

DAT18A

Project Participants:

Martin Kjeldgaard Nikolajsen, 100183

Philip Thomas Barkow, 160293

Rasmus Neo Lassen, 070373

Frederik Nyborg Skou, 030291

Marcus Sangill Hindsbøl, 231297

Document Version: 00.07.00

1) Vision	2
Virksomhedsbeskrivelse	2
Vision	3
2) Diagrammer:	4
Domain Model:	4
Klassediagram:	4
3) Krav	4
4) Use Cases	6
USE CASE - BRIEF	6
For Sekretæren:	8
For Facilitator:	9
USE CASE - CASUAL:	9

UC4: Administrer Begivenhedstype.uf	9
UC2: Administrer Arrangement.	10
UC8: Importer Data.	12
USE CASE - FULLY DRESSED	12
UC2: Administrer Arrangement.	12
UC4: Administrer Begivenhedstype.	14
UC8: Importer Data.	15
System Sequence Diagram:	16
Sequence Diagram:	18
5) Supplementary Specification	18
FURPS+	18
Functional Requirements	18
Non-functional requirements	20
OOP:	20
Hvordan programmet virker:	20
6) Other Information?	24
Faseplan:	24
7) Appendix	25
Glossary	25

Historik

Date	Version	Author	Description
2018-19-11	00.01.00	<i>Nikolajsen, Barkow, Lassen, Skou.</i>	First Draft, Krav.
2018-20-11	00.02.00	<i>Nikolajsen, Barkow, Lassen, Skou.</i>	(F)URPS. Use Case.
2018-21-11	00.03.00	<i>Nikolajsen, Barkow, Lassen, Skou.</i>	(F)URPS. Use Case videreudvikling samt konkretisering af ikke funktionelle krav samt opsætning af funktionelle krav i Visual Paradigm.
2018-23-11	00.04.00	<i>Nikolajsen, Barkow, Lassen, Skou, Hindsbøl.</i>	

2018-04-12	00.05.00	<i>Nikolajsen, Barkow, Lassen, Skou, Hindsbøl.</i>	Vision, Krav, VP Diagrammer, Use Case.
2018-05-12	00.06.00	<i>Nikolajsen, Barkow, Lassen, Skou, Hindsbøl.</i>	ITO+OOAD Vision, Diagrammer.
2018-06-12	00.07.00	<i>Nikolajsen, Barkow, Lassen, Skou, Hindsbøl.</i>	Glossary, SD Diagrammer, Program Beskrivelse.
2018-07-12	00.08.00	<i>Nikolajsen, Barkow, Lassen, Skou, Hindsbøl.</i>	Finpudsning.

1) Vision

Virksomhedsbeskrivelse

PlanOrgan er et eventbureau med en klar målsætning om at skabe merværdi hos kunderne igennem skræddersyede events og konferencer. PlanOrgan har gjort det til sit speciale at kortlægge kundernes behov og ønsker til tiden uden at gå på kompromis med kvaliteten. Lige fra den lille firmafest til den store festival oplevelse, stimulerer PlanOrgan virksomheders motivation til at gøre noget ekstra for deres medarbejdere. Igennem storslåede arrangementer og oplevelser, sørger PlanOrgan for at give alle de deltagende en uforglemmelig aften, der huskes.

Hos PlanOrgan er kreativiteten og storheden i højsædet, hvilket giver sig udslag i tesen om, at "alt kan lade sig gøre". PlanOrgan sørger for at afdække samtlige krav til arrangementet, der altid starter med en grundig analysefase sammenholdt med de ressourcer, der er til rådighed. For PlanOrgan handler det om at skabe værdi til kunden med sikker styring, planlægning og eksekvering. At planlægge og udføre et event kræver stor erfaring og know-how, som de fleste virksomheder ikke besidder. Derfor er PlanOrgan det sikre valg som strategisk sparringspartner, når eventplanlægningen skal foretages.

PlanOrgan blev stiftet af Hans Kristen i 2015 med det formål af blive virksomhedernes fortrukne eventbureau i Danmark. Ud over Hans Kristen sidder Grethe Wirkmann og holder styr på arrangementerne, som på 3 år er firdoblet. I marken har Hans Kristen ansat 3 facilitatorer/projektledere til at stå for den daglige kontakt med kunderne og følge produktionerne i mål.

Kulturen hos PlanOrgan er skabt på baggrund af en klart defineret strategi, hvor kommunikation og grundigt fodarbejde er nøgleordene i de løsninger, der bliver valgt, og er

grundpillen i PlanOrgans virke som eventbureau.

Organisationsformen hos PlanOrgan er organisk lige fra arbejdsdelingen til magtdelingen, hvor alle tager del i ansvaret, og hvor interaktionen er tværgående. PlanOrgans styrke er den flade struktur, hvor alle bidrager med viden og erfaring, og hvor man hurtigt kan omstille sig og reagere på nye krav.

PlanOrgan har sammenfattet 6 punkter i deres organisationsform, der hver især skal bidrage til den daglige kreative tankegang, og samtidig være katalysator for innovation og nytænkning:

- At tonen og ledelsesformen hos PlanOrgan er uformel
- At ansvaret og beslutningsprocesserne hos PlanOrgan ligger hos de ansatte
- At hierarkiet hos PlanOrgan er fladt, også hvad kommunikation angår
- At PlanOrgan lægger vægt på fleksibilitet, for at opnå så stor dynamik som muligt i hverdagen
- At PlanOrgan skal kunne tage hurtige beslutninger
- At alle ansatte hos PlanOrgan bidrager med deres kompetencer og viden

Ydermere har Hans Kristen sørget for at skabe den rette selvforståelse af PlanOrgans grundværdier, så alle de ansatte føler sig opmuntret og engageret til at yde det ekstra, der skal til.

Strategien kan også ses på bundlinjen, og gjorde, at man sidste år kunne løfte Børsens Gazelle pris over hovedet for bedste innovative virksomhed 2017.

Vision

PlanOrgan er vokset så hurtigt, at de har svært ved at holde styr på deres kunders arrangementer. Applikationen som vi skal udvikle, har til formål at gøre det lettere at organisere og holde styr på virksomhedens hverdag.

Applikationen skal kunne oprette, redigere og slette arrangementer og have mulighed for at tilknytte forskellige virksomheder, foreninger eller personer til de forskellige events. Applikationen skal kunne eksportere informationerne til filer, der kan sendes til virksomhedens facilitatorer.

2) Diagrammer:

Domain Model:

Secretary	Facilitator	Arrangement	Event	Event type
	-Name : string -PhoneNumber : int -Email : string -Info : string	-CustomerContactInfo : string -Event : string[] -Facilitator : string[] -TotalPrice : double -DateTimeStart : int -DateTimeEnd : int -Description : string -Isdone : boolean -isPayed : boolean	-Eventtype : string[] -DateTime Start : int -DateTimeEnd : int -Time : int -Price : double -Facilitator : string[] -Description : string	-Price : double -Location start : string -Location end : string -time : int -Name : string -Contact Info : string[] -Description : string -ExternalContactInfo : string
	Customer -Name : string -PhoneNumber : int -Email : string -Info : string -Company : string			
	ExternalContacts -Name : string -Email : string -PhoneNumber : int -Company : string -Info : string			

Klassediagram:

Vi har separeret grafik, logik og data. EOGUI står for grafik, EOManager står for logikken og databasen står for opbevaring af data, som vi taler med gennem EODatabaseInterface.

Dette kan ses som bilag under Klassediagram 2, Databasediagram.

Vi har lavet en klasse der hedder EOPanel, og den arver alle andre paneler fra. Alle paneler taler med EUGUI. Denne del af programmet er kun grafik. EOGUI er framen for alle paneler. Dette kan ses som bilag under Klassediagram 1, Guidiagram.

3) Krav

Krav indsamlet fra opgavebeskrivelsen vedrørende projektet "Event Organizer" til virksomheden **PlanOrgan**

Krav 1	Virksomhedens sekretær skal kunne oprette, redigere og slette arrangementer, og have mulighed for at tilknytte forskellige virksomheder, foreninger eller personer til begivenhederne.
Krav 2	Sekretæren skal herefter have mulighed for at eksportere informationerne til en fil, der kan sendes til virksomhedens medarbejdere der udføre begivenheden.

Krav 3	Hver facilitator har kun adgang, til at læse de lister de har hentet ind i deres egen applikationen. De kan altså ikke rediger og eksporter oplysningerne.
Krav 4	Et arrangement kan bestå af en eller flere "begivenheder", der kan variere i tidslængde
Krav 5	<p>For hver begivenhed der tilhører et arrangement, skal der mindst tages højde for følgende:</p> <ul style="list-style-type: none"> • Begivenhedens navn og beskrivelse • Begivenhedstyper (disse typer fastsætter i selv i jeres udviklingsteam) • Dato og tidspunkt for arrangementets start • Dato og tidspunkt for arrangementets afslutning • Information om den ansvarlige facilitator for den pågældende begivenhed • Fritekst beskrivelse
Krav 6	Ved oprettelse af en begivenhed skal det koste en pris i kr uden moms
Krav 7	<p>Et arrangement skal bestå af følgende:</p> <ul style="list-style-type: none"> • Arrangement navn • Varighed • Total pris med og uden moms
Krav 8	Applikationen skal kunne virke på Windows 7 og Windows 10 med alle funktionaliteter som sekretæren skal kunne (se evt. tidligere definerede krav for nærmere specificering).

Fig. 1. Supplerende tabel vedrørende skrive/slette/opdater rettigheder samt export/import af data for sekretæren og facilitatoren

	Sekretær	Facilitator
CRUD Arrangement	X	
CRUD Begivenhed	X	
CRUD Begivenhedstype	X	
CRUD Kunde kontakt information	X	
CRUD Facilitator kontakt	X	

information		
CRUD Ekstern kontakt information	X	
Export af data	X	
Import af data		X
Læs data	X	X

Som supplerende kommentar til slet rettigheden skal det siges at vi ikke på noget tidspunkt reelt sletter data men i stedet gør dem ikke synlige i systemet ved i stedet at give dataen en status som deaktiveret. På denne måde ser det ud som om at man sletter data i systemet uden at miste tidligere indtastet data med undtagelse af Arrangementer, Begivenheder, Kunde kontakt information. Disse slettes helt.

4) Use Cases

USE CASE - BRIEF

Vi har lavet wireframes til adskillige af vores Use Cases. Disse wireframes har vi brugt i selve kodningen af panelerne, men slut resultatet er i mange tilfælde lidt anderledes end de oprindelige wireframes. De oprindelige Wireframes kan findes i VP projektet.

Oprindelig wireframe: Start Menu

Start

Opret arrangement

Eksporter

Administrer facilitatorer

Administrer begivenhedstyper

Arrangementer

☒ Vis kun ikke afholdte

Nulstil sortering

Dato start	Dato slut	Navn	Facilitator	Er betalt	Er afholdt
1/9 2018 20:00	1/9 2018 23:00	KEA Intro	Brian Larsen	<input checked="" type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	KEA Intro	Brian Larsen	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
1/9 2018 20:00	1/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>
2/9 2018 20:00	2/9 2018 23:00	Futfut	Brain the man	<input type="checkbox"/>	<input type="checkbox"/> <div>Åbn</div> <div>Rediger</div> <div>Slet</div>

1, 2, 3, ..., 8, 9, 10

Den færdige Start Menu:

Event Organizer Administration 1.1

Start menu

Administrer Begivenhedstyper

Administrer Facilitatorer

Eksporter

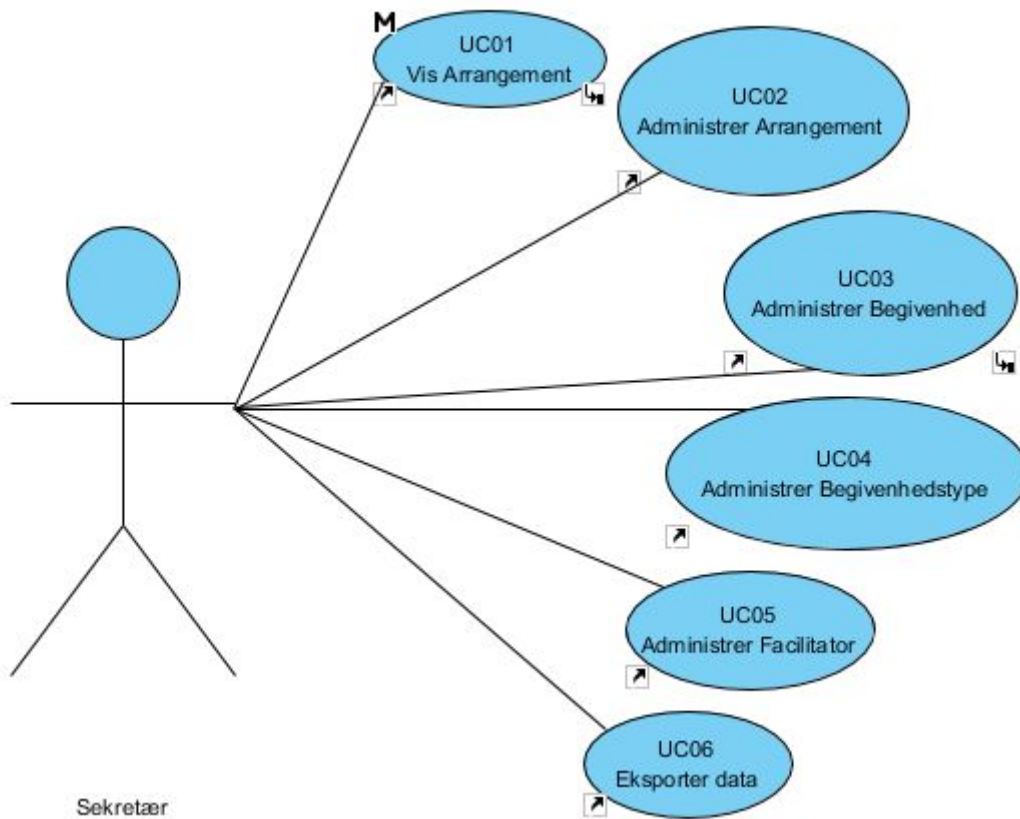
Opret Arrangement

Arrangementer

☒ Vis kun ikke afholdte

id	Dato start	Dato slut	Navn	Facilitator(er)	Er betalt	Er afholdt
----	------------	-----------	------	-----------------	-----------	------------

For Sekretæren:



UC1: Vis Arrangement.

Sekretæren starter programmet og trykker på et arrangement på listen. Sekretæren bliver præsenteret for et side med den tilgængelige information om det valgte arrangement.

UC2:Administrer Arrangement.

Sekretæren har valgt et arrangement på listen over arrangementer og bliver vist den information der er tilgængelig for det valgte arrangement eller trykker på knappen Opret Arrangement. Sekretæren kan også trykke på REDIGER eller SLET på listen over arrangementer i Start Menuen.

UC3: Administrer Begivenhed.

Sekretæren trykker på knappen Opret Arrangement og trykker derefter på knappen Opret under begivenheds listen. Sekretæren kan ændre, slette eller oprette nye begivenheder. Sekretæren udfylder derefter den nødvendige information og trykker gem.

UC4: Administrer Begivenhedstype.

Sekretæren trykker på knappen Administrer Begivenhedstype fra Start Menuen. Sekretæren udfylder den nødvendige information og trykker gem.

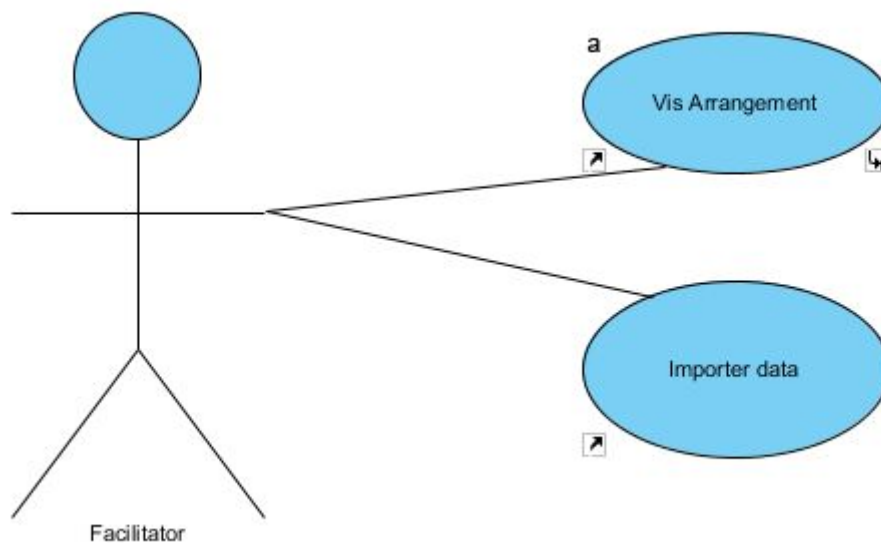
UC5: Administrer Facilitator.

Sekretæren trykker på knappen Administrer Facilitator fra Start Menuen. Her kan sekretæren oprette nye facilitatorer eller ændre på nuværende facilitatorer. Sekretæren udfylder den nødvendige information og trykker på gem.

UC6: Eksporter Data.

Sekretæren trykker på knappen Exporter. Her kan sekretæren vælge hvilken data der ska eksporter eller om alt data skal eksporter. Den valgte data bliver gemt som en CSV fil.

For Facilitator:



UC7: Vis Arrangement.

Facilitatoren trykker på det valgte arrangement fra listen i Start Menuen. Facilitatoren får den information der er tilknyttet arrangementet.

UC8: Importer Data.

Facilitatoren trykker på knappen Importer fra Start Menuen. Facilitatoren vælger hvilken data der skal importeres og trykker derefter på Import og får en CSV fil med den valgte data.

USE CASE - CASUAL:

UC4: Administrer Begivenhedstype.uf

1. Systemet viser Start Menu.
2. Sekretæren trykker på knappen Administrer Begivenhedstyper.
3. Sekretæren indtaster den nødvendige information.
4. Sekretæren trykker på knappen GEM.
5. Systemet gemmer de indtastede oplysninger i databasen.

6. Systemer viser sekretæren Start Menuen.

Alternative Flow 1:

3a:

1. Sekretæren vælger en Begivenhedstype fra listen.
2. Sekretæren trykker på knappen REDIGER.
3. Systemet viser data fra den valgte Begivenhedstype.
4. Sekretæren redigerer de nødvendige informationer.
5. Sekretæren trykker på knappen Gem Ændringer.
6. Systemet gemmer de indtastede oplysninger i databasen.

Alternative Flow 2:

3b:

1. Sekretæren vælger en Begivenhedstype fra listen.
2. Sekretæren trykker på knappen SLET.
3. Systemet sletter den valgte Begivenhedstype fra databasen.

Alternative Flow 3:

5a: Forkert angivet Varighed.

1. Systemet viser et popup vindue der fortæller sekretæren at Varighed skal indtastet som tal.
2. Sekretæren trykker OK.
3. Sekretæren indtaster de rigtige oplysninger.
4. Systemet gemmer de indtastede oplysninger i databasen.
5. Systemer viser sekretæren Start Menuen.

Alternative Flow 4:

5b: Forkert angivet Pris.

6. Systemet viser et popup vindue der fortæller sekretæren at Pris skal indtastet som tal.
7. Sekretæren trykker OK.
8. Sekretæren indtaster de rigtige oplysninger.
9. Systemet gemmer de indtastede oplysninger i databasen.
10. Systemer viser sekretæren Start Menuen.

UC2: Administrer Arrangement.

Main Success Scenario:

1. Systemet viser Start Menu.
2. Sekretæren trykker på knappen Opret Arrangement.
3. Sekretæren udfylder de nødvendige information.
4. Sekretæren trykker på knappen GEM

5. Systemet gemmer de indtastede oplysninger i databasen.
6. Systemet viser sekretæren Start Menuen.

Alternative Flow 1:

2a:

1. Sekretæren trykker på REDIGER under det ønskede arrangement på listen.
2. Systemet åbner Rediger Arrangement og viser den tilgængelige information.
3. Sekretæren redigerer i arrangementet.
4. Sekretæren trykker på knappen OPDATER.
5. Systemet opdaterer de givne informationer.
6. Systemet viser sekretæren Start Menuen.

Alternative Flow 2:

2b:

1. Sekretæren trykker på SLET under det ønskede arrangement på listen.
2. Systemet viser et popup vindue og spørger om sekretæren er sikker på at man vil slette.
3. Sekretæren trykker på knappen SLET.
4. Systemet sletter det valgte arrangement.
5. Systemet viser sekretæren Start Menuen.

Alternative Flow 3

2d:

1. Systemet viser Start Menu.
2. Sekretær vælger et arrangement fra listen og trykker på ÅBEN.
3. Systemet viser den tilgængelige information om det valgte arrangement.
4. Sekretæren udfylder den nødvendige information.
5. Sekretæren trykker på knappen GEM.
6. Systemet gemmer de indtastede oplysninger i databasen.
7. Systemet viser sekretæren Start Menuen.

Alternative Flow 4:

5a: Tidsformat er indtastet forkert.

1. Sekretæren trykker GEM.
2. Systemet prøver at gemme.
3. Systemet viser popup vindue der fortæller at tidsformater et indtastet forkert.
4. Sekretæren trykker OK.
5. Sekretæren retter fejlen.
6. Sekretæren trykker GEM.
7. Systemet gemmer.
8. Systemet viser Start Menuen:

UC8: Importer Data.

1. System viser Start Menu
2. Facilitator trykker på knappen Importer Arrangementer.
3. System viser Importer Menuen.
4. Facilitator trykker på Importer.
5. Systemet beder om facilitator om at vælge den rigtige CSV fil i et popup vindue.
6. Facilitator vælger den rigtige CSV fil.
7. Systemet importerer den valgte fil.
8. Systemet viser Start Menuen.

Alternative Flow 1:

4a:

1. Facilitator trykker på knappen Annuller.
2. Systemet viser Start Menuen.

Alternative Flow 2:

5a:

1. Facilitator trykker Cancel i popup vinduet.
2. Systemet viser Importer Menuen.

USE CASE - FULLY DRESSED

UC2: Administrer Arrangement.

Scope:

EO Manager.

Primary Actor:

Sekretær.

Stakeholder and Interests:

Sekretæren vil ændre på et valgt arrangement.

Preconditions:

Sekretæren har adgang til programmet og har startet det.

Succes Guarantee:

Det valgte arrangement bliver administreret og gemt i databasen.

Main Success Scenario:

1. Systemet viser Start Menu.
2. Sekretæren trykker på knappen Opret Arrangement.

3. Systemet viser Opret Arrangement menuen.
4. Sekretæren udfylder de nødvendige informationer.
5. Sekretæren trykker på knappen GEM
6. Systemet gemmer de indtastede oplysninger i databasen.
7. Systemet viser sekretæren Start Menuen.

Alternative Flow 1:

2a:

1. Sekretæren trykker på REDIGER under det ønskede arrangement på listen.
2. Systemet åbner Rediger Arrangement og viser den tilgængelige information.
3. Sekretæren redigerer i arrangementet.
4. Sekretæren trykker på knappen OPDATER.
5. Systemet opdaterer de givne informationer.
6. Systemet viser sekretæren Start Menuen.

Alternative Flow 2:

2b:

1. Sekretæren trykker på SLET under det ønskede arrangement på listen.
2. Systemet viser et popup vindue og spørger om sekretæren er sikker på at man vil slette.
3. Sekretæren trykker på knappen SLET.
4. Systemet sletter det valgte arrangement.
5. Systemet viser sekretæren Start Menuen.

Alternative Flow 3

2d:

1. Systemet viser Start Menu.
2. Sekretær vælger et arrangement fra listen og trykker på ÅBEN.
3. Systemet viser den tilgængelige information om det valgte arrangement.
4. Sekretæren udfylder den nødvendige information.
5. Sekretæren trykker på knappen GEM.
6. Systemet gemmer de indtastede oplysninger i databasen.
7. Systemet viser sekretæren Start Menuen.

Alternative Flow 4:

6a: Tidsformat er indtastet forkert.

1. Sekretæren trykker GEM.
2. Systemet prøver at gemme.
3. Systemet viser popup vindue der fortæller at tidsformater et indtastet forkert.
4. Sekretæren trykker OK.
5. Sekretæren retter fejlen.
6. Sekretæren trykker GEM.
7. Systemet gemmer.
8. Systemet viser Start Menuen:

Special requirements:

Ingen

Frequency of Occurrence:

Meget ofte.

UC4: Administrer Begivenhedstype.**Scope:**

EO Manager.

Primary Actor:

Sekretær.

Stakeholder and Interests:

Sekretæren vil administrer begivenhedstyper.

Preconditions:

Sekretæren har adgang til programmet og har startet det.

Succes Guarantee:

Den valgte begivenhedstype bliver administreret og gemt i databasen.

Main Success Scenario.

1. Systemet viser Start Menu.
2. Sekretæren trykker på knappen Administrer Begivenhedstyper.
3. Sekretæren indtaster den nødvendige information.
4. Sekretæren trykker på knappen Opret Ny Behivenhedstype.
5. Systemet gemmer de indtastede oplysninger i databasen.
6. Systemer viser sekretæren Start Menuen.

Alternative Flow 1:

3a:

1. Sekretæren vælger en Begivenhedstype fra listen.
2. Sekretæren trykker på knappen REDIGER.
3. Systemet viser data fra den valgte Begivenhedstype.
4. Sekretæren redigerer de nødvendige informationer.
5. Sekretæren trykker på knappen Gem Ændringer.
6. Systemet gemmer de indtastede oplysninger i databasen.

Alternative Flow 2:

3b:

1. Sekretæren vælger en Begivenhedstype fra listen.
2. Sekretæren trykker på knappen SLET.
3. Systemet sletter den valgte Begivenhedstype fra databasen.

Alternative Flow 3:

5a: Forkert angivet Varighed.

1. Systemet viser et popup vindue der fortæller sekretæren at Varighed skal indtastet som tal.
2. Sekretæren trykker OK.
3. Sekretæren indtaster de rigtige oplysninger.
4. Systemet gemmer de indtastede oplysninger i databasen.
5. Systemet viser sekretæren Start Menuen.

Alternative Flow 4:

5b: Forkert angivet Pris.

1. Systemet viser et popup vindue der fortæller sekretæren at Pris skal indtastet som tal.
2. Sekretæren trykker OK.
3. Sekretæren indtaster de rigtige oplysninger.
4. Systemet gemmer de indtastede oplysninger i databasen.
5. Systemet viser sekretæren Start Menuen.

Special requirements:

Ingen

Frequency of Occurrence:

Meget ofte.

UC8: Importer Data.

Scope:

EO Manager.

Primary Actor:

Facilitator.

Stakeholder and Interests:

Facilitator vil importere sine arrangementer.

Preconditions:

Facilitator har adgang til programmet og har startet det.

Facilitator har fået tilsendt en CSV fil af sekretæren.

Succes Guarantee:

Den valgte CSV fil bliver importeret og kan tilgås af facilitator.

Main Success Scenario:

1. System viser Start Menu
2. Facilitator trykker på knappen Importer Arrangementer.
3. System viser Importer Menuen.
4. Facilitator trykker på Importer.
5. Systemet beder om facilitator om at vælge den rigtige CSV fil i et popup vindue.
6. Facilitator vælger den rigtige CSV fil.
7. Systemet sletter data i databasen og importerer den valgte fil.
8. Systemet viser Start Menuen.

Alternative Flow 1:

4a:

1. Facilitator trykker på knappen Annuller.
2. Systemet viser Start Menuen.

Alternative Flow 2:

6a:

1. Facilitator trykker Cancel i popup vinduet.
2. Systemet viser Importer Menuen.

Special requirements:

Ingen

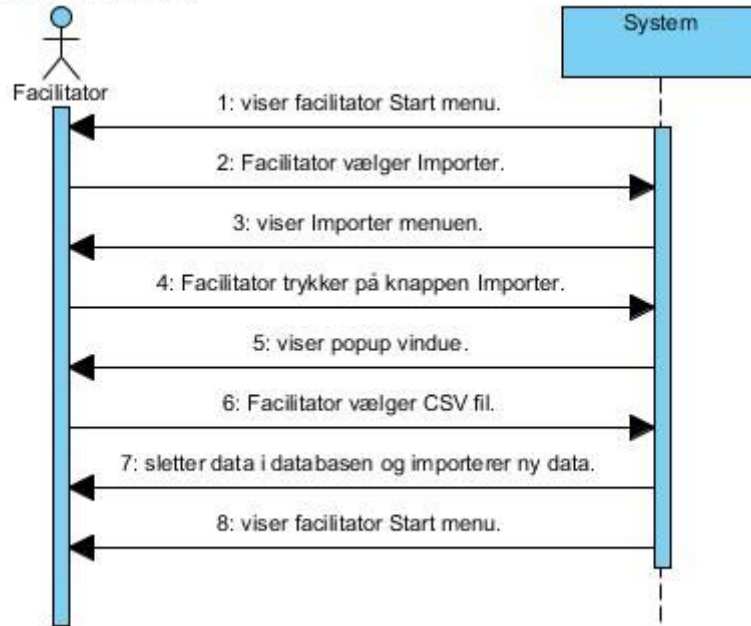
Frequency of Occurrence:

Meget ofte.

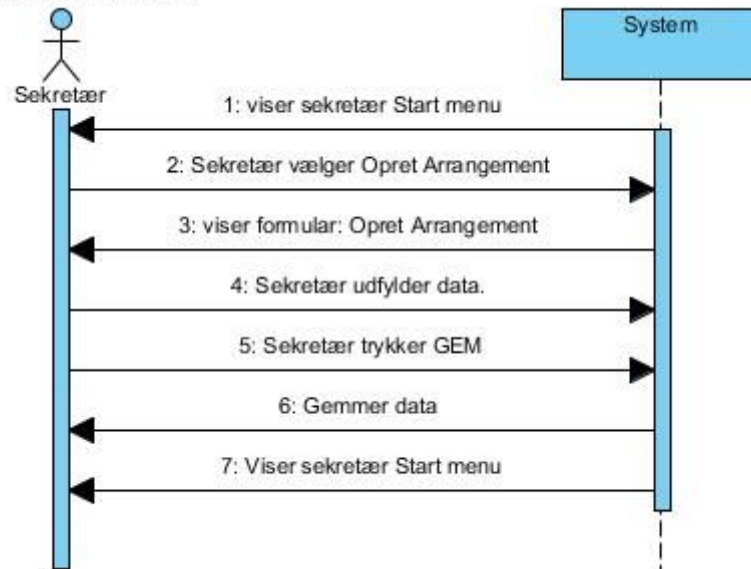
System Sequence Diagram:

Diagram Bilag

UC8: Importer Data.

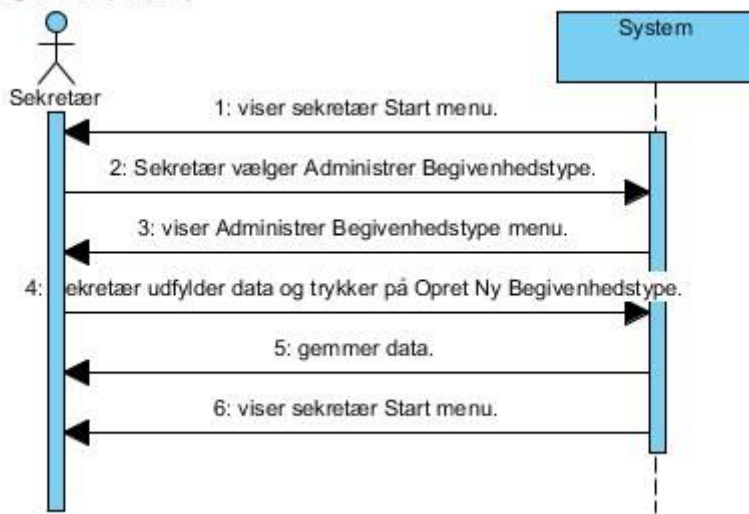


UC2: Administrer Arrangement.



UC4: Administrer Begivenhedstype.

Visual Paradigm Standard (Fred(KEA))



Sequence Diagram:

Vi har lavet Sequence Diagrammer for de tidligere beskrevne 3 Use Cases.

SD - Facilitator Import Data.

SD - Opret Arrangement.

SD - Opret Begivenhedstype.

Vi har har lagt disse diagrammer som bilag.

5) Supplementary Specification

FURPS+

Functional Requirements

Id	Requirement	Comment
FR001	Sekretæren skal kunne oprette et arrangement.	
FR003	Sekretæren skal kunne redigere et arrangement.	
FR004	Sekretæren skal kunne slette et arrangement.	
FR005	Sekretæren skal kunne tilknytte forskellige virksomheder til begivenheder.	
FR006	Sekretæren skal kunne tilknytte foreninger til forskellige begivenheder.	

FR007	Sekretæren skal kunne tilknytte personer til forskellige begivenheder..	
FR008	Facilitatorer skal kun kunne læse de filer de er tilknyttet.	
FR009	Facilitatorer skal ikke kunne redigerer arrangementer.	
FR010	Facilitatorer skal ikke kunne oprette arrangementer.	
FR011	Facilitatorer skal ikke kunne slette arrangementer.	
FR012	Et arrangement kan bestå af en eller flere begivenheder.	
FR013	Et begivenhed kan variere i længde.	
FR014	For hver begivenhed der tilhører et arrangement, skal der mindst tages højde for følgende: begivenhedsnavn og beskrivelse. Begivenhedstype (disse typer fastsætter i selv i jeres udviklingsteam). Dato og tidspunkt for arrangementets start. Dato og tidspunkt for arrangementets afslutning. Information om den ansvarlige facilitator for det pågældende event. Fritekst beskrivelse.	
FR015	Ved oprettelse af et event skal det koste en pris i kr uden moms.	
FR016	Et arrangement skal bestå af følgende: -Arrangement navn. -Varighed. -Total pris med og uden moms.	
FR017	Applikationen som sekretæren skal bruge skal kunne benyttes på Windows.	
FR018	Sekretæren skal kunne eksportere informationer til filer.	

Non-functional requirements

Id	Requirement	Comment
NFR001	Der skal udfærdiges et dokument med al dokumentation vedrørende designet af applikationen	Gøres med henblik på at kunden selv kan vælge hvem der evt. skal videreudvikle systemet
NFR002	Applikationen skal kunne eksekveres på Windows 7 og Windows 10	
NFR004	Applikationen skal kunne eksekveres på Windows 7, Windows 10 og Android	

OOP:

Event Organizer

Vi har lavet 2 forskellige programmer, et til facilitatoren og et til sekretæren, som overholder de krav som kunden har til hver af disse programmer.

Sekretær programmet består af følgende filer (der skal ligge i samme mappe)

- 1) Database.db
- 2) sqlite-jdbc-3.23.1.jar
- 3) EventOrganizer.jar

Sekretæren starter programmet ved at dobbelt klikke på EventOrganizer.jar

Facilitator programmet består af følgende filer (der skal ligge i samme mappe)

- 1) Database.db
- 2) sqlite-jdbc-3.23.1.jar
- 3) EventOrganizerFacilitator.jar

Facilitatoren starter programmet ved at dobbelt klikke på EventOrganizerFacilitator.jar

Hvordan programmet virker:

Programmet kan groft sagt opdeles i 4 dele.

Paneler (EOPanelXX: Taler med EOGUI)

GUI (EOGUI: Taler med EOManager)

Logik (EOManager: Taler med EOGUI og EODatabaseInterface)

Database (EODatabaseInterface: Taler med EOManager)

Vores data ligger i klasserne:

EOArrangement.

EOEvent.

EOEventType.

Alle paneler hedder EOPanel efterfulgt af navnet, f.eks. EOPanelStart eller EOPanelCreateEvent. GUI komponenter der er lavet til disse hedder EOGUI efterfuldt af navnet f.eks. EOGUIMultiSelect.

Interfaces der er brugt, hedder alle sammen Interface til sidst i navnet, f.eks. EOCSVInterface. Med undtagelse af EODatabaseInterface, som ikke er et interface.

Til at styre hvad EOGUI skal vise, bruger vi en enum type, der hedder DisplayType. Til at styre hvilken operation vi er igang med, har vi en enum type der hedder EOOperation. Ud over at være en Enum, har vores Enum type EOOperation også mulighed for at overføre et Objekt. Den har et default navn, der kan vises og en default Enum DisplayType.

Dvs. hvis man f.eks. trykker på en knap i vores panel: EOPanelStartMenu hvor der står Opret Arrangement

- så bliver EOGUI.runCommand kaldt med EOOperation.CREATEARRANGEMENT,
- EOGUI.runCommand kalder så EOManager.runCommand med EOOperation.CREATEARRANGEMENT
- EOManager ser hvad der skal gøres med evt. database (f.eks. skal der hentes diverse facilitatorer ud af databasen)
- EOManager.runCommand returnere EOOperation.CREATEARRANGEMENT til EOGUI som nu indeholder de facilitatorer der skal vises på panelet.
- EOGUI læser hvilke enum DisplayType som EOOperation.CREATEARRANGEMENT indeholder og viser dette panel, samtidig bliver EOOperation.CREATEARRANGEMENT sendt videre til panelet som så kan tage data ud af EOOperation.CREATEARRANGEMENT og fylde ind i panelets GUI komponenter.

Kundens krav:

For at møde kundens krav, har vi lavet det sådan, at man kan oprette arrangementer. Et arrangement kan så indeholde en eller flere begivenheder, hvor en begivenhed igen kan indeholde en eller flere begivenhedstyper. En begivenhedstype er defineret uden for arrangementet. Det er helt valgfrit, om man ønsker at tilføje en begivenhed til et arrangement eller ej. Man kan til hver en tid bruge beskrivelses tekstfeltet, til at skrive hvad der skal ske.

Da PlanOrgan er i kontakt med forskellige aktører, så har vi 3 forskellige kontakt typer i vores program:

ExternalContactInfo: Bruges til eksterne samarbejdspartnere

CustomerContactInfo: Bruges til kunden som har bestilt arrangementet

FacilitatorContactInfo: Angiver en medarbejder, som står for afholdelse af en begiven eller et arrangement

Eksport/Import CSV:

Vores CSV fil bliver genereret ud fra formatet

ObjectNavn1, data, data, data, data

ObjectNavn1.1 er en del af ObjectNavn1, data, data, data, data

Link ObjectNavn1, ObjectNavn1.1

Hvor vi escaper linie skift og ", ".

Vi escaper " med "".

Datotid er angivet som dag/måned år time:minut.

Da vores hoveddata klasse er EOArrangement, vil det være den vi tager udgangspunkt i, hvis der eksporteres data til en facilitator via facilitator valg eller arrangement valg.

Hvis man derimod vælger at eksportere alt data, så laves der objekter ud fra alle de ting, der er i databasen.

Et eksempel på data eksport er her (taget fra EOArrangement: exportCSV())

Linie1: "EOEXPORT", "2018-12-06T11:21:10.445", "mode=1"

Linie2: "EOArrangement", "1", "Deloitte Julefrokost", "Et brag af en julefest!", "21/5 1953 22:39", "21/5 1953 22:39", "100000.0", "false", "false"

Linie3: "FacilitatorContactInfo", "2", "Test mand 2", "12345678", "test@test.dk", "Bla"

Linie4: "Link", "EOArrangements_idEOArrangements", "1",
"EOFacilitatorContactInfo_idFacilitatorContactInfo", "2"

Linie5: "FacilitatorContactInfo", "4", "Test mand 4", "12345678", "test@test.dk", "Bla"

Linie6: "Link", "EOArrangements_idEOArrangements", "1",
"EOFacilitatorContactInfo_idFacilitatorContactInfo", "4"

Når data indlæses, starter vi med at slette alt data i databasen.

Derefter tjekkes der for, om den første linie i det første felt er EOEXPORT - Ellers er det ikke en EOEXPORT CSV fil.

Derefter, for hver linie, kigger vi på værdien i det første felt og ud fra det, opretter vi så et objekt af den type, og opretter det i databasen.

Nedarv:

Vi har brugt nedarvning flere steder i vores program. Et eksempel er fra vores abstract class ContactInfo. Denne klasse danner basis for vores 3 andre contact klasser CustomerContactInfo, FacilitatorContactInfo og ExternalContactInfo.

Grunden til at vi har brugt nedarv i lige præcis det her tilfælde, er at alle klasserne, skal kunne sætte id (fra databasen), kontakt personens navn, telefonnummer, email og generel information. ExternalContactInfo og CustomerContactInfo er klasser, som arver fra ContactInfo, og har derudover i deres klasser muligheden for at gemme hvilken firma kontaktpersonen er fra.

Et andet eksempel er EOPanel. Her arver vi fra javax.swing.JPanel for at lave vores eget panel. Her kunne vi have valgt at bruge et interface, men har valgt at bruge nedarv, da det er behændigt, at vi kan kalde metoder direkte i JPanel. Med andre ord trækker vi kun det ind, som man har brug for at få vist.

Vores nye EOPanel har de ekstra metoder

```
* public void setVisible(boolean visible, EOOperation data)
* public EOOperation getcurrentEOOperation()
```

Interface:

Vi har brugt interface en hel del steder. F.eks. bruger vi EOGUIMultiSelectInterface i EOArrangement, EOEvent, etc. Grunden til det er, at vi i klassen EOGUIMultiSelect så kan modtage objekter af typen EOGUIMultiSelectInterface. I klassen EOGUIMultiSelect har vi en liste, hvor vi udskriver getDisplayName() fra de EOGUIMultiSelectInterface objekter, som denne modtager.

Et andet eksempel på interface brug er EOCSVInterface. Det kræver, at klassen implementerer public String exportCSV(). Denne metode bruger vi så, når vi skal eksportere data i klassen til en CSV fil. Denne bliver brugt i alle de klasser, som vi har data for i databasen som f.eks. EOArrangement, EOEvent, EOEventType, ContactInfo, CustomerContactInfo, ExternalContactInfo og FacilitatorContactInfo.

Da vores CustomerInfo er hoved klassen for CustomerContactInfo, ExternalContactInfo, FacilitatorContactInfo, har vi i denne implementeret metoden public String exportCSV(String type). Dette er en hjælpemetode for subclasses til ContactInfo. Når en subklasse til ContactInfo eksporterer data, så sker det sådan her fra f.eks. CustomerContactInfo:

```
@Override
public String exportCSV()
{
    return(super.exportCSV("CustomerContactInfo") + ";" + getCompany() + "\n");
}
```

Hvor CustomerContactInfo har et ekstra felt Company.

GUI komponenter

Da vi manglet nogle GUI komponenter til styring af f.eks. dato valg, har vi bygget vores egen.

December 2018						
Man	Tir	Ons	Tor	Fre	Lør	Søn
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Dato:

Tid:

Her vælges dato ved at trykke på den dag i måneden som man ønsker at bruge. Tidspunkt angives under Tid. Vi kan så via vores EOPanel få et LocalDateTime ud af vores EOGUIDateTimePicker som vi kan bruge i vores Arrangement eller Begivenhed.

I alt er der lavet følgende GUI komponenter.

EOGUIArrangement: Bruges som tabel til at vise arrangementer på start siden.

EOGUIBreadcrumb: Bruges til at angive hvor brugeren er i programmet, og giver dem mulighed for at gå tilbage.

EOGUIMultiSelect: Er en select, som man kan bruge til at vælge en eller flere elementer fra.

Fælles for GUI komponenterne er at vi bruger et panel som default object, og så bygger videre på det ved at bruge Swing's andre GUI komponenter til at lave vores helt unikke GUI component.

6) Other Information?

Faseplan:

Uge 1:

Første udkast til domain og klassemodel.

Hvilke krav er der fra virksomheden.

Use Case.

Lave den grundlæggende kode.

Uge 2:

Det meste og vigtigste af koden skrives færdig.

Uge 3:

Koden skrives HELT færdig.

Diagrammer laves færdige i VP.

Rapporten skrives helt færdig.

Projektet er færdiggjort.

7) Appendix

Glossary

Arrangement

Et arrangement kan bestå af nul eller flere “begivenheder”, der kan variere i tidslængde. Arrangement er det overordnede element, hvori alle eventuelle begivenheder forekommer.

Begivenhed

Begivenheden/begivenhederne udgør et scenarie som udføres én gang per arrangement, hvori der kan foregå nogle begivenhedstyper. Der kan dog også bare blive lavet en beskrivelse af, hvad der skal foregå, hvis der ikke er nogle begivenhedstyper. Eks. er en fødselsdagsfest.

Begivenhedstype

Begivenhedstyper er forudbestemte, og kan blive tilføjet til en begivenhed.

Eks. er at man evt. bestiller en klovn til en fødselsdagsfest.

Her er der forudbestemt aftale med klovnen, mht. pris og varighed af begivenhedstypen og evt. ekstern kontakinformation til klovn.

Facilitator

Facilitatorerne er de ansvarlige for hele arrangementet. De har dog ofte forskellige ansvar.

En kan være ansvarlig for hele arrangementet, altså derfor lederen for arrangementet.

Der kan være nogle, som har ansvar for forskellige begivenheder og begivenhedstyper, som hænger sammen.

Facilitatorkontaktinformationer

Informationer om facilitator, og et sted hvor man kan se beskeder fra facilitator.

Kundekontaktinformationer

Informationer om kunden, og mulighed for at skrive forespørgsler i begrænset mængde.

Eksterne kontaktinformationer

Informationer om ekstern leverandør/udbyder mm.