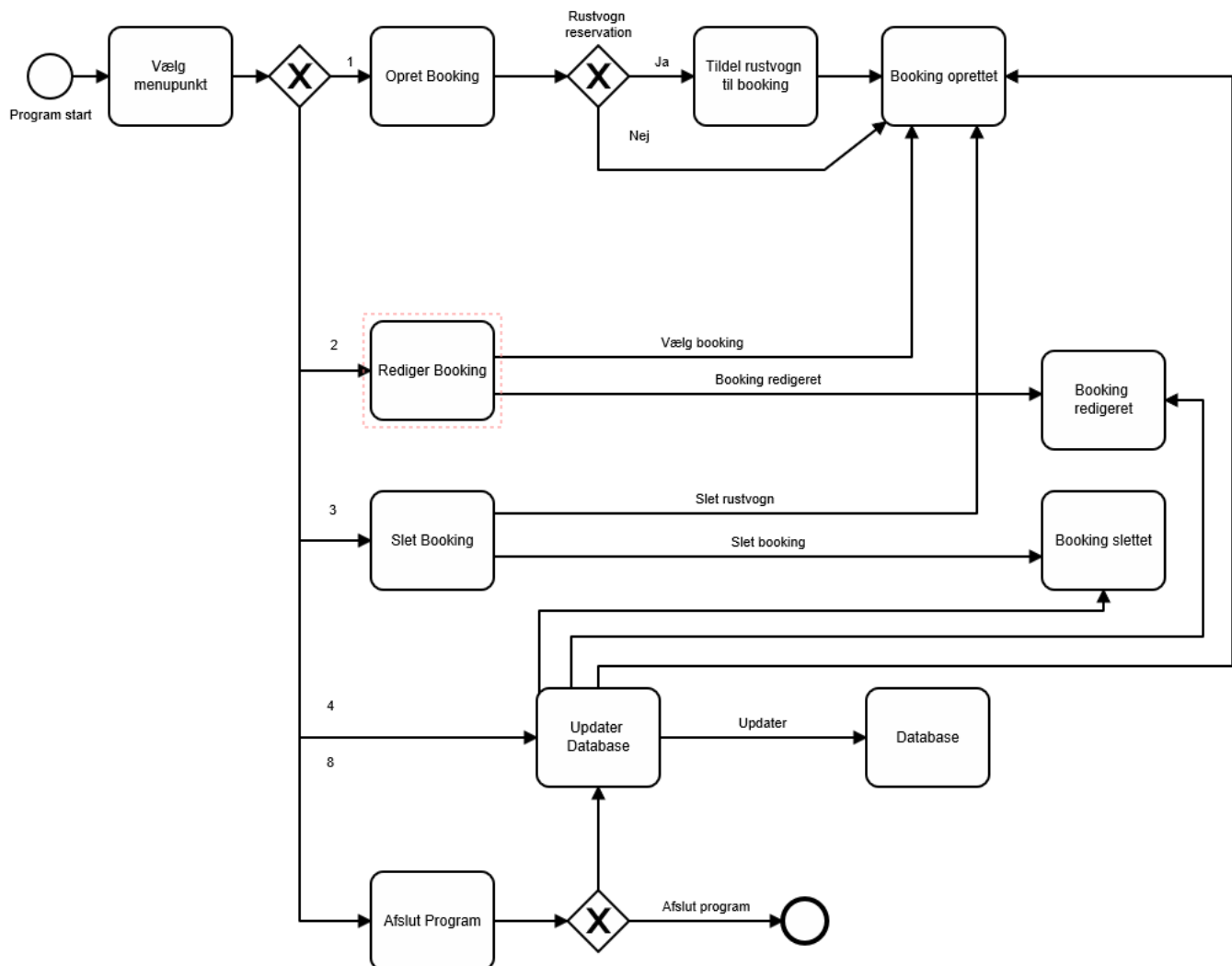


Sekvens diagram

I denne model tager vi udgangspunkt i at kunden vælger punktet *“Create Event”*. Når kunden starter programmet op, kalder programmet en metode som henter alt ned fra *DatebaseController* klassen sådan at programmet kan redigere i informationerne.



Business Process Model and Notation



Kode udsnit

Menuen har læst alt input som strings, så vi prøver at parse til "datetime". Derefter kalder vi repositoret til at lave begivenheden.

Følgende kode eksempel er fra en af vores metoder, denne metode står for at kreerer en begivenhed.

1 reference | Jes Lindekilde Johannsen, 2 days ago | 3 authors, 8 changes

```
public bool CreateEventType(bool reservation, string start, string end, string address, string comment)
{
    DateTime tend;
    DateTime tstart;
    if (!DateTime.TryParse(start, out tstart))
    {
        return false;
    }
    if (!DateTime.TryParse(end, out tend))
    {
        return false;
    }
    return eventRepository.CreateEvent(tstart, tend, address, comment, reservation);
}
```

Hvis en rustvogn skal reserveres, bliver alle rustvogne tjekket om de er frie på det pågældende tidspunkt.

Dette sker i rækkefølge af rustvognenes prioritering, og den første fri bliver tildelt.

```
// The CreateEvent method takes a number of parameters which we will use to make an event in our calendar.
1 reference | Emil Helge Knochenhauer Jürge..., 2 days ago | 1 author, 1 change
public bool CreateEvent(DateTime start, DateTime end, string address, string comment, bool hearseNeeded)
{
    // Here we run an if-else loop on whether or not a hearse is needed for the event.
    if (hearseNeeded)
    {
        // If a hearse is needed for the event, then it will tell the boolean, available, to return true.
        bool Available = true;

        // It will then run through the Event list to see if a hearse is available
        // by checking if the start is later than the end or the end is before the start.
        foreach (Hearse i in HearseRepo.GetCopyHearses())
        {
            foreach (CalendarEntry _Event in Eventslist)
            {
                if (_Event.Hearse == i && ((_Event.Start > end) || _Event.End < start))
                {
                    Available = false;
                }
            }

            // If it is neither of those, it means the hearse is available
            // and the AddEvent method is called to add the newly made event to the list.
            if (Available)
            {
                CalendarEntry Event = new CalendarEntry(FindHighestKey() + 1, start, end, address, comment, status.NewlyMade, i);
                AddEvent(Event);
                return true;
            }
            else
            {
                Available = true;
            }
        }
    }
}
```

Hvis der ikke skal bruges en rustvogn, bliver begivenheden bare oprettet.

```
// If a hearse is not needed. It will simply call the method AddEvent to add the event to the list.
else
{
    CalendarEntry _Event = new CalendarEntry(FindHighestKey() + 1, start, end, address, comment, status.NewlyMade, null);
    AddEvent(_Event);
    return true;
}
return false;
```


Processen – Projektarbejde

Organisering af gruppearbejde

Uge 43 mødtes vi for at organisere gruppearbejdet og projektet.

Vi prøvede at tage kontakt til virksomheder hvor vi også fik aftalt bl.a. med hinanden den generelle struktur for vores projektarbejde hvor alle blev enige i at det skulle bære præg af en agil tankegang og udvikling ved at bruge Scrum metodologien. Vi besluttede alle at bruge Scrum da det er en fleksibel projektledelsesmetode som vi alle havde erfaring med at bruge i tidligere projekter og som vi også fik kendskab til i undervisningen.

Vi fik også udarbejdet en gruppekontrakt, således at vores forventninger til hinanden bliver overholdt, udeblivelse i form af sygdom og andre årsager blev underrettet til gruppen i god tid m.m.

Torsdag den 8. november i uge 45 fik gruppen en ny femte medlemmer.

Etablering af kontakt med virksomhed

Vi havde meget besvær med at finde en virksomhed de kunne samarbejde med efter gruppedannelsen i uge 43, som også gjorde at vi kom relativt bagud end andre grupper der havde fået godkendt en virksomhed og problemformulering.

Det lykkedes os at få godkendt en virksomhed, torsdag den 8. november i uge 45, i gennem vores nye femte medlem.

Vi tog straks kontakt til virksomheden, og dens ejer: Bedemanden Peter Klæmmensen, hvor vi fik aftalt et første møde den 29. november.

Peter Klæmmensen som er ejeren af den virksomhed vi samarbejdede med skulle agere som vores Product Owner igennem projektet, og altså den eneste person vi skulle holde møder med.

Derudover blev vi enige om at alt kommunikation skulle foregå via Peters e-mail og hvis det var nødvendigt, at ringe til ham direkte på hans telefon nummer

Der blev formuleret en plan for møder, som blev sendt til Peters mail.

I mødeplanen blev der taget højde for evt. ændringer i vores sprints.

Vores første møde blev afholdt på en fredag men i det anden møde planlagde vi det tidligst på ugen da vores Product Owner havde mest travlt senere på ugen.

Det første møde var et opklarende møde, hvor vi kunne stille spørgsmål, så vi kunde komme videre med udviklingen af systemet. Det andet og sidste møde vi havde, med vores Product Owner blev brugt til fremvisning af programmet, således at Peter kunne vurdere hvorvidt vi var på rette spor. Til sidst tog vi kontakt med en underviser, som godkendte vores projekt.

Uforudsete problemer

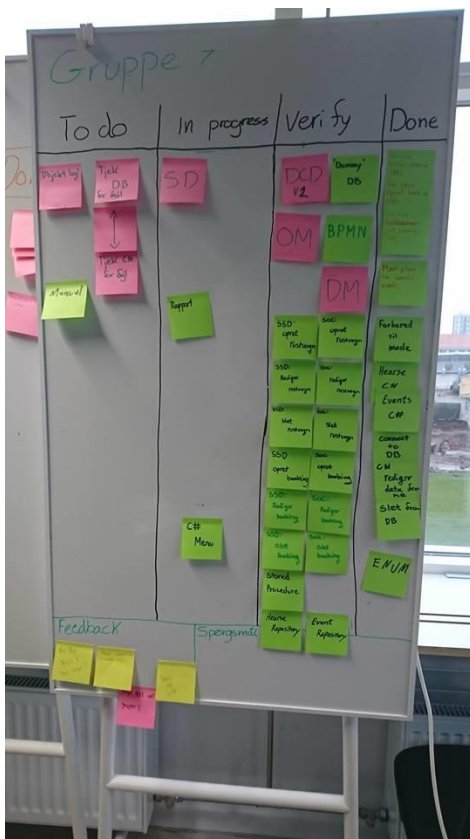
Gruppen har gennem hele projektprocessen haft det hårdt med uforudsete problemer i form af masser ting.

Disse er de mest primære problemer som har enten sænket og påvirket det endelige produkt:

1. Gruppen fik en virksomhed relativt sent som resulterede i at man skulle bruge tiden på at analysere virksomheden, skrive business case, udarbejde artefakter i første sprint m.m. som burde have været udført tidligere efter gruppens dannelse.
2. Der var meget sygdom i gruppen som gjorde at ikke alle medlemmerne altid var tilstede under projektarbejdet.
3. Vi fik også sent at vide nær afslutningen af vores projekt, at vi skulle omskrive vores kode så det havde et objekt lag så det kunne stemme overens med vores artefakter. Da meget af vores programmerings arbejde afhang af at bruge databasen for det meste, skulle vi ændre det så databasen fik mindre trafik og bruge c# programmet til at lave redigeringerne da det var bedre for fleksibilitet.

Scrum

Vi har benyttet os af scrum til at organisere vores projekt, med scrum møder hver anden dag for at holde overblikket. Hvem der har fungeret som scrum master har roteret rundt i gruppen alt efter hvem der tog initiativet på den pågældende dag. Ud over det har vi holdt et project backlog møde med vores PO en gang hvert sprint.



Brugervejledning

For at vælge et menupunkt naviger menuen men piletasterne, og brug enter for at vælge det markerede menupunkt.

Afhængigt af hvilket menupunkt du har valgt, vil du blive bedt om at give et input:

1. Hvis du har valgt at oprette en ny begivenhed, skal du først vælge start tidspunktet for din begivenhed her har du en række af mulige formater du kan skrive som er detaljeret neden for derefter trykker du enter. Derefter skal du vælge slut tidspunktet for begivenheden som vælges på samme måde. Derefter vælg om der skal bruges en rustvogn til begivenheden det kan gøres ved at skrive enten "j" eller "n" derefter trykker du enter. Du skal vælge en adresse som du skriver og trykker enter når du er færdig. Du gør det samme for kommentarer begge kan efterlades tomme.
2. Hvis du har valgt at rette en begivenhed, skal du først vælge hvilken begivenhed du vil redigere ved at indtaste surrogatnøglen starttidspunktet for din begivenhed her kan du efterlade den tom for ikke at lave nogle ændringer. Derefter skal du vælge slut tidspunktet for begivenheden som vælges på samme måde. Derefter vælg om der skal bruges en rustvogn til begivenheden det kan gøres ved at skrive rustvognens nummer eller skrive 0 for ikke at have nogle rustvogne. Du skal vælge en adresse som du skriver og trykker enter når du er færdig. Du gør det samme for kommentarer begge kan efterlades tomme hvis du ikke ønsker nogle ændringer.
3. Hvis du har valgt at slette en begivenhed, skriver du surrogatnøglen for den begivenhed du vil have slettet.
4. Hvis du har valgt at opdater databasen gemmer du alle ændringer du har lavet til databasen.
5. Hvis du har valgt afslutte program bliver dine ændringer gemt og programmet bliver lukket.

Du kan vælge at skrive datoen på flere forskellige måder du kan skrive dagen, måneden, og året sammensat med bindestreger efterfulgt af timer, minutter og sekunder sammensat med kolon på denne måde "13-02-19 13:30:55" du kan undlade sekunder "13-02-19 13:30" eller timer, minutter og sekunder "13-02-19". Du har også muligheden for at erstatte bindestregerne med mellemrum "13 02 19 13:30:55" eller erstatte måneden med dens tre bokstavsnavn "13 feb 19 13:30:55".

Kvalitetssikring

Vurderingskriterier

Objektmodel:

- Sporbarhed
- Sammenspil mellem objekter
- Læselig
- Udførlig
- Objekter afspejler virkelighed

Domænemodel:

- Sporbarhed
- Rumme realistiske eksempler
- Vise alle eksempler fra objektmodel
- Let læselig
- Illustrere sammenspil i IT-systemet

Use case

- Sporbarhed
- Realistisk
- Let forståelig
- Beskriver en interaktion mellem aktør og system

System Sekvens Diagram

- Sporbarhed
- Ét scenarie

Sekvensdiagram

- Sammenspil mellem objekter
- Relevans
- Omsættelighed

Design Klasse Diagram

- Sporbarhed
- Affinitet
- Omsættelighed

Kodestandard

I dette projekt har vi valgt at gå med engelsk til alt programmering, eksklusivt den programmering som bliver vist på konsolapplikationen, som vores product owner skal se og arbejde med. Derudover har vi gået med camel casing for alle vores variabler og pascal casing til vores metoder og properties.

Gloser

Gloseliste

Engelsk	Dansk
Adress	Adresse
CalendarEntry	Begivenhed
Event	Begivenhed
Booking	Begivenhed
Create	Opret
Alter	Rediger
Hearse	Rustvogn
Update	Updater

Konklusion

Peter Klæmmensens nuværende system kan ikke tildele en medarbejder til en begivenhed og det betyder derfor at han er ofte nødsaget til at bruge en del tid på at koordinerer med de andre medarbejdere. Vi har sammen med PO arbejdet på en kalender som skal gøre det nemmere og hurtigere for dem at koordinere med hinanden.

Vi har benyttet os af systemudviklingsartefakter til at udvikle vores løsning samt en agil arbejdsproces for at sikre en optimal arbejdsmoral og fremgang. Dette har resulteret i nye ideer som vi har integreret i vores løsning.

Derudover har vi opstillet kvalitetskriterier for at sikre at vores løsning er opbygget på en nem og overskuelig måde i tilfælde af at PO vil arbejde videre med vores koncept.