

Udvikling af IT-system til Midtby Delebilklub

- Semesterprojekt 2008

Udarbejdet af:

12 01 88 - Agge Skov Larsen
29 11 86 - Alexey Bessonov
05 01 87 - Frantz Furrer
12 10 87 - Jakob Witte Larsen

Synopsis

I denne rapport fortsættes en udvikling af et IT-system til Midtby Delebilklub. Rapporten er et resultat af et projektarbejde, hvor der er anvendt Unified Process til strukturering og udvikling af IT-systemet. Implementeringen af IT-systemet er udført i Java.

I resultatdelen vil der blive opstillet funktionelle og ikke-funktionelle krav, udarbejdet brugsmønsterbeskrivelser, tegnet analyseinteraktionsdiagrammer og analyseklassediagram, designinteraktionsdiagrammer og designklassediagram. Afslutningsvis vil de vigtigste dele af det implementerede blive beskrevet.

Det er vist gennem bearbejdning af centrale brugsmønstre, at det er muligt at udvikle et IT-system der giver en aflastning af administratoren og en bedre service overfor delebilklubbens medlemmer. På baggrund af dokumentationen, vil der sidst i rapporten være en beskrivelse af hvordan det videre forløb kunne tilrettelægges.

Indholdsfortegnelse

1	Indledning.....	3
2	Projektbeskrivelse.....	4
2.1	Problemanalyse.....	4
2.2	Problemformulering.....	4
2.3	Mål og forventede resultater.....	4
3	Procesmodel.....	6
3.1	Metode.....	6
3.2	Udvælgelse af brugsmønstre og krav.....	6
3.3	Overordnet plan for iterationer.....	8
	RESULTATER.....	10
4	Krav.....	10
4.1	Brugsmønstermodel.....	10
4.2	Kravmodel.....	11
4.3	"Ansøgning om medlemskab (B3)".....	13
4.4	"Reservation af delebil (B4)".....	15
4.5	"Annullering af reservation (B6)".....	17
4.6	"Aflevering af delebil (B8)".....	19
4.7	"Vedligeholdelse af bil (B14)".....	21
4.8	"Analyse af bilklubbens drift (B17)".....	23
5	Analyse.....	25
5.1	Generel beskrivelse af analyseklassediagram.....	25
5.2	Analyse af "Ansøgning om medlemskab (B3)".....	27
5.3	Analyse af "Reservation af delebil (B4)".....	28
5.4	Analyse af "Annullering af reservation (B6)".....	29
5.5	Analyse af "Aflevering af delebil (B8)".....	30
5.6	Analyse af "Vedligeholdelsesreservation (B14)".....	31
5.7	Analyse af "Analyse af delebilklubbens drift (B17)".....	32
6	Design.....	33
6.1	Generel beskrivelse af designklassediagram.....	33
6.2	Design af "Ansøg om medlemskab (B3)".....	34
6.3	Design af "Reservation af Delebil (B4)".....	34
6.4	Design af "Annullering af reservation (B6)".....	35
6.5	Design af "Aflevering af delebil (B8)".....	36
6.6	Design af "Vedligeholdelsesreservation (B14)".....	36
6.7	Design af "Analyse af delebilklubbens drift (B17)".....	37
7	Implementering.....	38
7.1	Implementering af removeReservation.....	38
7.2	Implementering af checkForOverlap.....	39
7.3	Implementering af getPreviousReservation.....	40
7.4	Implementering af getKmPrice.....	41
7.5	Implementering af saveFile.....	42
7.6	Implementering af DTMFdecoder klassen.....	43
8	Resterende arbejde.....	47
8.1	Generelt resterende arbejde.....	47
8.2	Resterende arbejde i brugsmønstrene.....	47
9	Konklusion.....	49
10	Perspektivering.....	50
11	Litteraturliste.....	51
11.1	Anvendte bøger.....	51
11.2	Anvendte internetsider.....	51
11.3	Anvendte øvelser.....	51
11.4	Programliste.....	51
	Appendiks 1 – Teori for afkodning af DTMF-signaler.....	52
	Bilag 1 – Tidsplaner.....	54
	Bilag 2 – Mockup for "Reservation af delebil (B4)".....	55
	Bilag 3 – Mockup for "Vedligeholdelse af bil (B14)".....	56
	Bilag 4 – Analyseklassediagram.....	57
	Bilag 5 – Designklassediagram.....	58
	Bilag 6 – Designinteraktionsdiagram B3.....	59
	Bilag 7 – Designinteraktionsdiagram B4.....	60
	Bilag 8 – Designinteraktionsdiagram B6.....	61
	Bilag 9 – Designinteraktionsdiagram B8.....	62
	Bilag 10 – Designinteraktionsdiagram B14.....	63
	Bilag 11 – Designinteraktionsdiagram B17.....	64

Læsevejledning

- Rapporten er inddelt i kapitler, afsnit og underafsnit. Til hvert kapitel vil der være en indledning.
- Fodnotehenvisninger er et tal med hævet skrift, eksempelvis ”¹”, og bliver brugt til kildehenvisning eller en uddybende kommentar.
- Kildehenvisningen skrives som ’Kilde #1 - Side #2’, hvor #1 er nummeret på kilden i litteraturlisten og #2 er siden i kilden. Eksempelvis: ’Kilde 1 - Side 377’ betyder ’UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design’ side 377.
- Figurer, tabeller, UML-diagrammer og kildekode er nummereret efter kapitler, eksempelvis ”**Figur 1.1**”. Dette er den første figur i første kapitel.
- UML-diagrammerne er lavet i Diagram Designer¹.
- Der benyttes engelske betegnelser i kildekoden og UML-diagrammerne (for at undgå æøå). I brugsmønsterbeskrivelserne benyttes danske betegnelser, for at gøre beskrivelserne lettere at læse.
- NDB er forkortelse for NetDelebilKlub.
- Kildekode er vist i en lyseblå ramme.
- Krav og brugsmønster er prioriteret ud fra MoSCoW metoden.
- Der er fold-ud sider bagerst i rapporten, som indeholder UML-diagrammer.

¹ Diagram Designer v1.21

1 Indledning

Dette semesterprojekt omhandler Midtby Delebilklub, som er en fiktiv virksomhed, hvor flere personer sammen kan eje og bruge fælles biler. Delebilklubben ønsker et IT-system udviklet til administration af medlemmer og reservation af biler for at aflaste administratoren. IT-systemet skal desuden forbedre klubbens service overfor medlemmerne.

I projektet arbejdes der videre med udviklingen af IT-systemet, på baggrund af det til gruppen udleverede projektoplæg² som indbefatter et inceptionsdokument, hvori brugsmønstre og krav er beskrevet, et elaborationsdokument med dertilhørende kildekode samt et casedokument. Inceptions- og elaborationsdokumentet indeholder det arbejde, der er foretaget forud for projektets start. Casedokumentet indeholder interview med delebilklubbens formand, liste over priser og gebyrer, og de informationer klubben ønsker at få, når potentielle medlemmer ansøger om medlemskab.

Softwareudviklingsforløbet er blevet struktureret på baggrund af Unified Process (UP). Der er fra starten, lagt meget vægt på struktureringen af softwareprojektet, fordi dette vil resultere i et bedre og mere produktivt projektforb. Strukturering af projektforb kan ses i tidsplanerne i bilag 1 og i projektgrundlaget.

Rapporten er udarbejdet af fire studerende på SDU, Det Tekniske Fakultet på 3. semester datateknologi. Projektet indeholder fagligheder fra de fag, der er blevet undervist i på dette semester: Systemudvikling, programmering, digital signalbehandling og computersystemer. Det understreges at rapporten er udarbejdet ud fra gruppens kompetencer på 3. semester, derfor vil målgruppen for rapporten være systemudviklere og studerende med tilsvarende fagligt niveau.

Den del af rapporten der henvender sig til kunden er primært krav, brugsmønstrebeskrivelser og til dels analyse.

Den medfølgende CD indeholder: Mødeindkaldelser, referater fra vejledermøder, arbejdsplaner, dokumentation for gruppens arbejdsproces, anvendte øvelser, projektgrundlag, kildekode med tilhørende dokumentation, brugervejledning til afprøvning af kildekode, samt anvendte freeware programmer.

God fornøjelse!

Agge Skov Larsen

Alexey Bessonov

Frantz Furrer

Jakob Witte Larsen

² Inceptions-, case- og elaborationsiterationsdokument med dertilhørende kildekode forefindes findes på cd'en.

2 Projektbeskrivelse

I dette kapitel vil projektoplægget blive bearbejdet. Der udarbejdes en problemanalyse, problemformulering, samt gives et overblik over gruppens mål og forventede resultater til projektet. Disse tre elementer vil danne grundlaget for den videre planlægning af softwareudviklingsforløbet.

2.1 Problemanalyse

Det udleverede projektoplæg, indeholder et inceptionsdokument og en elaborationsiteration hvor de to brugsmønstre ”Ansøgning om medlemskab (B3)” og ”Reservation af delebil (B4)” bearbejdes. Projektoplægget har på forhånd fastlagt nogle rammer for projektet, hvilket har indflydelse på de elementer, der skal analyseres.

Fra projektoplægget er givet at IT-systemet skal tilbyde følgende services³:

- Systemet skal bruges af klubbens medlemmer, klubbens formand og klubbens administrator.
- Systemet skal omhandle medlemmer med forskellige medlemstyper, biler i forskellige prisgrupper samt de aftaler, der indgås mellem medlemmerne og delebilklubben om benyttelse af bilerne.
- Systemet skal være et værktøj, som skal benyttes til at håndtere det enkelte medlems reservation og brug af debilerne og til at administrere medlemmerne og bilparken.
- Systemet skal endvidere bruges til at støtte formandens forskellige analyser af klubbens debilsordning.
- Systemet skal muliggøre registrering af afhentning eller aflevering af en delebil, ved hjælp af en telefon.

Det fremgår derudover af projektoplægget, at der skal foretages en eller flere elaborationsiterationer, hvor nogle udvalgte af de brugsmønstre, som er beskrevet i inceptionsdokumentet, skal bearbejdes.

2.2 Problemformulering

Midtby Delebilklub ønsker at aflaste administratoren og give en bedre service overfor medlemmerne, blandt andet ved at tilbyde services via telefonen. Gruppen vil i projektperioden bestræbe sig på at afklare følgende:

Er det muligt gennem bearbejdning af centrale brugsmønstre for IT-systemet, at udarbejde analyse- og designmodeller med dertilhørende implementering, der viser Midtby Delebilklub at det er muligt at udvikle et IT-system der giver en aflastning af administratoren og en bedre service overfor medlemmerne?

2.3 Mål og forventede resultater

På baggrund af problemanalysen og problemformulering er der opstillet en række mål for projektforsløbet. Målene for projektforsløbet er at:

- Udarbejde detaljerede beskrivelser af centrale brugsmønstre med udgangspunkt i de fire primære aktørers tilgang til systemet. De fire primære aktører til IT-systemet er: potentielt medlem, medlem, administrator og formand.
- Validere tilstødende brugsmønstre og krav der har indflydelse på IT-systemet ved afslutningen af hver iteration, og foretage en overvejelse om hvorvidt det er nødvendigt at realisere disse i efterfølgende iterationer.

³ Inceptionsdokument.

- Foretage eventuelle interviews vedrørende brugsmønstre og krav, for at klarlægge Midtby Delebilklubs ønsker.
- Tage højde for at en grafisk brugergrænseflade senere kan udvikles til IT-systemet.
- Udarbejde en analysemodel over de centrale brugsmønstre.
- Udarbejde en designmodel der viser den overordnede struktur for IT-systemet.
- Undersøge princippet for anvendelse af en telefon til registrering af afhentning eller aflevering af bil. Det påpeges at det er et krav fra projektoplægget, at dette element skal behandles og implementeres.
- Implementere den overordnede struktur, som berører de fire aktørers tilgang og indflydelse på IT-systemet.
- Udarbejde dokumentation så det er muligt at forsætte med udviklingen af softwarearkitekturen.

For at kunne opfylde de nævnte mål er det nødvendigt at strukturere udviklingen af IT-systemet. Struktureringen af projektet som vil danne grundlag for det videre projektarbejde, vil blive foretaget i næste kapitel.

Den dokumentation der udarbejdes for projektet, forefindes i rapportens resultatdel som starter på side 10.

3 Procesmodel

I dette kapitel beskrives den anvendte metode der benyttes i softwareudviklingsprojektet og dennes fordele og ulemper. Herefter vil der blive redegjort for prioritering og udvælgelse af brugsmønstre. Afslutningsvis vil den foreløbige plan for iterationerne blive præsenteret.

3.1 Metode

Procesmodellen der anvendes under dette softwareudviklingsprojekt er Unified Proces (UP) og dokumentationen vil blive uarbejdet ved brug af UML 2.0 (Unified Modeling Language), som er en verdensstandard til beskrivelse af systemer. IT-systemet vil blive udviklet i Java da der er blevet undervist i Java og da den udleverede kildekode er skrevet i Java. Kildekoden vil blive dokumenteret i Javadoc.

Overordnet består UP af fire faser: *Inception*, *elaboration*, *construction* og *transition*, og de enkelte faser deles op i en række iterationer.

Det er naturligt at anvende UP i projektet, da der for det første undervises i denne metode på 3. semester i systemudvikling, og da der i forbindelse med projektoplægget er udleveret en indledende iteration på IT-systemet. Hermed er det muligt at foretage yderligere elaborationsiterationer, som behandler udvalgte brugsmønstre og krav, for at opfylde målsætningen med projektet.

Generelt har UP en del fordele i forbindelse med softwareudvikling, disse fordele er:

- Det er muligt at dele projektet op i miniprojekter, hvilket muliggør at udviklerne gradvist kan sætte sig ind i forskellige faglige områder, og på den måde vil det være muligt at få tidlig og vedvarende feedback fra kunden.
- Der opnås en fungerende prototype af IT-systemet, der kan besvare problemformuleringen selvom systemet ikke bliver færdigudviklet.
- UP er særdeles velegnet til at skabe en softwarearkitektur som er let at videreudvikle og vedligeholde.

Samtidig er der også nogle ulemper ved UP, disse ulemper er:

- Det er en ny metode for gruppen, som dermed skal indlæres og derfor forløber processen langsommere.
- Det er svært at udforme en endelig kontrakt med en virksomhed, da der kan støde nye krav til eller være krav som det ikke er muligt at udforme efter virksomhedens ønsker.

Der vil blive draget nytte af fordelene ved UP under projektførelsen, hvilket betyder at det IT-system samt den dokumentation som udarbejdes, vil være let at videreudvikle, således at en gruppe systemudviklere vil kunne færdiggøre IT-systemet.

Samtidig er gruppen selvfølgelig også opmærksom på ulemperne ved processen og erkender at processen i starten vil forløbe langsommere i kraft af at metoden er ny for gruppens medlemmer.

3.2 Udvalgelse af brugsmønstre og krav

For at få klarlagt prioriteten af de enkelte brugsmønstre i forbindelse med det samlede IT-system, har gruppen valgt at vurdere brugsmønstrene ud fra deres arkitekturmæssige signifikans, den kompetenceudvikling, samt den risiko de indbefatter. Med risiko forstås kompleksiteten af de metoder som brugsmønstret kræver implementeret, samt hvorvidt implementeringen kræver undersøgelser af ny teknologi. Det vil sige at jo højere et brugsmønster er prioriteret, jo højere betydning har brugsmønstret i det samlede system.

For at sikre at problemformuleringen opfyldes, bliver der derudover vurderet vigtigheden af brugsmønstrene i forhold til målene med projektet, hvilket er at beskrive den overordnede struktur i IT-systemet og at illustrere de fire primære aktørers tilgangsvinkel til systemet. I denne vurdering er den overordnede prioritering af brugsmønstrene taget i betragtning, og det er herudfra vurderet hvilke brugsmønstre det er nødvendigt at implementere.

Vurderingen af brugsmønstrenes prioritet og vigtighed i forhold til mål og forventede resultater er vist i tabel 3.1, herunder:

Prioritet	ID	Brugsmønster	Vigtighed i forhold til mål og forventede resultater ⁴
1	B4	Reservation af delebil	Must
2	B8	Aflevering af delebil	Must
3	B7	Afhentning af delebil	Could
4	B17	Analyse af bilklubbens drift	Must
5	B16	Opgørelse af bilforbrug	Could
6	B12	Administrer medlemmer	Could
7	B3	Ansøgning om medlemskab	Must
8	B14	Vedligeholdelse af bil	Should
9	B6	Annullering af reservation	Should
10	B5	Forlængelse af reservation	Could
11	B11	Udmeldelse	Could
12	B1	Adgangskontrol	Want
13	B10a	Ændring af medlemsoplysninger	Want
14	B13	Anskaffelse af bil	Could
15	B15	Afskaffelse af bil	Want
16	B9	Indmeldelse af ekstra husstandsmedlem	Want
17	B10b	Skift af medlemskabstype	Could
18	B2	Præsentation af debilklub	Want

Tabel 3.1: Prioritering og vigtighed i forhold til gruppens mål med projektet for de enkelte brugsmønstre.

I vurderingen af vigtigheden er der taget højde for at visse brugsmønstre overlapper hinanden, både strukturelt, risikomæssigt og kompetencemæssigt, og såfremt det ene brugsmønster implementeres, falder vigtigheden af det andet. Dette gælder især for de to brugsmønstre B7 og B8, hvor det er vurderet at kun det ene af de to der har høj vigtighed i forbindelse med målene med projektet.

Ud fra tabel 3.1 kan det ses at gruppen har tildelt seks brugsmønstre de to højeste vigtigheder, inklusiv de to brugsmønstre B4 og B3, som den udleverede iteration 1 behandler. Disse seks brugsmønstre vil derfor blive behandlet i projektperioden. Herudover medtages to sekundære brugsmønstre, henholdsvis B5 og B16, som kan behandles hvis tiden tillader det. Disse to brugsmønstre er udvalgt på baggrund af deres prioritet.

ID	Brugsmønster	Prioritet
B4	Reservation af delebil	Must
B8	Aflevering af delebil	Must
B17	Analyse af bilklubbens drift	Must
B3	Ansøgning om medlemskab	Must
B14	Vedligeholdelse af bil	Should
B6	Annullering af reservation	Should
B16	Opgørelse af bilforbrug	Could
B5	Forlængelse af reservation	Could

Tabel 3.2: Udvalgte brugsmønstre med prioritering.

Brugsmønstrene i tabel 3.2 vil senere blive fordelt på de iterationer der foretages. Det skal påpeges at brugsmønster ”Afhentning af delebil (B7)” eller ”Aflevering af delebil (B8)” skal behandles ifølge projektoplægget, da det indbefatter registrering af afhentning og aflevering af delebil via telefon. Der arbejdes med B8, da den er mere omfangsrig og dermed mere vigtig for systemet.

Udover brugsmønstre i inceptionsdokumentet, foreligger der også krav, som vil få tildelt en prioritet. Denne prioritering vil blive gennemgået ved de enkelte brugsmønstre. På den måde vil det være muligt at koncentrere sig om at udvikle IT-systemet på de områder som er centrale og vigtige for at kunne vise Midtby Delebilklub, at det er muligt at udvikle et system der giver en aflastning af administratoren og en bedre service overfor medlemmerne.

⁴ Vigtigheden er prioriteret efter MoSCoW metoden.

I aktørbeskrivelsen herunder er der en kort beskrivelse af de fire primære aktører og hvilke mål aktørerne har i forhold til systemet. Målene er begrænset i forhold til de udvalgte brugsmønstre.

Primære aktører	Beskrivelse	Brugsmønstre	Mål
Potentielt medlem	Offentligheden skal have mulighed for at sende medlemskabsansøgning.	”Ansøgning om medlemskab (B3)”	- at ansøge om medlemskab i delebilklubben
Medlem	Delebilklubbens medlemmer skal have mulighed for at reservere delebil, annullere reservation og forlænge reservation.	”Reservation af delebil (B4)” ”Aflevering af delebil (B8)” ”Annullering af reservation (B6)”	- at reservere debiler - at afslutte en reservation - at annullere en reservation
Administrator	Delebilklubbens administrator skal godkende ansøgninger, ændring af medlemskabstype, udmelding af delebilklubben og foretage vedligeholdelses reservationer af debilerne.	”Vedligeholdelse af bil (B14)”	- at reservere en bil til vedligeholdelse eller rengøring
Formand	Delebilklubbens formand skal bruge systemet til at lave analyse af delebilklubbens drift.	”Analyse af bilklubbens drift (B17)”	- at analysere delebilklubbens drift

Tabel 3.3: Aktørbeskrivelse af de primære aktører samt de brugsmønstre de tilhører.

Som det fremgår af aktørbeskrivelsen er det muligt, med de udvalgte brugsmønstre, at illustrere de fire primære aktørers tilgangsvinkel til systemet og dermed opfylde målene med projektet. Det ses desuden at der er udvalgt tre brugsmønstre som omhandler medlemmer af delebilklubben. Dette skyldes at medlemmet er den aktør, der har brug for flest funktionaliteter i systemet, da det er medlemmerne som ved selvbetjening skal aflaste administratoren.

3.3 Overordnet plan for iterationer

Der er udleveret en elaborationsiteration over to brugsmønstre, og det videre projektforsløb vil beskæftige sig med iterationer i elaborationsfasen. Der vil blive foretaget yderligere to iterationer på de otte udvalgte brugsmønstre, og de højst prioriterede brugsmønstre vil blive fordelt ligeligt på de to iterationer, så det sikres at disse implementeres.

3.3.1 Plan for iterationer

Da projektperioden efter færdiggørelsen af projektgrundlaget strækker sig over i alt 8 uger, er det blevet besluttet at afsætte 3 uger til hver iteration, så en overordnet tidsplan er:

Uge	42	43	44	45	46	47	48	49	50	51
Aktivitet	Udarbejdelse af projektgrundlag		Iteration 2			Iteration 3			Projektafslutning og rapportretning.	

Tabel 3.4: Overordnet plan for iterationer.

Der følger nu en beskrivelse af iterationerne, som giver et overblik over hvad der skal foretages i de to iterationer samt hvad der foreligger fra den udleverede iteration.

3.3.2 Iteration 1 (udleverede iteration)

Den udleverede iteration 1 er den første elaborationsiteration der er udarbejdet på baggrund af det inceptionsdokument der foreligger.

I iteration 1 er de to brugsmønstre ”Ansøgning om medlemskab (B3)” og ”Reservation af delebil (B4)” blevet bearbejdet. På nuværende tidspunkt foreligger der følgende:

- Brugsmønstrebeskrivelsen for B3 er udarbejdet og den indeholder pt. solskinssceneriet.
- Der er foretaget en overfladisk analyse og design samt implementering af de vigtigste metoder for brugsmønstre B4.

Da den dokumentation der foreligger fra den udleverede iteration kun er nogle arbejdsrapporter som mangler forklaringer, vil der naturligvis være en viderebearbejdning af den udleverede iteration for at kunne opnå en fyldestgørende dokumentation.

3.3.3 Iteration 2

I iteration 2 fortsættes det videre arbejde på baggrund af inceptionsdokumentet og de arbejdsrapporter der er blevet udarbejdet i iteration 1.

I første halvdel af iteration 2 behandles brugsmønstrene: "Ansøgning om medlemskab (B3)" og "Reservation af delebil (B4)" og på den måde arbejdes videre på de arbejdsrapporter der foreligger og dokumentationen hertil færdiggøres.

Dermed kan der skabes et overblik over hvad der foreligger fra iteration 1, da det er nødvendigt for at kunne arbejde videre med udviklingen af IT-systemet og dermed bliver det også lettere at fortsætte med implementeringen senere.

For at opnå dette fortages følgende:

- En analyse og et design af brugsmønster B3, og en implementering.
- En udvidet analyse og design af brugsmønster B4, og yderligere avancerede metoder implementeres.

I anden halvdel af iteration 2 behandles brugsmønsteret: "Vedligeholdelse af bil (B14)" da dette brugsmønster er nært beslægtet med "Reservation af delebil (B4)" og hvis tiden tillader det behandles brugsmønsteret "Forlængelse af reservation (B5)".

For at opnå dette fortages følgende:

- En analyse og et design af brugsmønster B14, og implementere af de avancerede metoder.
- Såfremt tiden tillader det analyseres, designes og implementeres de overordnede metoder i brugsmønster B5.

3.3.4 Iteration 3

I iteration 3 fortsættes det videre arbejde på baggrund af inceptionsdokumentet. Følgende brugsmønstre er placeret i iteration 3: "Annullering af reservation (B6)", "Afl levering af delebil (B8)", "Analyse af bilklubbens drift (B17)" og "Opgørelse af bilforbrug (B16)".

I denne iteration vil brugsmønstrene B6, B8 og B17 blive bearbejdet samtidig og B16 påbegyndes til sidst i iterationen, såfremt tiden tillader det.

Der vil blive fortaget følgende i iterationen:

- Brugsmønster B8 vil blive analyseret, designet og implementeret i så stort et omfang, at registrering af afhentning af debilen via telefon er illustreret fuldt ud og at det er muligt at teste funktionaliteten.
- Der vil blive foretaget analyse, design og implementering af alle scenarier for brugsmønster B6 således, at det er illustreret hvordan medlemmer pålægges et gebyr såfremt de annullerer en reservation for sent.
- Der vil foreligge en overordnet analyse og design af brugsmønster B17, og hovedfunktionalitet i brugsmønsteret vil blive implementeret, således at fremgangsmåden for analyse af bilklubbens drift kan illustreres for formanden.
- Såfremt tiden tillader det, vil B16 blive analyseret, designet og implementeret.

Iterationernes aktiviteter er planlagt, som det fremgår i bilag 1.

RESULTATER

I denne del af rapporten vil resultaterne for de seks bearbejdede brugsmønstre blive præsenteret. Resultatdelen af rapporten vil være inddelt i fem kapitler: krav, analyse, design, implementering og resterende arbejde. Kapitlerne krav, analyse, design og resterende arbejde vil være opdelt i afsnit efter brugsmønstrene. Implementering vil være en samlet gennemgang af de avancerede metoder fra brugsmønstrene.

Krav, brugsmønsterbeskrivelser og til dels analysen vil primært være til kunden. Krav og brugsmønstre skal sikre at de ønsker som kunden har, er blevet tolket korrekt. Analyse, design og implementering skal bruges som dokumentation for beslutninger der er foretaget, og som grundlag for en videreudvikling af IT-systemet, ud fra det arbejde der er foretaget i dette projekt.

Både analyseklasse- og designklassediagrammet samt designinteraktionsdiagrammerne for brugsmønstrene vil være vedlagt som fold-ud diagrammer i bilag.

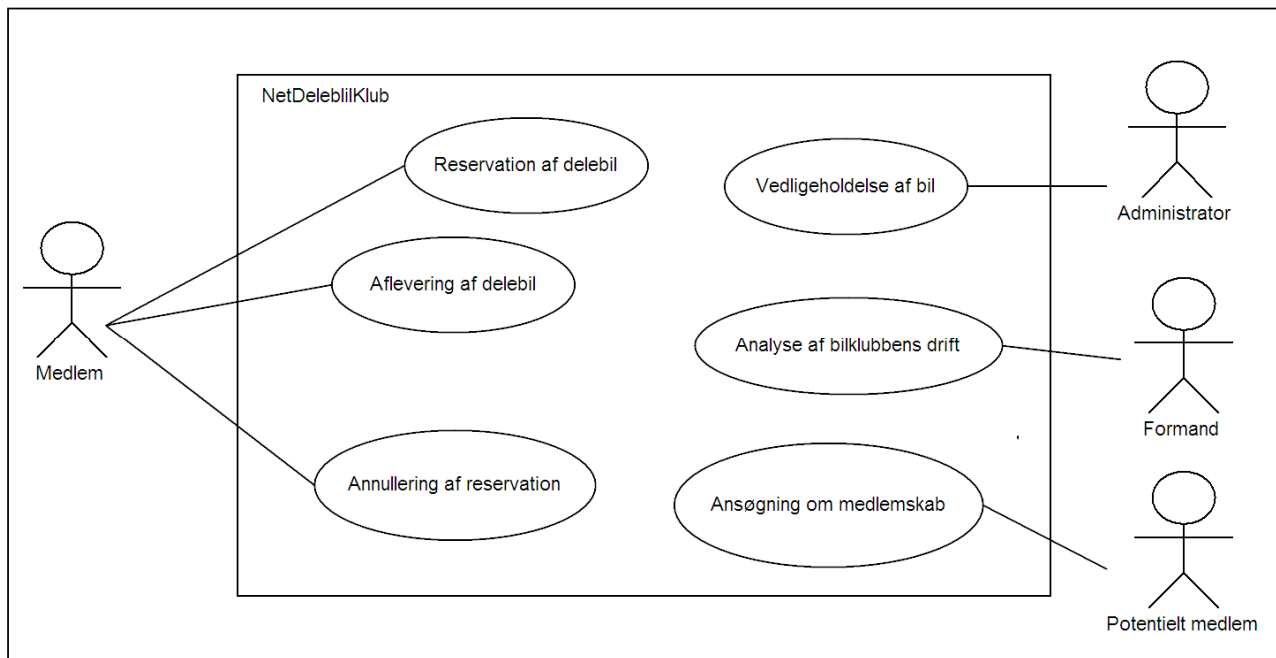
4 Krav

I dette kapitel vil der blive vist en brugsmønstermodel der illustrerer de brugsmønstre der bliver behandlet. Der vil desuden blive opstillet en kravmodel, der beskriver de krav som de enkelte brugsmønstre dækker over. Herefter vil der være en prioritering af de tilhørende krav, samt en brugsmønsterbeskrivelse for hvert af de behandlede brugsmønstre.

4.1 Brugsmønstermodel

Brugsmønstermodellen illustrerer de behandlede brugsmønstre samt de berørte aktører og deres tilgang til systemet. Alle de illustrerede aktører er primære, da de valgte brugsmønstre ikke involverer sekundære aktører. Det bemærkes desuden at alle de fire primære aktører til IT-systemet er repræsenteret, og det udviklede IT-system vil derfor dække over dem alle.

Brugsmønstermodellen for den del af systemet som behandles i dokumentationen er vist herunder:



Figur 4.1: Brugsmønstermodel for de behandlede brugsmønstre i IT-systemet.

Som det fremgår af brugsmønstermodellen er der i projektet kun bearbejdet de seks primære brugsmønstre, da der ikke har været tilstrækkelig tid til at behandle de to sekundære brugsmønstre: ”Forlængelse af reservation (B5)” og ”Opgørelse af forbrug (B16)”.

4.2 Kravmodel

I kravmodellen herunder ses de krav som de udvalgte brugsmønstre berører. I kravmodellen er kravene vist ud for de brugsmønstre, som de tilhører eller har indflydelse på og derfor vil nogle af kravene optræde flere gange.

De krav der tilhører et brugsmønster vil blive prioriteret når det enkelte brugsmønster behandles i afsnit 4.3 – 4.8, og der vil blive taget højde for de krav der har indflydelse på brugsmønsteret under bearbejdningen af de enkelte brugsmønsterbeskrivelser. De krav der kun skal tages højde for i brugsmønstrene er i kravmodellen markeret med *kursiv*. Kravene har et ID, hvor *F* er et funktionelt krav og *K* er et ikke-funktionelt krav.

Kravmodellen er udarbejdet på baggrund af inceptionsdokumentet:

Brugs-mønster	ID	Type	Detaljer	Særlige forhold	Kilde
B3	F01	Medlemskab	NDB skal gøre det muligt for en ansøger at anmode om medlemskab af delebilklubben.	Ansøger skal være 21 år og have haft kørekort i mindst et år.	Case
	<i>F02</i>	<i>Medlemskab</i>	<i>NDB skal gøre det muligt for delebilklubbens administrator at acceptere eller afvise en ansøgers anmodning om medlemskab.</i>	<i>Ansøger skal være 21 år og have haft kørekort i mindst et år.</i>	<i>Case</i>
	F03	Medlemskab	NDB skal ved indmeldelse sikre at de gældende betingelser for medlemskab er opfyldt, såsom alderskrav ol.	Ansøger skal være 21 år og have haft kørekort i mindst et år	Case
	F04	Medlemskab	NDB skal tilbyde forskellige former for medlemskaber.	Pt. A- B- og C-medlemskab. Nærmere oplysninger fremgår af nuværende prisliste.	Case
	<i>F05</i>	<i>Medlemmer</i>	<i>NDB skal gøre det muligt at indmelde ekstra husstandsmedlemmer</i>		<i>Case</i>
	F19	Medlemskab	NDB skal registrere oplysninger svarende til de oplysninger der fremgår af nuværende indmeldelsesblanket.		Case
B4	F08	Bestilling	NDB skal gøre det muligt at se bilmærke, udleveringssted (evt. med tlf. nr.), og tilgængelighed for klubbens biler.		Case
	F09	Bestilling	NDB skal gøre det muligt at bestille en bil så ofte og så længe et medlem ønsker indenfor en nærmere bestemt maksimal periode.	P.t. max. et år frem i tiden	Case
	F14	Bestilling	NDB skal gemme informationer om medlemsnummer, bil, biltype, reservationsperiode og udleveringssted i forbindelse med en reservation.		Case
	<i>F15</i>	<i>Rengøring og vedligeholdelse</i>	<i>NDB skal gøre det muligt for administratoren at reservere en bil til rengøring og vedligeholdelse.</i>		<i>Case</i>
	F21	Biler	NDB skal registrere oplysninger om biler svarende til de oplysninger der fremgår af nuværende prisliste		Case
	<i>F23</i>	<i>Bestilling</i>	<i>NDB skal gøre det muligt at bestille biler hos tredjepart.</i>		<i>Case</i>
	K01	Brugbarhed (tilgængelighed)	NDB skal gøre det muligt for medlemmer at reservere delebiler døgnet rundt.		Case
B6	F11	Bestilling	NDB skal gøre det muligt at annullere en reservationsperiode op til den maksimale annulleringstid før bilen skulle have været brugt.	Maksimal annulleringstid er 2 timer	Case

B6	F12	Bestilling	NDB skal pålægge medlemmet et gebyr, hvis en reservation annulleres senere end den maksimale annulleringstid. Hvis starttidspunktet for den reservation som medlemmet ønsker at annullere er overskredet gælder samme regler som, hvis medlemmet annullerede inden reservationens start.		Case Interview
	F13	Bestilling	NDB skal gøre det muligt for et medlem at aflyse en reservationsperiode uden gebyr, hvis bilen ikke er på udleveringsstedet ved afhentning.		Case
	F22	Gebyrer	<i>NDB skal registrere oplysninger og gebyrer svarende til de oplysninger der fremgår af nuværende prisliste</i>		Case
B8	F16	Aflevering	NDB skal registrere antal kørte kilometer automatisk ved aflevering af bil	Kunden har ikke fastlagt hvilken teknologi kravet skal løses med.	Case
	F17	Aflevering	<i>NDB skal beregne ekstra gebyr, timetakst og kilometer-takst ved for sen aflevering af bil</i>		Case
	F27	Aflevering	<i>NDB skal pålægge gebyr for overskydende registreringsperiode ved for tidligt aflevering.</i>	<i>Hvis en bil afleveres mere end 24 timer for tidligt</i>	Interview ⁵
	K05	Brugbarhed (tilgængelighed)	NDB skal gøre det muligt at registrere aflevering af bil vha. en telefon.	I første omgang via en mobilsimulering (forsøgsopstilling med telefontastatur)	Møde1
B14	F15	Rengøring og vedligeholdelse	NDB skal gøre det muligt for administratoren at reservere en bil til rengøring og vedligeholdelse.		Case
	F24	Rengøring og vedligeholdelse	NDB skal oplyse berørte medlemmer såfremt deres reservation annulleres af administrator.		Interview
	F25	Rengøring og vedligeholdelse	Hvis et medlems reservation annulleres af administrator, skal NDB tilbyde medlemmet en ny bil i den pågældende reservationsperiode. Hvis der ikke er nogen biler til rådighed hos delebilklubben, tilbydes en bil fra tredjepart.	Hvis der ikke er en anden bil af samme biltype til rådighed i perioden, kan medlemmet frit vælge mellem bilparkens ledige biler. Prisen for medlemmet forbliver dog den samme.	Interview
	F26	Rengøring og vedligeholdelse	NDB skal registrere hvornår bilerne sidst er blevet rengjort og har været til service.		Interview
B17	F28	Analyse	NDB skal gøre det muligt for formanden at få vist bilernes kørte km.		Interview
	F29	Analyse	NDB skal gøre det muligt for formanden at få vist bilernes udlejningstid i en given periode.		Interview
	F30	Analyse	NDB skal gøre det muligt for formanden at få vist antallet af tilkommende medlemmer i en periode.		Interview
	F31	Analyse	NDB skal gøre det muligt for formanden at få vist reservationer pr. medlem i en periode.		Interview
	F32	Analyse	NDB skal gøre det muligt for formanden at få vist ulykker pr. år.		Interview

Tabel 4.1: Beskrivelse af de krav som de udvalgte brugsmønstre dækker over.

⁵ Krav udarbejdet ud fra interview med repræsentant for delebilklubben, i dette tilfælde gruppens vejleder, i iteration 3.

4.3 ”Ansøgning om medlemskab (B3)”

For brugsmønster B3 illustreres solskinsscenariet, for at vise de overordnede funktionaliteter brugsmønsteret indeholder.

Nedenstående tabel indeholder en beskrivelse af de krav der tilhører brugsmønsteret, samt hvordan disse krav er prioriteret:

Krav ID	Detaljer	Særlige forhold	Prioritet
F01	NDB skal gøre det muligt for en ansøger at anmode om medlemskab af delebilklubben.	Ansøger skal være 21 år og have haft kørekort i mindst et år.	Must
F02	<i>NDB skal gøre det muligt for delebilklubbens administrator at acceptere eller afvise en ansøgers anmodning om medlemskab.</i>	<i>Ansøger skal være 21 år og have haft kørekort i mindst et år.</i>	-
F03	NDB skal ved indmeldelse sikre at de gældende betingelser for medlemskab er opfyldt, såsom alderskrav ol.	Ansøger skal være 21 år og have haft kørekort i mindst et år	Could
F04	NDB skal tilbyde forskellige former for medlemskaber.	Pt. A- B- og C-medlemskab. Nærmere oplysninger fremgår af nuværende prisliste.	Must
F05	<i>NDB skal gøre det muligt at indmelde ekstra husstandsmedlemmer</i>		-
F19	NDB skal registrere oplysninger om medlemmer svarende til de oplysninger der fremgår af nuværende indmeldelsesblanket.		Want

Tabel 4.2: Beskrivelse af de krav som brugsmønsteret dækker over samt prioriteten af de enkelte krav.

Som det ses i tabellen er F01 og F04 prioriteret højest, da disse krav beskriver hovedfunktionaliteten af hvordan ansøgningen skal fungere. F02 og F05 er ikke prioritet fordi de ikke direkte tilhører brugsmønsteret, men beskriver noget der skal tages hensyn til i behandlingen af dette brugsmønster. F19 er prioriteret lavt, da de fleste indmeldelsesoplysninger ikke er nødvendige ved oprettelse af et nyt medlem på nuværende tidspunkt, de er derfor ikke nødvendige for at vise hovedfunktionaliteten i brugsmønsteret.

F03 er prioriteret lidt lavere end F01 og F04, da dette krav ikke er nødvendig for at illustrere proceduren for ansøgning om medlemskab. Der tages i den videre analyse og design, på nuværende tidspunkt ikke højde for kontrol af indtastningsfejl og kontrol af hvorvidt personer er gamle nok til at ansøge om medlemskab af delebilklubben.

4.3.1 Brugsmønsterbeskrivelse for B3⁶

ID:	B3
Formål:	At give et potentielt medlem mulighed for at indsende en ansøgning om medlemskab af delebilklubben.
Oversigt:	Når en person ønsker at blive medlem af delebilklubben, indsender personen en ansøgning om medlemskab ved at udfylde en formular på klubbens hjemmeside. I ansøgningen anføres en række personlige oplysninger. Ansøgningen viderebehandles af administratoren i brugsmønstret "Administrer Medlemmer (B12)".
Primære aktører:	Potentielt medlem.
Sekundære aktører:	Ingen.
Interesserter:	<i>Potentielt medlem:</i> Ønsker at få mulighed for at angive de nødvendige oplysninger på en enkel måde og at få behandlet sin ansøgning hurtigt og effektivt. <i>Delebilklub:</i> Ønsker på en nem måde at få alle nødvendige oplysninger om et potentielt medlem der ansøger om medlemskab.
Prækondition:	Ingen.
Postkondition:	Delebilklubben har modtaget det potentielle medlems ansøgning og har kvitteret for denne.
Krydsreferencer:	F01, F02, F03, F04, F05, F19.
Datakrav:	Indmeldelsesoplysninger fremgår af den nuværende indmeldelsesblanket ⁷ .
Udvidelser:	Brugerautorisationen og autentifikationsprocedure i brugsmønstret "Adgangskontrol (B1)".
Typisk hændelsesforløb (Hovedsucces-scenariet)	
Aktør	System
1. Et potentielt medlem vælger at ansøge om medlemskab af delebilklubben.	2. Systemet adspørger aktøren om indmeldelsesoplysninger.
3. Det potentielle medlem afgiver indmeldelsesoplysninger.	4. Systemet kontrollerer indmeldelsesoplysningerne for fejl og mangler.
	5. Systemet fortæller det potentielle medlem om delebilklubbens godkendelsesprocedure og spørger om ansøgning ønskes afsendt.
6. Det potentielle medlem accepterer	7. Systemet kvitterer for at have modtaget ansøgning og indmeldelsesoplysningerne.
Alternative forløb:	
Ingen	

⁶ Brugsmønstret er udvidet ud fra det udleverede brugsmønster fra iteration 1.

⁷ Projektoplægget (Casedokumentet)

4.4 ”Reservation af delebil (B4)”

Der arbejdes med solskinsscenariet for brugsmønster B4, for at vise proceduren for reservation af debiler overfor debilklubben.

I tabellen herunder er angivet en beskrivelse af de krav der tilhører brugsmønsteret, og der er foretaget en prioritering af disse:

Krav ID	Detaljer	Særlige forhold	Prioritet
F08	NDB skal gøre det muligt at se bilmærke, udleveringssted (evt. med tlf. nr.) og tilgængelighed for klubbens biler.		Must
F09	NDB skal gøre det muligt at bestille en bil så ofte og så længe et medlem ønsker indenfor en nærmere bestemt maksimal periode.	P.t. max. et år frem i tiden	Should
F14	NDB skal gemme informationer om medlemsnummer, bil, biltype, reservationsperiode og udleveringssted i forbindelse med en reservation.		Must
F15	<i>NDB skal gøre det muligt for administratoren at reservere en bil til rengøring og vedligeholdelse.</i>		-
F21	NDB skal registrere oplysninger om biler svarende til de oplysninger der fremgår af nuværende prisliste		Should
F23	NDB skal gøre det muligt at bestille biler hos tredjepart.		Want
K01	NDB skal gøre det muligt for medlemmer at reservere debiler døgnet rundt.		Want

Tabel 4.3: Beskrivelse af de krav som brugsmønsteret dækker over samt prioriteten af de enkelte krav.

Da det centrale i brugsmønsteret er at systemet skal opfylde F14, er dette krav prioriteret højt. I iteration 1 er der foretaget en bearbejdning af brugsmønsteret, hvor F14 er blevet implementeret. Derfor behandles F08 i iteration 2, da dette krav illustrerer en væsentlig del af proceduren for foretagelsen af reservationer. Kravet F15 vil blive behandlet i et nyt brugsmønster, da det udgør et alternativt forløb til B4, og kravet er derfor ikke prioriteret ved dette brugsmønster.

På næste side forefindes dette brugsmønsters brugsmønsterbeskrivelse. Punkt 3 og 4 i hovedhændelsesscenariet er beskrevet på baggrund af den mockup for oplistningen af biler, som illustreret i bilag 2. Det er vurderet, at for at kunne tilfredsstille kundens ønsker angående oplistningen af biler, er det nødvendigt i første omgang at få oplistet alle bilerne, med deres tilhørende biltype, udleveringssted samt med de reservationer, der er foretaget på bilen. Herefter er det så muligt for medlemmet at indskrænke oplistningen ved at angive specifik biltype, udleveringssted eller periode. Hvis biltype og udleveringssted matcher den angivne indskrænkning, oplistes bilen sammen med de reservationer der er foretaget på bilen, såfremt der foreligger nogle reservationer.

Det er desuden blevet vurderet, at det kun skal være muligt at angive én biltype eller ét udleveringssted idet oplistningen gentages.

4.4.1 Brugsmønsterbeskrivelse for B4⁸

ID:	B4
Formål:	At reservere en bil fra et bestemt udleveringssted i en bestemt periode.
Oversigt:	<p>Et medlem ønsker at reservere en bil. Medlemmet får mulighed for at se hvilke biler, der er ledige på de forskellige udleveringssteder i forskellige perioder.</p> <p>En bil kan reserveres så længe man ønsker, til max. et år frem i tiden fra den dato reservationen foretages.</p> <p>Medlemmet kan eventuelt udføre flere reservationer i samme arbejdsgang.</p> <p>Når medlemmet har truffet sine valg, registreres reservationen eller reservationerne i systemet med den valgte bil, det tilhørende udleveringssted og den valgte periode.</p>
Primære aktører:	Medlem
Sekundære aktører:	Ingen
Interesser:	<p><i>Medlem:</i> Ønsker at kunne få et overblik over ledige biler. Medlemmet ønsker desuden døgnnet rundt, hjemmefra, at kunne foretage en reservation af en bestemt bil.</p> <p><i>Administrator:</i> Ønsker at kunne foretage en reservation af en bil, på vegne af et medlem.</p> <p><i>Delebilklub:</i> Ønsker at begrænse det administrative arbejde, ved at lade medlemmer foretage reservationer på egen hånd.</p>
Prækondition:	Medlem har fået adgang til at benytte systemet (se brugsmønsteret "Adgangskontrol (B1)" i inceptionsdokumentet).
Postkondition:	Reservationen eller reservationerne er gemt
Krydsreferencer	F08, F09, F14, F15, F23, K01
Typisk hændelsesforløb (Hovedsucces-scenariet)	
Aktør	System
1. Et medlem ønsker at reservere en delebil.	2. Systemet viser de generelle muligheder der er for reservation. Alle biler oplyses med tilhørende type, udleveringssted og hvornår bilen er reserveret.
3. Hvis medlemmet ønsker at begrænse visningen af debiler, angiver medlemmet den ønskede biltype, det ønskede udleveringssted eller den ønskede periode.	4. Systemet viser de muligheder der er for reservation. De mulige biltyper, de mulige udleveringssteder, de mulige perioder indenfor de afgivne ønsker.
<i>Trin 3 og 4 springes over eller gøres så længe medlemmet ønsker det.</i>	
5. Medlemmet reserverer en bestemt bil, et bestemt udleveringssted i en bestemt periode.	6. Systemet reserverer den valgte bil, fra det valgte udleveringssted for den valgte periode og indikerer overfor medlemmet at reservationen har fundet sted, samt viser prisen for den reservede periode (ekskl. prisen for kørte km).
<i>Trin 2-6 gentager sig så længe medlemmet ønsker det.</i>	
7. Medlemmet afslutter brugsmønsteret.	
Alternative forløb:	
Vedligeholdelsesreservation	Administratoren kan reservere biler til rengøring eller reparation. Når en bil reserveres til vedligeholdelse, skal den ikke kunne reserveres af medlemmer (se brugsmønsteret "Vedligeholdelse af bil (B14)").
Bil reserveret	Hvis den ønskede bil er reserveret i den indtastede periode, giver systemet en fejlmeddelelse, og sender brugeren tilbage til punkt 5.
Reservation ved tredjepart	Hvis der ikke er flere biler til rådighed i den ønskede periode, giver systemet medlemmet mulighed for at reservere bil ved eksternt biludlejningsfirma.
Reservation på vegne af medlem	Administratoren foretager reservationer på vegne af et medlem.

⁸ Brugsmønsteret er udvidet ud fra det udleverede brugsmønster fra iteration 1.

4.5 ”Annullering af reservation (B6)”

Det er valgt at behandle alle scenarier for annullering af reservation og ikke kun solskinsscenariet, som det er tilfældet med de andre brugsmønstre. De alternative scenarier vil derfor blive indarbejdet i hændesscenariet i brugsmønsterbeskrivelsen.

I tabellen herunder er angivet prioriteringen af brugsmønsterets krav:

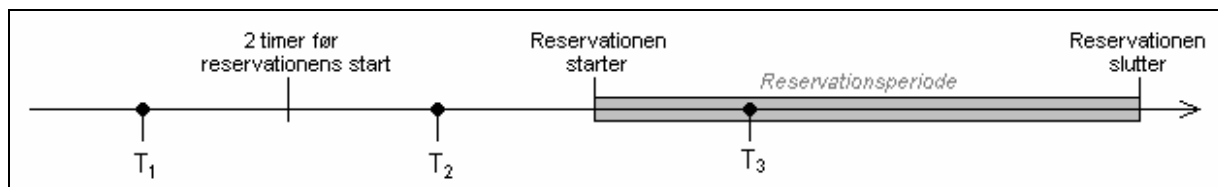
Krav ID	Detaljer	Særlige forhold	Prioritet
F11	NDB skal gøre det muligt at annullere en reservationsperiode op til den maksimale annulleringstid før bilen skulle have været brugt.	Maksimal annullerings tid er 2 timer	Should
F12	NDB skal pålægge medlemmet et gebyr, hvis en reservation annulleres senere end den maksimale annulleringstid. Hvis starttidspunktet for den reservation, som medlemmet ønsker at annullere, er overskrevet gælder samme regler som, hvis medlemmet annullerede inden reservationens start.		Must
F13	NDB skal gøre det muligt for et medlem at aflyse en reservationsperiode uden gebyr, hvis bilen ikke er på udleveringsstedet ved afhentning.		Must
F22	<i>NDB skal registrere oplysninger og gebyrer svarende til de oplysninger der fremgår af nuværende prisliste</i>		-

Tabel 4.4: Beskrivelse af de krav som brugsmønsteret dækker over samt prioriteten af de enkelte krav.

De tre krav der hører til brugsmønsteret, F11, F12 og F13, er alle prioriteret højt, da alle annulleringsscenarierne vil blive behandlet. Det er vurderet at kravene F12 og F13 er henholdsvis de vigtigste og mest omfattende at realiseret, da dette kræver kontrol af afleveringstidspunkt og tilgængelighed for bilen samt at medlemmet pålægges et gebyr. Da F12 inddrager kontrol af afleveringstiden, har kravet fået næsthøjeste prioritet.

Krav F22 er medtaget under brugsmønsteret, fordi det har indflydelse på brugsmønsteret, da det er nødvendigt at have informationer omkring gebyrer for at kunne behandle F12.

F11, F12 og F13 indbefatter tre forskellige tidspunkter hvor medlemmet kan fortage en annullering af en reservation. Disse tre annulleringstidspunkter er illustreret i figuren herunder:



Figur 4.1: Tidspunkter for annullering af reservation.

De tre annulleringstidspunkter lægger på baggrund af kravene op til fire forskellige scenarier, som skal behandles forskelligt af systemet. De fire scenarier er klarlagt i tabellen herunder:

Tidspunkt	Beskrivelse	Systemets håndtering af anmodning om annullering
T_1	Reservationen annulleres mere end 2 timer før reservationens start.	Reservationen slettes og medlemmet pålægges <u>ikke</u> gebyr.
T_2	Reservationen annulleres mindre end 2 timer før reservationens start.	Reservationen slettes og medlemmet pålægges aflysningsgebyr, på 100 kr.
T_3	Reservationen annulleres efter reservationens start og bilen har ikke været til rådighed ved reservationens start.	Reservationen slettes og medlemmet pålægges <u>ikke</u> gebyr.
T_3	Reservationen annulleres efter reservationens start og bilen har været til rådighed ved reservationens start.	Reservationen slettes og medlemmet pålægges aflysningsgebyr, på 100 kr.

Tabel 4.5: Beskrivelse af de fire scenarier der kan forekomme når et medlem vælger at annullere en reservation.

De fire scenarier og systemets håndtering heraf, er medtaget i brugsmønsterbeskrivelsen på næste side.

4.5.1 Brugsmønsterbeskrivelse for B6

ID:	B6
Formål:	At give medlemmer af delebilklubben mulighed for at annullere deres reservationer.
Oversigt:	Et medlem har foretaget en reservation der viser sig ikke at være nødvendig, og ønsker derfor at annullere denne. Medlemmet får mulighed for at se sine reservationer og kan så vælge at annullere en eller flere af disse. Når medlemmet har besluttet hvilken reservation der skal annulleres, fjernes denne fra medlemmets aktive reservationer.
Primære aktører:	Medlem
Sekundære aktører:	Ingen
Interesserter:	<i>Medlem:</i> Ønsker at kunne annullere sine reservationer. <i>Delebilklub:</i> Ønsker at overlade arbejdet med at annullere reservationer til medlemmerne for at aflaste administratoren. Delebilklubben ønsker endvidere at gebyrtilskrivning skal ske automatisk ved overtrædelse af annulleringsfristen.
Prækondition:	Medlem har fået adgang til systemet (se brugsmønster ”Adgangskontrol (B1)” i inceptionsdokumentet) og har tidligere foretaget en eller flere reservationer.
Postkondition:	Reservationen er fjernet og medlemmet er pålagt gebyr hvis annulleringsfristen overskrides.
Krydsreferencer:	F11, F12, F13, F22
Datakrav:	Gebyrer fremgår af nuværende prisliste ⁹ .
Udvidelser:	Ingen

Hændelsesforløb	
Aktør	System
1. Et medlem ønsker at annullere en reservation.	
	2. Systemet oplister medlemmets reservationer i perioden fra dags dato til et år frem i tiden.
3. Medlemmet vælger den reservation der skal annulleres.	4. Hvis annulleringen sker mere end 2 timer før reservationens start. 4.1 Systemet fjerner reservationen. 4.2 Systemet meddeler medlemmet at reservationen er fjernet.
	5. Ellers hvis annulleringen sker mindre end 2 timer før reservationens start, men før reservationsstart. 5.1 Systemet pålægger medlemmet et gebyr. 5.2 Systemet fjerner reservationen. 5.3 Systemet meddeler medlemmet at reservationen er fjernet og der er opkrævet aflysningsgebyr.
	6. Ellers hvis annulleringen sker efter reservationens start. 6.1 Hvis den pågældende bil ikke er til rådighed ved reservationens start. 6.1.1 Systemet sletter reservationen. 6.1.2 Systemet meddeler medlemmet at reservationen er slettet.
	6.2 Ellers 6.2.1 Systemet pålægger medlemmet et gebyr. 6.2.2 Systemet fjerner reservationen. 6.2.3 Systemet meddeler medlemmet at reservationen er fjernet og der er opkrævet aflysningsgebyr.
<i>Trin 2-6 gentages så længe medlemmet ønsker det.</i>	
7. Medlemmet afslutter brugsmønsteret.	

Alternative forløb:	Ingen
----------------------------	-------

⁹ Projektoplæg (Casedokument)

4.6 ”Aflevering af delebil (B8)”

For dette brugsmønster vises solskinsscenariet, da der ønskes at de centrale dele angående registrering af aflevering af delebil skal implementeres i henhold til projektoplæggets krav.

Herunder er brugsmønsterets tilhørende krav oplistet og prioriteret:

Krav ID	Detaljer	Særlige forhold	Prioritet
F16	NDB skal registrere antal kørte kilometer automatisk ved aflevering af bil	Kunden har ikke fastlagt hvilken teknologi kravet skal løses med.	Must
F17	<i>NDB skal beregne ekstra gebyr, timetakst og kilometer-takst ved for sen aflevering af bil</i>		-
F27	<i>NDB skal pålægge gebyr for overskydende registreringsperiode ved for tidligt aflevering.</i>	<i>Hvis en bil afleveres mere end et fastlagt antal timer for tidligt</i>	-
K05	NDB skal gøre det muligt at registrere aflevering af bil vha. en telefon.	I første omgang via en mobilsimulering (forsøgsopstilling med telefontastatur)	Must

Tabel 4.6: Beskrivelse af de krav som brugsmønsteret dækker over samt prioriteten af de enkelte krav.

Det er naturligt at K05 er prioriteret højest, da IT-systemet i henhold til projektoplægget skal opfylde dette krav. Da Delebilklubben ønsker at antal kørte km skal registreres automatisk og da dette krav også er en del af solskinsscenariet har F16 også fået højeste prioritet.

Der tages desuden højde for F17 og F27, men disse behandles ikke i dette brugsmønster da brugsmønsteret ellers ville blive for omfangsrigt. Der bliver i stedet for lavet to nye brugsmønstre: ”Delebil afleveret for tidligt” og ”Delebil afleveret for sent” som varetager F17 og F27.

Automatiseringsgraden i F16 er ikke fastlagt, men det er op til udvikleren at løse det i sådan en grad, at Delebilklubben kan vurdere om det er en tilstrækkelig løsning. Det er besluttet, at en høj automatiseringsgrad, hvor medlemmet er fri for nogen former for aflæsning, vil være den bedste løsning, da det giver den bedste service overfor medlemmerne. Dette løses ved at der installeres et computersystem i bilen, som aflæser de kørte kilometre ved afslutning af reservationen, og som sender data til NDB. Dette er illustreret i figuren herunder:



Figur 4.2: Overførsel af data fra delebil til IT-systemet, til registrering af aflevering.

I det videre analyse- og designforløb, er beskrivelsen af kommunikationsmetoden præget af denne beslutning, men det vil ikke have nogen indflydelse på implementeringsdelen, da NDB-systemet ikke er afhængig af om det er bilcomputeren eller medlemmet der foretager opkaldet. Derfor vil systemet i første omgang kunne testes ved hjælp af en mobilsimulering, bestående af en forsøgsopstilling med telefontastatur¹⁰.

¹⁰ Sammen med projektet blev der udleveret en fastnettelefon, som kan slutes til computerens lydindgang, og hermed kan en telefonlinje ind til systemet simuleres.

4.6.1 Brugsmønsterbeskrivelse for B8

ID:	B8	
Formål:	At registrere antallet af kørte kilometer for den reserverede periode samt registrere at anvendelsen af bilen er afsluttet og at bilen dermed atter er til rådighed.	
Oversigt:	<p>Når et medlem er færdig med at benytte en bil skal den afleveres på dens faste udleveringssted inden reservationsperiodens udløb. Antal kørte km skal registreres og nøglen hænges på plads.</p> <p>Det er et ønske fra delebilklubbens side at antal kørte km skal registreres automatisk. Derfor er det nødvendigt at udvikle et system i bilen (bilcomputeren) der kan registre antallet af kørte km og sende oplysningerne til Delebilklubbens system.</p>	
Primære aktører:	Medlem	
Sekundære aktører:	Ingen	
Interesserter:	<p><i>Medlem:</i> Ønsker en hurtig og enkel afleveringsprocedure.</p> <p><i>Delebilklub:</i> Ønsker at antal kørte km skal registreres automatisk og at afleveringstidspunktet bliver registreret, så der kan pålægges gebyr, hvis bilen afleveres for sent.</p>	
Prækondition:	En reservationsperiode er i gang, og medlemmet har indtastet medlemsnummer og reservationsnummer ved afhentning i bilcomputeren. Aktøren er ankommet til udleveringsstedet.	
Postkondition:	Medlemsnummer, reservationsnummer og antal kørte km er registreret i systemet, derefter oplyser bilcomputer medlemmet om at data er registreret.	
Krydsreferencer:	K05, F16, F17, F27	
Datakrav:	Antal kørte km skal registreres. System modtager data gennem telefonlinjen fra bilcomputeren.	
Typisk hændelsesforløb (Hovedsucces-scenariet)		
Aktør	Bilcomputer	System
1. Et medlem ønsker at afslutte en reservation ved at benytte bilens bilcomputer.	2. Bilcomputeren registrerer antal kørte kilometer og sender medlemsnummer, reservationsnummer og kørte km til systemet.	3. Gemmer kørte km og tidspunkt for aflevering i den pågældende reservation.
	5. Bilcomputer kvitterer for aflevering af delebil.	4. Systemet beregner forbrugspris og periodepris og gemmer disse værdier i systemet.
Alternative forløb:		
Delebil afleveres for tidligt	Hvis delebilen afleveres mere end et fastlagt antal timer tidligere end reservationens udløb, afkorter systemet automatisk reservationen, og der pålægges et gebyr for overbooking.	
Delebil afleveres for sent	Hvis delebilen afleveres for sent pålægges automatisk et gebyr for overskridelse af reservationsperioden.	

4.7 ”Vedligeholdelse af bil (B14)”

For at vise den overordnede funktion af brugsmønsteret, er det blevet valgt i første ombæring kun at implementere solskinsscenariet.

Prioritering af brugsmønsterets tilhørende krav er vist i tabellen herunder:

Krav ID	Detaljer	Særlige forhold	Prioritet
F15	NDB skal gøre det muligt for administratoren at reservere en bil til rengøring og vedligeholdelse, også selvom bilen er reserveret af et medlem.	Bil kan reserveres så længe det er nødvendigt.	Must
F24	NDB skal oplyse berørte medlemmer såfremt deres reservation annulleres af administrator.		Should
F25	Hvis et medlems reservation annulleres skal NDB tilbyde medlemmet en ny bil i den pågældende reservationsperiode. Hvis der ikke er nogen biler til rådighed hos delebilklubben tilbydes en bil fra tredjepart.	Hvis der ikke er en anden bil af samme biltype til rådighed i perioden, kan medlemmet frit vælge mellem bilparkens ledige biler. Prisen for medlemmet forbliver dog den samme.	Want
F26	NDB skal registrere hvornår bilerne sidst er blevet rengjort og har været til service.		Must

Tabel 4.7: Beskrivelse af de krav som brugsmønsteret dækker over samt prioriteten af de enkelte krav.

Da det er valgt at brugsmønsteret kun skal bearbejdes så det er muligt at vise solskinsscenariet, er de tilhørende krav prioriteret på baggrund af dette. Krav F24-F26 er udformet på baggrund af interview¹¹ med formanden af delebilklubben.

Det kan ses i tabellen er krav F15 og F26 prioriteret højest, fordi disse krav er essentielle for at få implementeret solskinsscenariet. F25 har fået den laveste prioritering, fordi den overordnede funktion af brugsmønsteret godt kan vises, selvom dette krav ikke er opfyldt. F24 har fået næsthøjeste prioritering da det er vigtigt at gøre det medlem, hvis reservation bliver annulleret, opmærksom på dette. Krav F24 kunne tænkes at sørge for at sende en e-mail til berørte medlemmer, når systemet er bearbejdet yderligere.

Da det kun er solskinsscenariet der vil blive bearbejdet, vil der ikke være noget tjek af oplysninger som administratoren indtaster. En mulig brugergrænseflade fremgår af bilag 3, og brugsmønsterbeskrivelsen er udarbejdet på baggrund af denne.

¹¹ Projektoplægget (Casedokumentet)

4.7.1 Brugsmønsterbeskrivelse for B14

ID:	B14
Formål:	At registrere hvornår en bil skal til serviceeftersyn og rengøring.
Oversigt:	Administratoren ønsker at reservere en bil til enten rengøring eller serviceeftersyn. Administratoren får mulighed for at se hvilke reservationer, der er fortaget på alle bilerne i bilparken samt dato for sidste serviceeftersyn og rengøring. Administratoren kan reservere bilerne så længe det er nødvendigt, og også selvom bilen er reserveret af et medlem. Når administratoren har truffet sine valg registreres reservationen i systemet med den valgte bil, det valgte udleveringssted og for den valgte periode. Systemet oplyser administratoren om hvilke medlemsreservationer hans reservation berører, så han informere de pågældende medlemmer, og ændre deres reservationer. Administratoren kan eventuelt udføre flere reservationer i samme arbejdsdag.
Primære aktører:	Administrator
Sekundære aktører:	Ingen
Interesserter:	<i>Administrator:</i> Ønsker at kunne reservere en bil til enten rengøring eller serviceeftersyn. <i>Medlem:</i> Ønsker at blive kontaktet såfremt hans reservation bliver annulleret af administratoren. <i>Delebilklub:</i> Ønsker at gøre det muligt for administratoren at reservere biler til rengøring eller service, og informere berørte medlemmer om ændringen i deres reservationer.
Prækondition:	Administratoren har fået adgang til systemet (se brugsmønsteret ”Adgangskontrol (B1)” i inceptionsdokumentet).
Postkondition:	Reservationen eller reservationerne er gemt i systemet, og administratoren har fået besked om hvilke medlemsreservationer hans vedligeholdelsesreservation berører.
Krydsreferencer:	F15, F24, F25, F26
Udvidelser:	Administratoren skal senere, efter at have været i kontakt med kunden, have mulighed for at kunne tildele de implicerede medlemsreservationer en ny bil.
Typisk hændelsesforløb (Hovedsucces-scenariet)	
Aktør	System
1. Administratoren ønsker at reservere en delebil til rengøring eller vedligeholdelse.	2. Systemet oplister alle biler i bilparken med tilhørende biltype, udleveringssted, dato for sidste rengøring, dato for sidste serviceeftersyn og hvornår bilen er reserveret.
3. Hvis administratoren ønsker at begrænse visningen af mulighederne, anfører administratoren den eller de ønskede biltyper, udleveringssteder eller perioder, samt hvorvidt han ønsker at sortere bilerne efter en af de oplistede parametre.	4. Systemet oplister igen bilerne som falder inden for de angivne kriterier, med tilhørende biltype, udleveringssted, dato for sidste rengøring, dato for sidste serviceeftersyn og hvornår bilen er reserveret.
<i>Trin 3 og 4 springes over eller gentages så længe administrator ønsker det.</i>	
5. Administrator reserverer en bestemt bil i en bestemt periode.	6. Systemet kontrollerer om der er foretaget nogle medlemsreservationer i den angivne periode. Hvis dette er tilfældet returneres disse reservationer til administratoren, med det pågældende medlems medlemsnummer, telefonnummer og e-mail.
<i>Trin 2-7 gentages så længe administratoren ønsker det.</i>	
7. Systemet reserverer den valgte bil i den valgte periode, og indikerer over for administrator at reservationen har fundet sted.	
8. Administratoren afslutter brugsmønsteret.	
Alternative forløb:	
Bil reserveret af administrator	Såfremt der allerede er foretaget en vedligeholdelsesreservation af den pågældende bil i den angivne periode skal systemet give en fejlmeddelelse, da det ikke er muligt at foretage to vedligeholdelsesreservationer oven i hinanden. Det er dog muligt for administratoren at forlænge eller annullere reservationen, men dette falder inden for brugsmønstrene ”Forlængelse af reservation (B5)” og ”Annullering af reservation (B6)”.

4.8 ”Analyse af bilklubbens drift (B17)”

I brugsmønster B17 skal en række forskelligt data omhandlende Delebilklubben, dens medlemmer, biler og reservationer, trækkes ud af IT-systemet, og præsenteres for formanden, så han kan benytte det i sin analyse af klubbens drift. Det betyder, at dette brugsmønster er meget omfattende idet den del af systemet, som omhandler formandens analyse indeholder mange elementer. Derfor behandles der i første omgang kun solskinssceneriet for et af disse elementer, da det er tilstrækkeligt for at illustrere proceduren, og dermed formandens tilgang, til analyse af bilklubbens drift.

I tabellen herunder er angivet de krav som Delebilklubben ønsker at systemet skal opfylde i forbindelse med formandens analyse af Delebilklubbens drift:

Krav ID	Detaljer	Særlige forhold	Prioritet
F28	NDB skal gøre det muligt for formanden at få vist bilernes kørte km.		Want
F29	NDB skal gøre det muligt for formanden at få vist bilernes udlejningstid i en given periode.		Must
F30	NDB skal gøre det muligt for formanden at få vist antallet af tilkommende medlemmer i en periode.		Want
F31	NDB skal gøre det muligt for formanden at få vist reservationer pr medlem i en periode.		Want
F32	NDB skal gøre det muligt for formanden at få vist ulykker pr. år.		Want

Tabel 4.8: Beskrivelse af de krav som brugsmønsteret dækker over samt prioriteten af de enkelte krav.

Som sagt behandles kun et af disse elementer, som i første omgang bliver at indhente data omkring bilernes udnyttelsesgrad i en given periode.

Kravet F29 har derfor fået højeste prioritet fordi de som nævnt er væsentlige i forhold til formandens analyse af klubbens drift på nuværende tidspunkt, da det viser hvordan formanden kan bruge systemet. Det betyder samtidig at de resterende krav er prioriteret med laveste prioritet, da de tænkes at blive implementeret i fremtiden, eventuelt i adskilte brugsmønstre.

Der gøres desuden opmærksom på at listen af krav tilhørende dette brugsmønster sandsynligvis vil blive udvidet i forbindelse med at Delebilklubben får specificeret yderligere ønsker til analysen af klubbens drift. Det er dog på nuværende tidspunkt ikke nødvendigt at diskutere kravene nærmere med Delebilklubben, da der i første omgang kun ønskes illustreret proceduren for formandens tilgang til systemet.

4.8.1 Brugsmønsterbeskrivelse for B17

ID:	B17
Formål:	At give formanden for delebilklubben mulighed for at analysere bilklubbens drift.
Oversigt:	Formanden af delebilklubben skal foretage analyse af bilklubbens drift og skal i den forbindelse hente data ud af systemet. Når formanden har besluttet hvad der skal laves analyse af, vises data så formanden kan foretage analysen.
Primære aktører:	Formand
Sekundære aktører:	Ingen
Interesserter:	<i>Formand:</i> Ønsker at få mulighed for at få oplysninger så der kan foretages analyse af bilklubbens drift. <i>Delebilklub:</i> Ønsker at kunne optimere sin drift på baggrund af formandens analyser.
Prækondition:	Formanden har fået adgang til systemet (se brugsmønster ”Adgangskontrol (B1)” i inceptionsdokumentet).
Postkondition:	Data er vist så denne kan anvendes til analyse.
Krydsreferencer:	F28, F29, F30, F31, F32
Udvidelser:	Det vil senere blive muligt for formanden at få flere oplysninger fra systemet til yderligere analyse.
Typisk hændelsesforløb (Hovedsucces-scenariet)	
Aktør	System
1. Formanden vælger at foretage analyse af bilernes udlejningstid og antal kørte km.	2. Systemet adspørger hvilken periode oplysningerne skal vises for.
3. Formanden angiver periode.	4. Systemet beregner og viser udlejningstid, og procentuel udnyttelse for de enkelte biler indenfor den valgte periode.
6. Formanden angiver sit svar med ja eller nej.	5. Systemet adspørger formanden om oplysningerne skal gemmes i en fil.
8. Formanden afslutter brugsmønsteret.	7. Hvis ja. 7.1 Systemet gemmer data i en fil. 7.2 Systemet kvitterer for at have modtaget og gemt data.
Alternative forløb:	
Ingen	

5 Analyse

I foregående kapitel blev brugsmønsterbeskrivelser udarbejdet for de seks udvalgte brugsmønstre, og disse brugsmønsterbeskrivelser ligger til grund for den analyse som foretages i dette kapitel. I analysen udarbejdes et fælles analyseklassediagram for de seks brugsmønstre, og derudover vil hændelsesforløbet fra brugsmønsterbeskrivelserne blive illustreret i interaktionsdiagrammer.

For at undgå skift mellem dansk og engelsk fra analyse til design, udarbejdes samtlige UML-diagrammer i rapporten på engelsk. Dette gøres da koden ligeledes vil være udarbejdet på engelsk, og derfor skal designdiagrammer være udarbejdet på engelsk.

Først i kapitlet foreligger en generel beskrivelse af analyseklassediagrammet, hvor de enkelte analyseklasser vil blive beskrevet, og herefter gennemgås analysen for hvert af de seks brugsmønstre, med deres tilhørende analyseinteraktionsdiagrammer.

5.1 Generel beskrivelse af analyseklassediagram

I tabellen herunder er angivet en beskrivelse af de enkelte klasser, som optræder på analyseklassediagrammet. UML-diagrammet forefindes i bilag.

Klasse	Beskrivelse
CarClub	<p><i>CarClub</i> driver en delebilsordning for sine <i>members</i>. <i>CarClub</i> har et <i>name</i>, en <i>address</i>, en <i>chairman</i>, en <i>administrator</i>, <i>members</i>, <i>memberships</i>, <i>membershipTypes</i> og <i>cars</i>.</p> <p>Det skelnes mellem medlemmer som er blevet godkendt af delebilklubben (<i>members</i>), og medlemmer som har ansøgt om medlemskab, men som ikke er blevet accepteret endnu (<i>potentialMembers</i>).</p> <p><i>CarClub</i> kender både <i>members</i>, <i>memberships</i> og <i>membershipTypes</i>, idet <i>memberships</i> benyttes ved opgørelse af forbrug og <i>membershipTypes</i> benyttes ved oprettelse af nye <i>members</i>.</p> <p>Det er de enkelte <i>cars</i>, som er tilknyttet direkte til <i>CarClub</i>, og ikke <i>ParkingLot</i> eller <i>CarType</i>, da det er de enkelte <i>cars</i> som delebilklubben administrerer og udlejer. Når der foretages en reservation fra et <i>Member</i> er det således på en specifik <i>Car</i> og ikke blot på en <i>CarType</i> eller <i>ParkingLot</i>.</p>
Member	<p>Et <i>Member</i> er en person, der er tilknyttet en <i>CarClub</i>. Når en person ansøger om medlemskab i delebilklubben, angives det nye <i>Member</i> som potentielt medlem, med <i>status</i> til inaktiv. Potentielle medlemmer kan efterfølgende godkendes af administratoren, og deres <i>status</i> ændres til aktiv. Når det første medlem af en husstand indmelder sig i delebilklubben, medføre dette at der oprettes et nyt <i>Membership</i>.</p> <p><i>Member</i> er tilknyttet et <i>Membership</i> alene eller sammen med andre af samme husstand.</p> <p>Et <i>Member</i> er kendetegnet ved en række attributter som <i>name</i>, <i>birthday</i> og <i>phoneNumber</i>. Desuden indeholder <i>Member</i> informationer om medlemmets kørekort: <i>licenceNumber</i> og <i>licenceIssueDate</i>.</p> <p><i>Member</i> kan oprette <i>memberReservations</i> som tilknyttes direkte til det aktuelle <i>Member</i> og den aktuelle <i>Car</i>. Hvis et medlem ønsker at annullere en reservation, slettes reservationen ikke fra <i>Member</i>, men flyttes til en liste over annullerede reservationer. Dette gøres for at delebilklubben kan bevise over for medlemmerne at reservationen er blevet annulleret, og eventuelt for at kunne genskabe reservationer, som er annulleret ved en fejl.</p> <p>Når en reservation afsluttes, oprettes et <i>Usage</i> som tilknyttes det enkelte <i>Member</i> og når den månedlige forbrugsopgørelse udregnes, betales det <i>Usage</i>, som er blevet oprettet i løbet af den foregående måned. <i>Member</i> vil således have to former for <i>Usage</i>, henholdsvis betalte og ikke betalte.</p> <p>Administratoren er også et <i>Member</i> i <i>CarClub</i>, som til forskel fra almindelige medlemmer kan foretage vedligeholdelsesreservationer (<i>MaintReservation</i>).</p>
Membership	<p>Et <i>Membership</i> af en <i>CarClub</i> kan have tilknyttet en eller flere personer, der bor på samme <i>address</i>. <i>Membership</i> oprettes når et <i>Member</i> har ansøgt om medlemskabet af delebilklubben. Når et nyt <i>Membership</i> oprettes foregår det på samme måde som ved</p>

	<p>oprettelse af et nyt <i>Member</i>. Et <i>Membership</i> af delebilklubben er kendetegnet ved en bestemt <i>MembershipType</i>. Et <i>Membership</i> har en <i>address</i> og <i>zipCode</i> samt andre medlemsoplysningerne som fremgår af den nuværende tilmeldingsblanket.</p>
MembershipType	<p><i>MembershipType</i> specificerer medlemskabsbetingelser for et <i>Membership</i>. <i>MembershipType</i> er kendetegnet ved at have attributterne <i>registrationFeeFirstMember</i>, <i>registrationFeeOtherMembers</i> og <i>kmPrice</i> for de forskellige <i>carTypes</i>.</p>
Reservation	<p><i>Reservation</i> er en skabelon som <i>MaintReservation</i> og <i>MemberReservation</i> skabes ud fra. <i>Reservation</i> indeholder en <i>startDate</i> og <i>endDate</i>. Til hver <i>Reservation</i> er der tilknyttet ét <i>Member</i>, som foretager reservationen, samt én <i>Car</i>, da det er denne der reserveres. Hermed er der gennem <i>Car</i> desuden knyttet henholdsvis én <i>ParkingLot</i> og én <i>CarType</i> til reservationen. Det ses derudover at der til det enkelte <i>Member</i> og den enkelte <i>Car</i> kan være tilknyttet et vilkårligt antal <i>reservations</i> på samme tid. Når en reservation først er oprettet, slettes den ikke fra systemet selvom den er blevet annulleret, da det skal være muligt for delebilklubben at kunne bevise at reservationer har eksisteret. <i>Reservation</i> vil således altid være tilknyttet <i>member</i>, enten som <i>reservation</i>, <i>cancelledReservation</i> eller gennem et <i>Usage</i>-objekt.</p>
MemberReservation	<p>En <i>MemberReservation</i> foretages af et <i>Member</i> og vedrører en bestemt <i>Car</i> i en bestemt <i>period</i>, fra en bestemt <i>ParkingLot</i>. Når en <i>MemberReservation</i> afsluttes, fjernes den fra <i>Member</i> og der oprettes et nyt <i>Usage</i>-objekt, som reservationen tilknyttes. <i>Usage</i>-objektet tilknyttes derefter til <i>Member</i>.</p>
MaintReservation	<p>En <i>MaintReservation</i> foretages af <i>Administrator</i> og vedrører en bestemt <i>Car</i> i en bestemt <i>period</i>, fra en bestemt <i>ParkingLot</i>. <i>MaintReservation</i> indeholder en <i>type</i>, som er service eller rengøring, der beskriver den pågældende <i>MaintReservation</i>.</p>
Usage	<p><i>Usage</i> tilhører en afsluttet <i>MemberReservation</i> og indeholder <i>drivenKm</i>, <i>kmPrice</i>, <i>feeType</i> og <i>cancellationFee</i>. Disse oplysninger gemmes og bruges senere når der skal opgøres forbrug. Når en <i>MemberReservation</i> afsluttes oprettes et <i>Usage</i> hvortil <i>MemberReservation</i> tilknyttes. Det vil således være muligt at opgøre forbruget for de enkelte <i>members</i> ud fra <i>Usage</i>.</p>
Car	<p>En <i>Car</i> er en konkret bil, som <i>Member</i> kan benytte. En <i>Car</i> har et <i>licenceNo</i>, en <i>CarType</i> og hører til en bestemt <i>ParkingLot</i>. <i>Car</i> indeholder desuden omlysninger omkring <i>lastCleaning</i> og <i>lastService</i>, som benyttes af administratoren til at planlægge rengøring og serviceeftersyn af bilerne. Når en <i>Car</i> er anskaffet registreres den i systemet, og fra det tidspunkt er den til rådighed til <i>Reservation</i>. <i>Car</i> kan reserveres på to måder, dels som <i>MemberReservation</i> af medlemmer og dels som <i>MaintReservation</i> af administratoren.</p>
CarType	<p>En <i>CarType</i> er kendetegnet ved <i>name</i>, en <i>pricePrHour</i>, <i>pricePrDay</i> og <i>pricePrWeek</i> og en <i>kmPrice</i> for hver <i>MembershipType</i>. Enhver <i>Car</i> har én <i>CarType</i>, men der kan godt være flere <i>cars</i> af samme <i>type</i>.</p>
ParkingLot	<p><i>ParkingLot</i> indeholder informationer om de enkelte udleveringssteder som delebilklubben besidder. <i>ParkingLot</i> er således kendetegnet ved et <i>parkingLotID</i> og en <i>address</i>.</p>
KmPrice	<p><i>KmPrice</i> indeholder informationer om forbrugsprisen, for benyttelsen af de enkelte <i>carTypes</i>. I kraft af at delebilklubben ønsker at priser pr. kørt kilometer er forskellige afhængig af hvilket <i>Membership Member</i> har, og at <i>kmPrice</i> desuden varierer afhængig af <i>CarType</i>, er det nødvendigt at påhæfte en <i>kmPrice</i> på forbindelsen mellem <i>CarType</i> og <i>Membership</i>. Dette gøres ved hjælp af en associationsklasse, som illustrerer at der for hver enkelt af <i>CarType</i> er præcis én <i>kmPrice</i> vedhæftet pr. <i>MembershipType</i>.</p>

Tabel 5.1: Beskrivelse af de enkelte klasser i analyseklassediagrammet.

5.2 Analyse af ”Ansøgning om medlemskab (B3)”

Når det potentielle medlem vælger at ansøge om medlemskabet, så indebærer det oprettelse af et nyt medlem og et nyt medlemskab. En eventuel oprettelse af et nyt medlem fra samme husstand hører under brugsmønster ”Indmeldelse af ekstra husstandsmedlem (B9)”.

Der er tre klasser, der er relevante i forbindelse med analysen af brugsmønsteret B3: *CarClub*, *Member* og *Membership*. Når brugsmønsteret startes, bedes det potentielle medlem om at indtaste indmeldelsesoplysninger, og disse sendes herefter til *CarClub*. Derefter vil systemet oprette et nyt *Member*- og *Membership*-objekter med de indtastede oplysninger, og med attributten *status* sat til *false* (inaktiv), således at medlemmet og medlemskabet er deaktiveret i første omgang. Hermed kan administratoren i brugsmønster ”Administrer medlemmer (B12)” aktivere medlemskabet ved at ændre *status* til *true* (aktiv), således tages der højde for kravet F02.

Et *Member* objekt tilknyttes derefter til *Membership*-objektet, for at husstandsmedlemmet kan høre under et medlemskab. På længere sigt, kan flere husstandsmedlemmer være under samme *Membership*. Tilknytningen er nødvendig for at den månedlige opgørelse kan beregnes.

Ud over det, skal *Member* og *Membership* også tilknyttes til *CarClub*, således at de er tilgængelige for administratoren. Der afsluttes ved at systemet udskriver en kvittering for ansøgningen.

Proceduren for ansøgning om medlemskab er illustreret i interaktionsdiagrammet herunder:

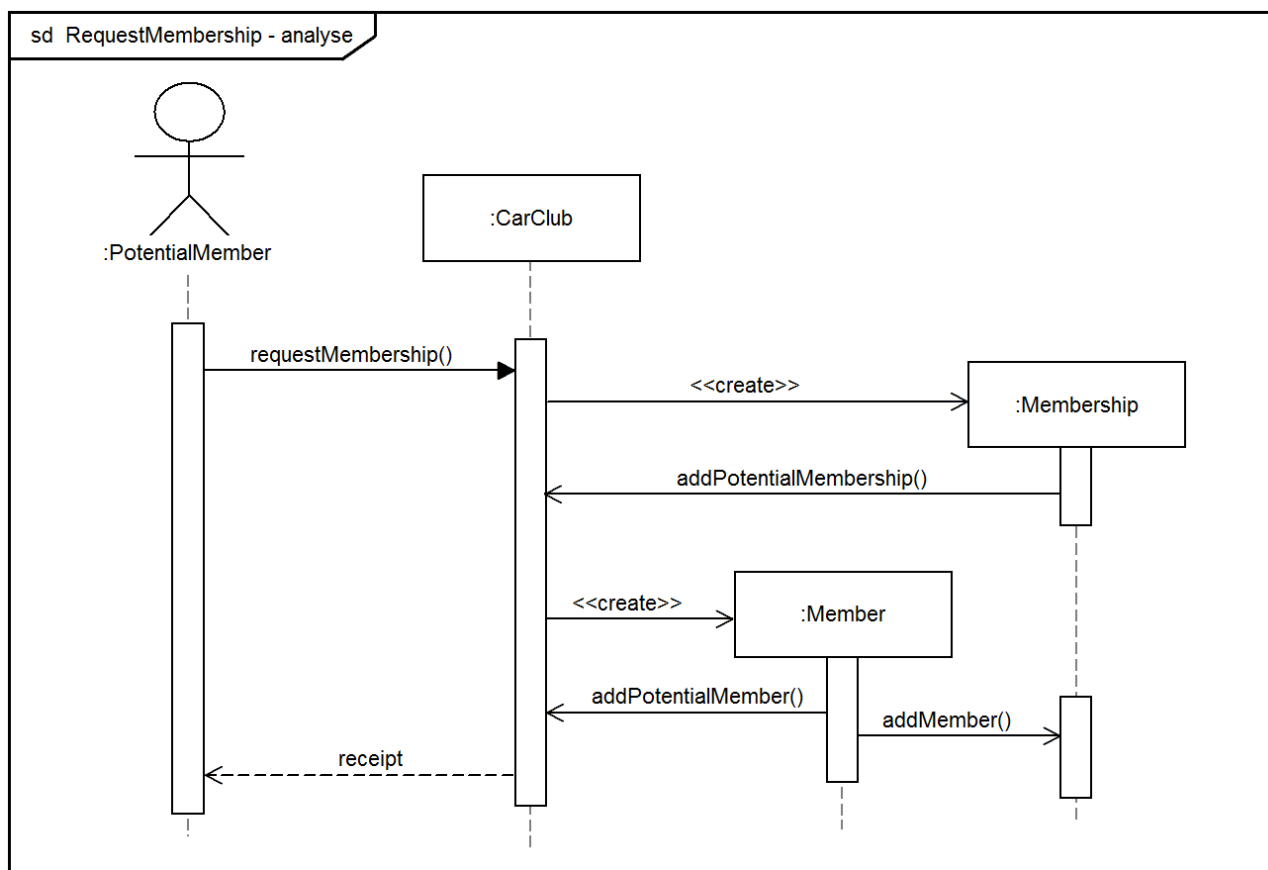


Diagram 5.1: Analyseinteraktionsdiagram for Ansøgning om medlemskab (B3).

Da der arbejdes med solskinsscenariet, udelades kontrol for fejl og mangler i de indtastede indmeldingsoplysninger.

5.3 Analyse af ”Reservation af delebil (B4)”

Brugsmønster B4 giver medlemmer af delebilklubben mulighed for at foretage reservationer af klubbens debiler.

Rent praktisk vil reservationen af en delebil foregå ved at *CarClub* først oplister samtlige biler, med deres tilhørende biltype, udleveringssted, kilometerpris og perioder hvor bilen er reserveret, ved at hente disse oplysninger fra de enkelte *Car*-objekter. Herefter er det muligt for medlemmet at gentage oplistningen, ved at sende eventuelle begrænsninger af de enkelte parametre til *CarClub*. En måde denne oplistning af debilerne kunne foregå på, er illustreret ved mockup i bilag 2.

Når medlemmet har bestemt sig for en bil og en reservationsperiode indtastes disse værdier og sendes til *CarClub*. Reservationen registreres dermed ved at der oprettes et nyt objekt af typen *MemberReservation*, og dette objekt tilknyttes derefter til både den aktuelle bil og det aktuelle medlem. Til sidst beregnes prisen for perioden, som *CarClub* returnerer sammen med en meddelelse om at reservationsproceduren er afsluttet.

Proceduren for reservation af delebil er illustreret i interaktionsdiagrammet herunder:

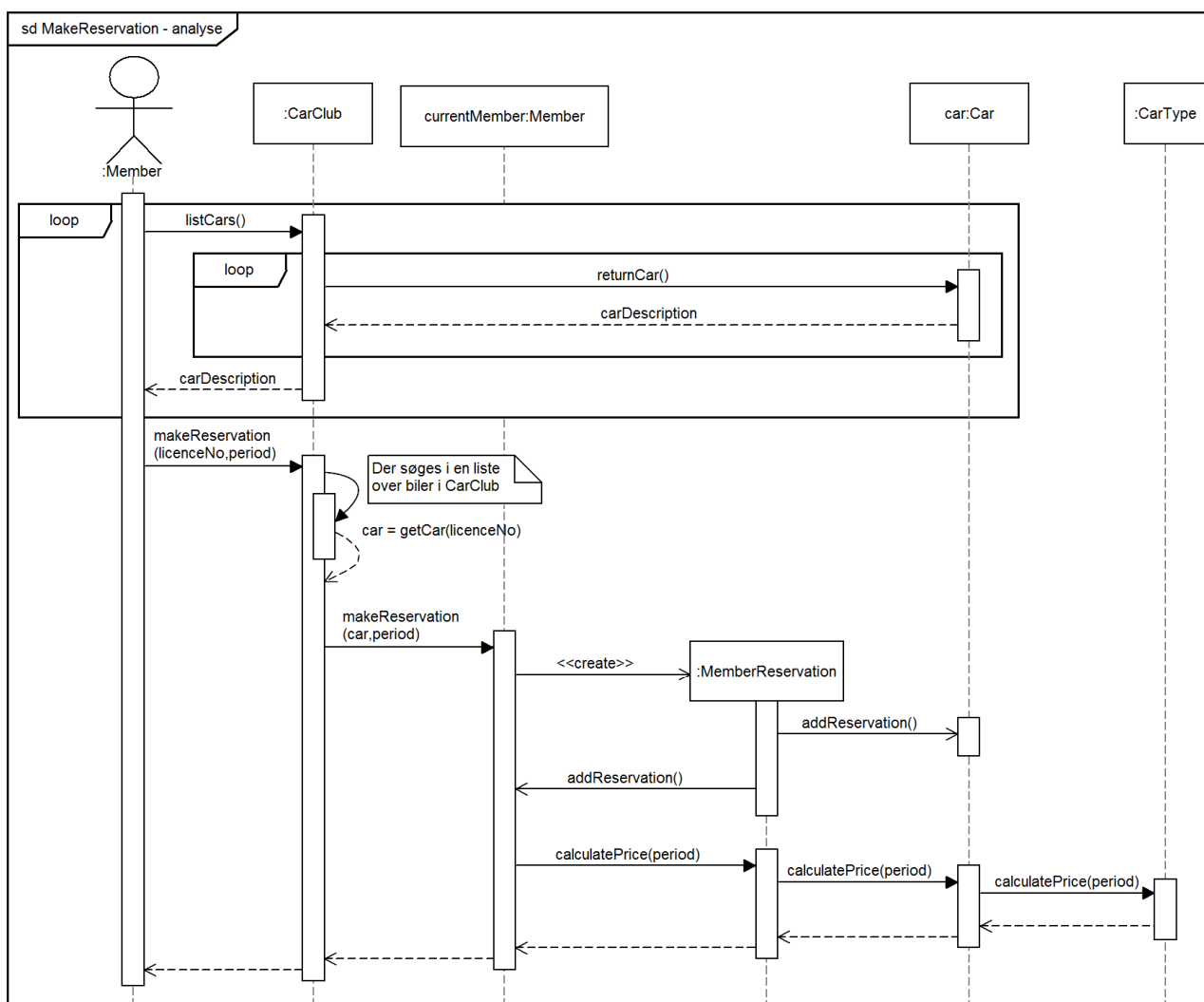


Diagram 5.2: Analyseinteraktionsdiagram for Reservation af Delebil (B4).

I første omgang er der ikke taget noget forbehold for hvorvidt de indtastede parametre er korrekte, eller hvorvidt bilen allerede er reserveret i det tidsrum man ønsker. Det er altså foreløbig muligt at foretage reservationer af biler, som allerede er reserveret i den angivne periode.

5.4 Analyse af ”Annullering af reservation (B6)”

Dette brugsmønster giver medlemmer mulighed for at annullere en reservation. Annullering af reservationer kan ske hvis medlemmet ikke skal benytte delebilen, men også i tilfælde af at delebilen ikke har været tilgængelig ved reservationens starttidspunkt.

Når et medlem ønsker at annullere en reservation oplister *CarClub* først alle medlemmets reservationer. Efter oplistningen kan medlemmet annullere en reservation ved at angive det pågældende *reservationID*. Herefter kontrolleres for de fire mulige scenarier, som beskrevet i tabel 4.6. Såfremt reservationen ikke medfører, at medlemmet pålægges et gebyr, flyttes reservationen fra listen over aktive reservationer til listen over annullerede reservationer. I tilfælde af at annulleringen medfører at medlemmet pålægges annulleringsgebyr oprettes desuden et *Usage*-objekt, som indeholder gebyr for at annullere for sent. *Usage*-objektet indeholder i dette tilfælde kun oplysninger om gebyr, idet bilen ikke er blevet benyttet.

Returbeskeden, *information*, som vises til medlemmet efter en reservation er annulleret, indeholder information om den reservation der er blevet slettet og fortæller brugeren om eventuel gebyr for annulleringen og størrelsen af denne.

Proceduren for annullering af reservation er illustreret i interaktionsdiagrammet herunder:

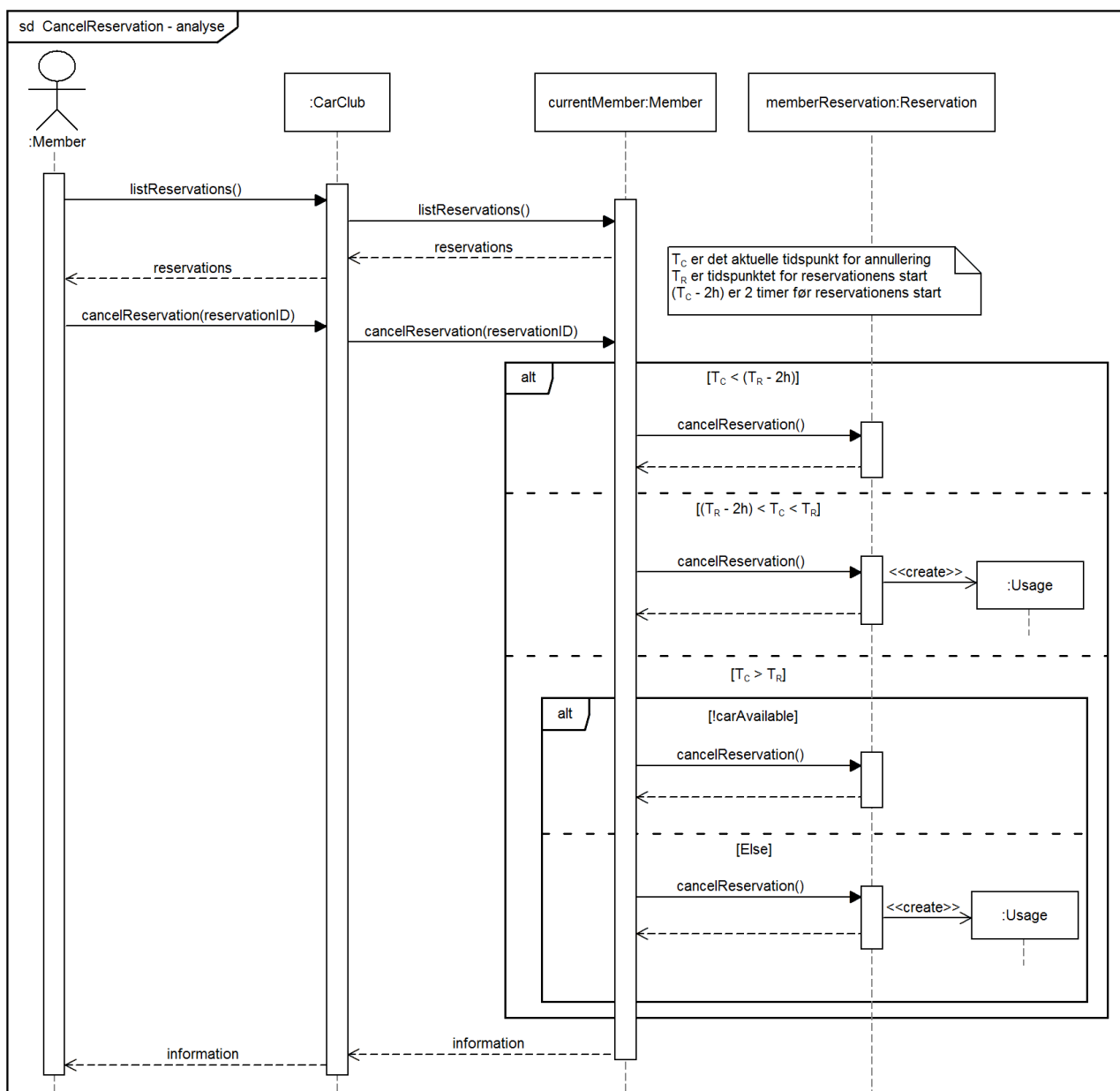


Diagram 5.3: Analyseinteraktionsdiagram for ”Annullering af reservation (B6)”.

5.5 Analyse af "Afl levering af delebil (B8)"

Når et medlem ønsker at afslutte en reservation, angiver han dette på bilens computer, som registrer medlemsnummer, reservationsnummer og kørte kilometer i reservationsperioden. Bilens computer foretager derefter et opkald til NDB, og sender de pågældende informationer via dette opkald. Disse informationer indlæses i systemet og sendes til *CarClub*.

CarClub bestemmer herefter det pågældende medlem, hvis reservation ønskes afsluttet, ud fra det oplyste *memberID* og *reservationID*. Systemet kan herefter afslutte reservationen ved at oprette et *Usage*-objekt hvori forbrugspris og kørte kilometer gemmes. *Usage*-objektet tilknyttes den pågældende reservation.

Proceduren for afslutningen af en reservation er illustreret i interaktionsdiagrammet herunder:

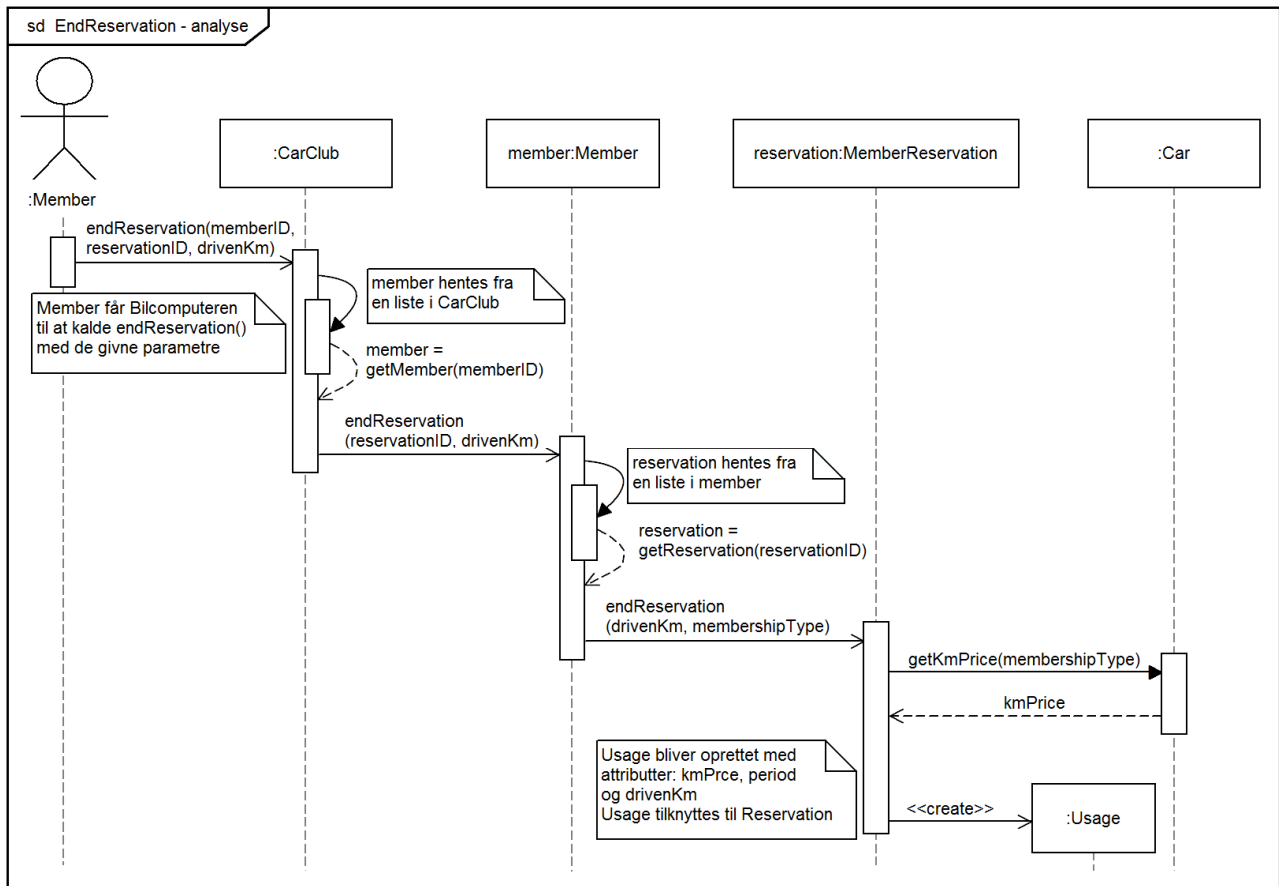


Diagram 5.4: Analyseinteraktionsdiagram for "Afl levering af delebil (B8)".

5.6 Analyse af ”Vedligeholdelsesreservation (B14)”

Dette brugsmønster giver administratoren mulighed for at reservere en bil til serviceeftersyn og rengøring, og af sin natur i analysefasen ligner det på mange punkter brugsmønsteret ”Reservation af delebil (B4)”. I de områder, hvor der benyttes lignende metoder, udelades en dybere analyse, da deres struktur allerede er beskrevet i afsnit 5.3.

Reservationen begynder ved at *CarClub* oplister alle reservationer der er foretaget på alle biler, inklusiv vedligeholdelsesreservationer. Den sidste dato for rengøring og service bliver også vist, så administratoren hurtigt kan danne sig et overblik over hvilke biler der skal vedligeholdes. Et forslag til hvordan opstillingen af delebilklubbens biler kan foregå er illustreret ved mockup i bilag 3.

Derefter angiver administratoren reservationsoplysninger, som sendes til *CarClub*. Der oprettes et objekt, *MaintReservation*, som tilknyttes til *Member* og *Car*. Undervejs i oprettelsen, kontrolleres der om der sker overlap med andre medlemsreservationer i *Car*, og hvis dette er tilfældet får administratoren kontaktoplysninger for de medlemmer, hvis reservation påvirkes, så disse medlemmer kan underrettes.

Proceduren for reservation af delebil til enten serviceeftersyn eller rengøring er illustreret i interaktionsdiagrammet herunder:

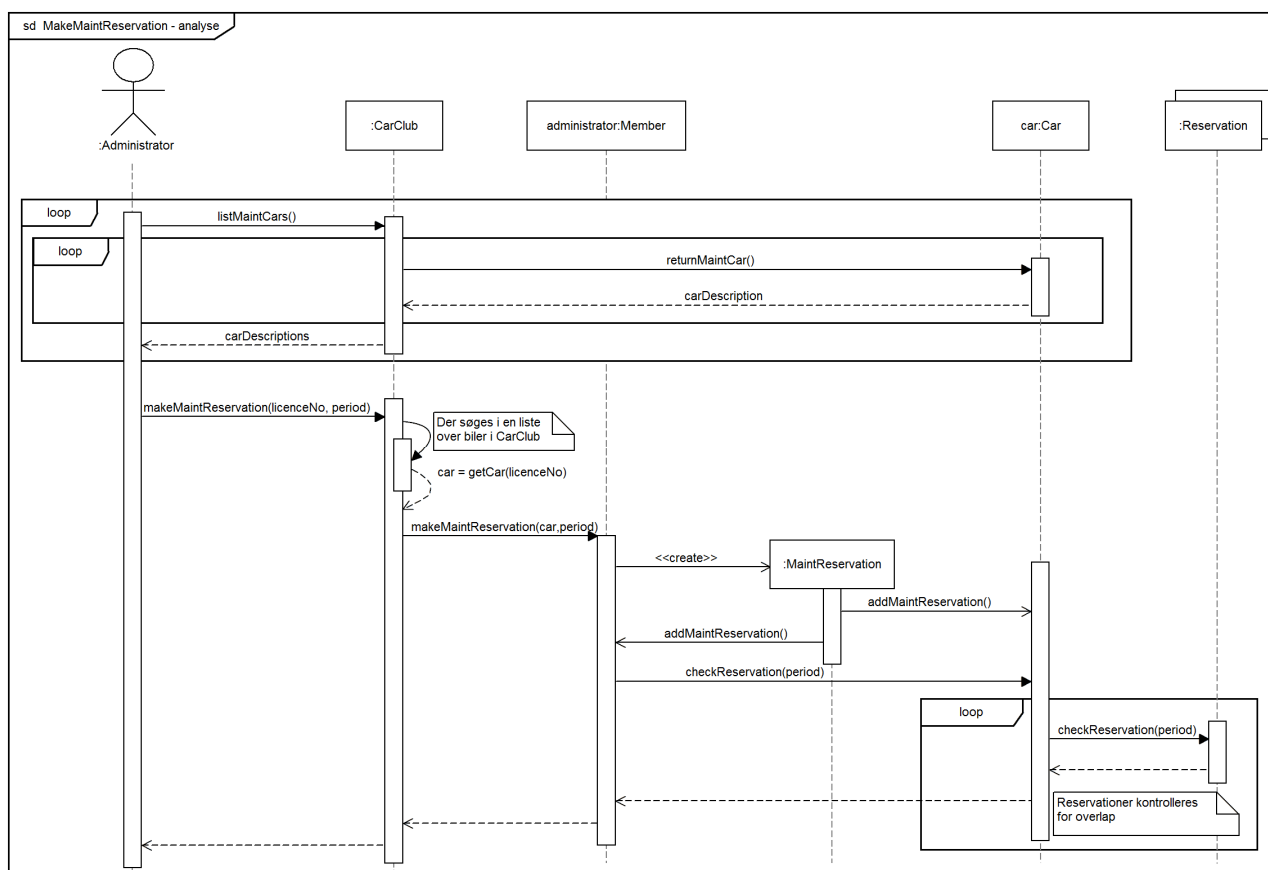


Diagram 5.5: Analyseinteraktionsdiagram for ”Vedligeholdelse af bil (B14)”.

5.7 Analyse af ”Analyse af delebilklubbens drift (B17)”

Brugsmønster B17 giver formanden for delebilklubben mulighed for at foretage en analyse af delebilklubbens drift. I første omgang henter brugsmønsteret udelukkende oplysninger om de enkelte bilers udnyttelsesgrad. Denne indsamling af information foregår via de enkelte *Car*-objekter, og benytter i vid udstrækning eksisterende metoder, som andre brugsmønstre har tilføjet til analyseklassediagrammet.

Når brugsmønsteret startes, angiver administratoren en periode for den analyse han vil foretage, og denne sendes til *CarClub*. *CarClub* vil herefter gennemgå hver enkelt *Car*-objekt og inden for angivne periode, hente længden af de enkelte *MemberReservation*-objekter. Når de ønskede oplysninger er indsamlet, vil de blive vist for formanden. Hver bil vil således have oplyst hvor mange timer den har været udlejet i den givne periode. Desuden vil den procentuelle udnyttelse af bilen blive beregnet og vist for formanden.

Efter opstillingen af de ønskede oplysninger vil formanden have mulighed for at få dem gemt i en tekstfil, så oplysningerne eventuelt kan eksporteres til et program der kan lave grafisk fremstilling af disse.

Proceduren for at foretage analyse af udnyttelsen af bilerne i delebilklubben er illustreret i interaktionsdiagrammet herunder:

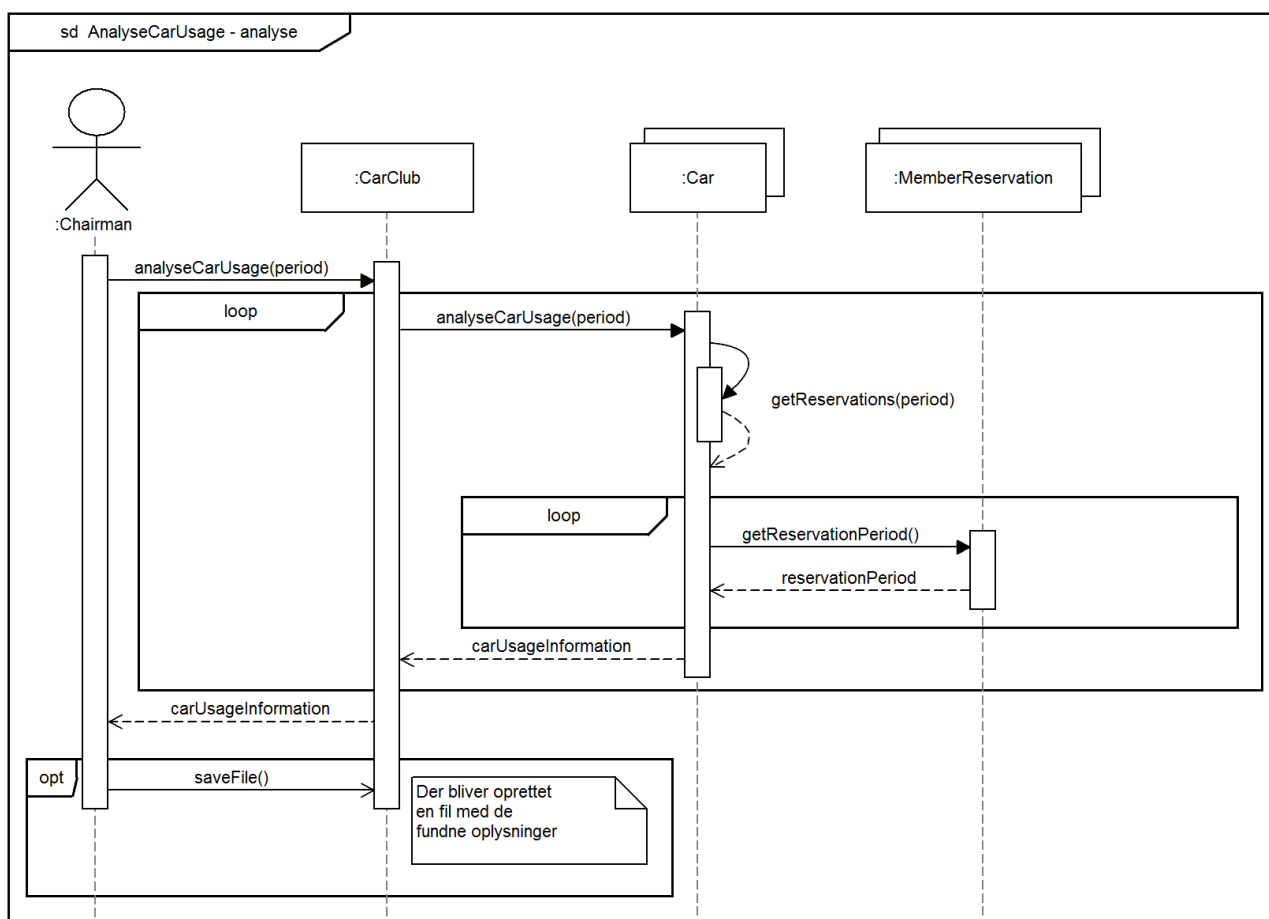


Diagram 5.6: Analyseinteraktionsdiagram for ”Analyse af delebilklubbens drift (B17)”.

6 Design

I det foregående kapitel blev analyse af de seks udvalgte brugsmønstre udarbejdet. Analysen af brugsmønstrene vil danne grundlag for det design der foretages i dette kapitel. I designet udarbejdes der først et fælles designklassediagram for de seks brugsmønstre, herefter udarbejdes designinteraktionsdiagrammer ud fra de foreliggende analyseinteraktionsdiagrammer.

Hvert af afsnittene i dette kapitel har et tilhørende UML-diagram, som forefindes som fold-ud sider, sidst i rapporten. Ved et par af de omfattende brugsmønstre vil nogle af de kaldte metoder kun være beskrevet i noter på designinteraktionsdiagrammerne, dette er gjort på grund af det ikke har været muligt at få plads til alle metoder på diagrammerne. De metoder der ikke er medtaget på designinteraktionsdiagrammerne, er de metoder der er vurderet til at være de simpleste.

IT-systemet er opdelt i tre lag *presentation*, *domain* og *technicalServices*. Der bliver i dette projektforløb primært arbejdet med *domain*-laget, som indeholder alle de metoder der sørger for at de ønskede funktionaliteter i IT-systemet er til stede. *presentation*-laget er opbygget som et lag der på et senere tidspunkt kan udskiftes med en grafisk brugergrænseflade og på nuværende tidspunkt udskrives data blot i et terminalvindue, da det ikke er et mål for projektet at udvikle en brugergrænseflade. På alle designinteraktionsdiagrammer vil der derfor være indført en *UI-klasse*, som er en repræsentation af *presentation*-laget.

technicalServices-laget indeholder klasser der blandt andet håndterer input, output og databaser.

Herunder ses de tre lag i IT-systemet:

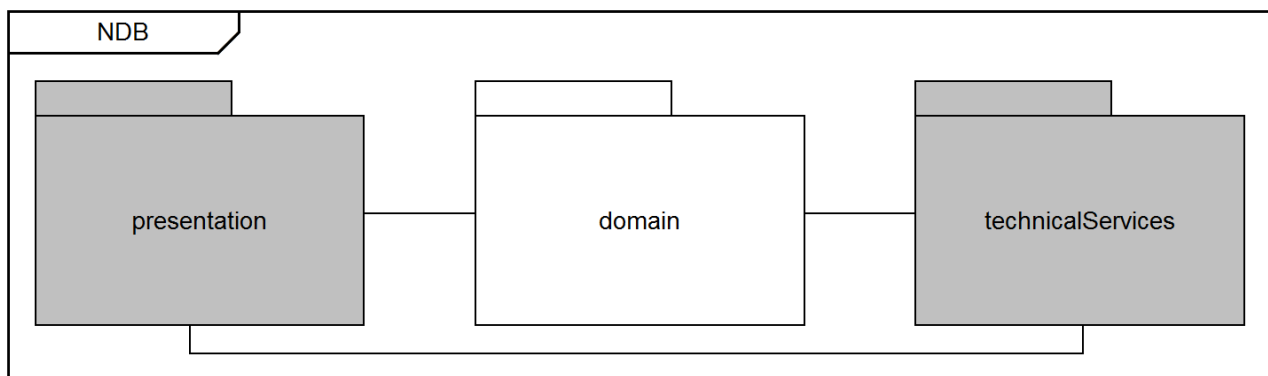


Diagram 6.1: De tre lag i IT-systemet.

Først i kapitlet er der en generel beskrivelse af designklassediagrammet, hvor forbindelserne mellem klasserne beskrives. Herefter gennemgås designet af de seks brugsmønstre, med udgangspunkt i de tilhørende fold-ud designinteraktionsdiagram.

6.1 Generel beskrivelse af designklassediagram

Designklassediagrammet er en videreudvikling af analyseklassediagrammet. De nye metoder der er blevet opdaget i forbindelse med designprocessen, er tilføjet, og de eksisterende udvidet, og parametrene og deres datatyper er nu angivet. Der er også tilføjet nye associationsforbindelser mellem designklasserne, og forbindelser med multiplicitet er ændret til aggregations- og composition-forbindelser. Som et eksempel, kan det nævnes, at *Member* og *Reservation* har en aggregations-forbindelse, da *Reservation* kan være tilknyttet et hvilket som helst *Member*, mens *CarClub* og *Member* har en composition-forbindelse, da *Member* er tilknyttet til det bestemte *CarClub*, Midtby Delebilklub.

Forbindelsesnavne er opdateret, så de nu repræsenterer de variabler der skal benyttes i kommunikationen mellem klasserne. Designklassen *Reservation* er ændret til en abstrakt klasse, da den indeholder en abstrakt metode *cancelReservation()*. Annulleringsproceduren i *MemberReservation* adskiller sig fra proceduren i *MaintReservation*, derfor er *cancelReservation()* først defineret i underklasserne.

Klassen *KmPrice*, som er en associationsklasse i analysen, er ændret til en normal klasse i designet, da intet almindeligt objekt orienteret programmeringssprog understøtter associationsklasser¹². Forbindelser fra *KmPrice* til andre klasser er derved også ændret. Designklassediagrammet indeholder alle metoder der kan findes i designinteraktionsdiagrammerne, da begge er et produkt af samme proces.

6.2 Design af ”Ansøg om medlemskab (B3)”

Af de datatyper, der er tilføjet til designklassediagrammet, er det værd at fremhæve at *memberID* og *membershipID* er af typen *String*, da ID’erne indeholder bogstaver.

Når det potentielle medlem angiver sine oplysninger beder systemet medlemmet om at acceptere godkendelsesproceduren. Dette registreres med attributten *accept* af typen *boolean*. Et medlem og medlemskab bliver kun oprettet såfremt *accept* er *true*.

Når et nyt medlems ID skal genereres, benyttes metoden *getNewMemberID()* der returner et unikt ID af typen *String* på formen ”mb###”. Metoden benytter variabelen *newMemberID* som er af typen *integer*, og som indeholder et nummer, der bliver forøget hver gang der oprettes et nyt ID. Denne *integer* tilføjes dermed ”mb” og returneres som *String*. Der kan være flere forskellige fremgangsmåder for oprettelse af unikke ID’er, men i første omgang benyttes denne simple løsning.

Indmeldingsoplysninger bliver gemt som tekst, og derfor er de af typen *String*. Disse oplysninger bliver ikke kontrolleret i solskinsscenariet, men i det videre udviklingsforløb kan datatypen ændres, hvis dette viser sig at være nødvendigt.

status er en attribut under *Member* og *Membership* der er af typen *boolean*, da den kan have to tilstande, aktiveret eller deaktiveret. Administratoren ændrer status til aktiveret (*true*) når han godkender det potentielle medlem i ”Administration af medlemmer (B12)”. Ændringen af *status* giver også mulighed for midlertidigt at deaktivere medlemskabet, hvis brugeren ikke ønsker at benytte sig af Delebilkubbens services i en periode.

Når de nye *Membership* objekter er oprettet, gemmes de i *potentialMemberships* som er en liste over alle nye ansøgninger om medlemskab. Administratoren kan se hvilke ansøgninger om medlemskab der venter på godkendelse og behandle dem i B12. Ligeledes gemmes *Member* i *potentialMembers* under *CarClub*. Objektet tilføjes også til *membersMembership* under *Membership*, dermed indeholder *Membership* en liste over samtlige medlemmer der er tilknyttet til medlemskabet. De to objekter bliver tilknyttet til de forskellige lister ved hjælp af *add*-metoder.

Til forskel fra analyseklassediagram, er den centrale del i oprettelsesmetoden placeret i en *if-else* sætning, der kontrollerer om ansøgeren har accepteret godkendelsesproceduren, således at ansøgningen ikke bliver afsendt hvis det potentielle medlem ikke accepterer denne.

6.3 Design af ”Reservation af Delebil (B4)”

Når brugsmønster B4 startes, oplistes først samtlige biler med deres nummerplade, biltype, udleveringssted, forbrugspris samt de reservationer der falder indenfor den af systemet fastlagte periode. Oplistingen kunne ske som det fremgår af mockup i bilag 2.

Rent praktisk vil denne oplistning foregå ved at *presentation*-laget, illustreret ved *ReservationUI*, henter de enkelte data der skal udskrives, fra de enkelte *Car*- og *Reservation*-objekter, som illustreret på interaktionsdiagrammet i bilag. Alle disse informationer er i første omgang gemt i de enkelte objekter som tekst. For at kunne kalde ned på den enkelte *Car* eller *Reservation* hentes først en liste over disse ved hjælp af henholdsvis *getCars()*, som kaldes på *CarClub*, og *getReservations()*, som kaldes på den enkelt *Car*. *getReservations()* kaldes med to parametre, *startStr* og *endStr*, som angiver den periode hvori reservationer ønskes oplistet. *getReservation()* kontrollerer derfor om den enkelte *Reservation* ligger inden for denne periode ved hjælp af *checkForOverlap()*. *checkForOverlap()* kaldes ligeledes med *startStr* og *endStr*, og returnerer *true* såfremt reservationen overlapper denne periode.

For at returnere forbrugsprisen for bilerne henter *presentation*-laget desuden *membershipType*, for *currentMember*, i *CarClub* som i første omgang blot returneres som en *String*. Denne

¹² Kilde 1 – Side 377

membershipType benyttes så i *carType* for at bestemme hvilke af de tilknyttede *KmPrice*-objekter, som stemmer overens med medlemskabstypen og *kmPrice* kan derefter returneres.

Efter den første opstilling af bilerne og deres tilhørende informationer, gives der mulighed for at gentage opstilling med en ny periode, og opstillingen vil herefter bliver gentaget med nye parametre for *startStr* og *endStr*.

Når medlemmet er færdig med at søge efter biler gives der mulighed for at foretage en reservation af en af delebilklubbens biler. Dette foregår ved at medlemmet angiver *licenceNo*, *startStr* og *endStr*, og disse parametre bliver benyttet af *CarClub* i metoden *makeReservation()*, til at oprette et nyt *MemberReservation*-objekt. *makeReservation()* kalder desuden *calculatePrice()* på det nyoprettede objekt, og prisen for reservationsperioden returneres og udskrives til medlemmet. Dette foregår i praksis ved at *MemberReservation* beregner reservationsperioden i timer ved hjælp af *getPeriodLength()*, og sender dette timeantal ud til den pågældende *CarType* via *Car*. I *CarType* bestemmes herefter hvor mange uger, dage og timer som det angivne timeantal dækker over og beregner derefter prisen ud fra de i *CarType* angivne *pricePrWeek*, *pricePrDay* og *pricePrHour*.

Efter at medlemmet har gennemført reservationen gives mulighed for at genstarte brugsmønsteret og foretage endnu en reservation eller at returnere til hovedmenuen.

6.4 Design af “Annullering af reservation (B6)”

Idet et medlem af delebilklubben starter brugsmønster B6, og ønsker at annullere en af sine reservationer, oplister systemet først alle medlemmets reservationer og det vil herefter være muligt for medlemmet at angive hvilken reservation som ønskes annulleret.

Oplistingen af reservationerne foregår, i stil med det foregående brugsmønster, ved at *presentation*-laget, her illustreret ved *CancellationUI*, henter en liste over medlemmets *Reservations* fra *CarClub*. *CarClub* kender *currentMember*, og kan derfor hente dette medlems reservationer ved *getReservations()*. *CancellationUI* henter i første omgang *reservationID*, *startStr* og *endStr* fra samtlige reservationer i denne liste.

Efter oplistingen får medlemmet mulighed for at angive *reservationID* på den reservation der ønskes annulleret, og ud fra dette ID kan *cancelReservation()* kaldes ud på den pågældende reservation via *CarClub* og *Member*. *cancelReservation()* vil returnere *fee* som er af datatypen *double*, der beskriver størrelsen af det gebyr der bliver pålagt medlemmet. Såfremt der ikke pålægges gebyr, vil *fee* være 0.

Foretages annulleringen mere end to timer før reservationens start, angivet ved dens *startDate*, kaldes *removeReservation()* på den pågældende *Member*, og *deleteReservation()* på den pågældende *Car*. *removeReservation()* flytter reservationen fra listen over reservationer (*reservations*) til listen over annullerede reservationer (*cancelledReservations*) i *Member*, mens *deleteReservation()* blot sletter reservationen fra *Car*. Hvis annulleringen sker mindre end to timer før reservationens start, men inden reservationen er begyndt, annulleres reservationen på samme måde som før, men der oprettes derudover et *Usage*-objekt hvortil reservationen tilknyttes. *Usage*-objektet oprettes med *drivenKm* og *kmPrice* sat til 0, idet bilen ikke har været benyttet, og med *feeType* sat til "cancel", som angiver at der skal påføres et annulleringsgebyr. Størrelsen af annulleringsgebyret, som i første omgang er 100 kr., ligger i *Usage*.

Annulleres reservationen derimod efter reservationens start, er det nødvendigt at hente afleveringstidspunktet på den reservation som ligger forud for den reservation, som ønskes annulleret. Dette gøres for at kontrollere hvorvidt delebilen har været tilgængelig ved starten af den reservation som skal annulleres. *previousReturnDate* hentes på den pågældende *Car*, ved hjælp af *getPreviousReturnDate()*, som først henter den foregående reservation med *getPreviousReservation()* og derefter henter dennes *returnDate*. Såfremt *previousReturnDate* ikke eksisterer endnu eller ligger efter *startDate* på den reservation som ønskes annulleret, har delebilen ikke været tilgængelig, og reservationen annulleres blot. Er dette ikke tilfældet har delebilen været tilgængelig, og medlemmet pålægges derfor et gebyr efter reservationen er annulleret.

Efter at reservationen er annulleret udskrives en kvittering for annulleringen og størrelsen af gebyret, hvis dette er blevet opkrævet. Medlemmet får derudover mulighed for at genstarte brugsmønsteret og foretage endnu en annullering eller at returnere til hovedmenuen.

6.5 Design af ”Aflevering af delebil (B8)”

Når medlemmet afslutter reservationen, sender bilcomputeren medlemmets og reservationens ID, samt kørte km til IT-systemet via en telefonlinje. Disse oplysninger modtages i form af DTMF-signaler¹³ og skal derfor afkodes¹⁴. Det sørger klassen *DTMF-decoder*, placeret i *technicalServices*, for. Klassen gemmer oplysningerne i tre variabler *memberID*, *reservationID* og *drivenKm*. ID’erne er af typen *String*, da de eksisterende medlemmer og reservationer er af denne datatype. Derved er det muligt at finde det pågældende medlem og reservation i systemet når bilen afleveres. *drivenKm* afkodes til et heltal af typen *int*, eftersom kilometertaksten afregnes pr. kørte km.

Ud fra de afkodende oplysninger, findes den pågældende reservation med *endReservation()*, via *CarClub* og *Member*. *CarClub* indeholder en liste over alle medlemmer, og *Member* indeholder en liste over dennes reservationer, derfor skal metoderne *getMember()* og *getReservation()* benyttes, for at finde frem til reservationen.

Når den pågældende reservation er fundet, registreres afleveringstidspunktet med *setReturnDate()*. Afleveringstidspunktet benyttes i ”Annullering af reservation (B6)”.

Da de kørte km er registreret i reservationen, kan *kmPrice* beregnes. Idet der er forskel på kilometertaksten for de to medlemskabstyper, bliver medlemskabstypen for den pågældende reservation fundet med *getMembershipType()*. Kilometerprisen bliver beregnet på samme måde som i ”Reservation af delebil (B4)”.

Der oprettes herefter et *Usage*-objekt hvori forbrugspris og kørte kilometer gemmes. *Usage*-objektet tilknyttes den pågældende reservation. Desuden tilføjes *Usage* til en liste, *unpaidUsage*, for det pågældende *Member* med *addUsageToUnpaid()*.

6.6 Design af ”Vedligeholdelsesreservation (B14)”

På mange områder minder brugsmønster B14 meget om ”Reservation af delebil (B4)”, og derfor vil nogle elementer i designet af B14 kun blive beskrevet overfladisk. For en mere detaljeret beskrivelse henvises til afsnit 6.3.

Når brugsmønster B14 startes, oplister systemet alle biler med deres tilhørende informationer, og generelt foregår denne opstilling som i B4. Dog oplistes forbrugsprisen ikke, da administratoren ikke skal bruge disse oplysninger. Derimod oplistes oplysninger omkring sidste rengøring, *lastCleaning*, og sidste serviceeftersyn, *lastService*, for de enkelte biler, samt informationer omkring reservationernes *type*. Denne *type* vil være enten almindelig, rengøring eller service, og skal bruges af administrator til at kende forskel på medlems- og vedligeholdelsesreservationer. Grunden til denne adskillelse er nødvendig i opstillingen, er at det ikke vil være muligt for administratoren at reservere oven i en vedligeholdelsesreservation, men at han har mulighed for at overskrive medlemmernes reservationer. Derimod vil det være muligt at forlænge vedligeholdelsesreservationer frem for at overskrive dem, hvilket tænkes behandlet i ”Forlængelse af reservation (B5)”. At det ikke er muligt at foretage en vedligeholdelsesreservation oven i en eksisterende vedligeholdelsesreservation vil dog ikke blive implementeret i første omgang.

Efter at bilerne er blevet oplistet første gang får administrator, på samme måde som medlemmer, mulighed for at indskrænke eller udvide opstillingen ved at angive en ny periode.

Når administratoren er færdig med at opliste fortages en vedligeholdelsesreservation ved at administratoren angiver *licenceNo*, *type*, *startStr* og *endStr*. *CarClub* kalder *makeMaintReservation()* med disse parametre for at oprette et *MaintReservation*-objekt. Reservationstypen gemmes i første omgang som en attribut af typen *String* i *MaintReservation* med værdier ”r” for rengøring og ”s” for service. Ved at have en attribut der angiver typen af reservationen er det, ud over at gøre det muligt at skelne mellem reservationerne i opstillingen, muligt at udvide systemet såfremt der ansættes en rengøringsassistent til rengøring af bilerne. Rengøringsassistenten kan så tildeles sit eget login og dermed få mulighed for at foretage rengøringsreservationer, men skal selvfølgelig ikke have mulighed for at reservere en bil til service.

¹³ DTMF står for Dual-Tone Multi-Frequency, og er betegnelsen for de toner som tastetryk på telefoner frembringer.

¹⁴ Se appendiks 1

Når *makeMaintReservation()* kaldes på *CarClub* returneres desuden en liste over de reservationer på den pågældende bil, som ligger inden for den af administratoren angivne periode. Denne liste hentes igen ved hjælp af *getReservations()*, og kan selvfølgelig være tom såfremt der ikke eksisterer nogen overlappende reservationer. Hvis listen ikke er tom indhentes informationer omkring de pågældende reservationer og de medlemmer som har foretaget dem, og disse oplysninger udskrives derefter til administratoren, så det er muligt for ham at kontakte de berørte medlemmer og ændre deres reservationer. Denne metode kan i fremtiden udvides så medlemmerne automatisk vil få besked idet en af deres reservationer overskrives af administratoren.

6.7 Design af ”Analyse af delebilklubbens drift (B17)”

Idet formanden for delebilklubben starter brugsmønster B17, og ønsker at foretage analyse af bilerne i bilparken, hentes en liste med alle bilerne. Det vil herefter være muligt at hente følgende informationer om bilerne: Nummerplade, biltype, udlejningstimer og udnyttelsesgrad.

Oplistingen af informationer for de enkelte biler foregår ved at presentationslaget, illustreret ved *analyseUI*, henter alle bilerne fra *CarClub*, med *getCars()*, der returner en liste. Herefter bliver der i et loop, i *analyseUI*, hentet nummerplade med metoden *getLicenceNo()*, biltype med metoden *getCarType()*, udlejningstimer med metode *getReservationHours()* og udnyttelsesgrad med metoden *calculateUsagePercentage()*, for alle bilerne. Mens informationerne hentes fra de enkelte biler bliver disse gemt som en *String* i en *ArrayList*, *carInfo*. Efter informationerne er blevet oplistet for bilerne, har formanden mulighed for at få gemt disse i en Excel-fil, med metoden *saveFile()* i *CarClub*.

Metoden *getReservationHours()* kaldes med to attributter, *startStr* og *endStr*, og disse indeholder start- og slutdato for analyseperioden. Når *getReservationHours()* kaldes på den enkelte *Car* hentes der først alle de *MemberReservations*, der overlapper den angivne periode. Herefter startes der et loop over alle reservationerne, hvor der med metoden *getAnalysisLength()* returneres hvor mange timer af reservation der ligger indenfor den givne periode. Det vil sige at hvis en reservation for eksempel starter før perioden, men slutter inde i perioden bliver timerne beregnet fra periodens start og til reservationen slutter.

Metoden *saveFile()* kaldes med to attributter, *carInfo* og *fileName*. *carInfo* er den *ArrayList* der indeholder de fundne analyseoplysninger og *fileName* er en *String* der indeholder det ønskede filnavn. Når filen skal skrives hentes hver enkelt linje ud af *ArrayList* og skrives til filen, dette gøres med alle de linjer der er i listen. Filen gemmes i Microsoft Excel format.

7 Implementering

I dette kapitel redegøres der for implementeringen af de centrale dele af kildekoden. De dele der ikke bliver gennemgået, er primært simple metoder, som henter og flytter informationer i systemet. Opbygningen af disse kan forstås ud fra det Javadoc der er lavet til kildekoden, som forefindes på den medfølgende cd. På cd'en medfølger der desuden: En eksekverbar programfil og en brugervejledning, samt en instruktionsvideo der viser programmets funktionaliteter.

Selve implementeringen er foretaget på baggrund af designklassediagrammet og designinteraktionsdiagrammerne. Designklassediagrammet beskriver attributterne og metoderne for de enkelte klasser og forbindelser mellem disse, mens designinteraktionsdiagrammerne viser hvilke kald metoderne foretager på hvilke klasser.

Under implementeringen er der indbygget nogle testfaciliteter i systemet, blandt andet føres en log over indgående DTMF-signaler i klassen *DTMFdecoder*. Denne log kan desuden benyttes til at demonstrere systemets funktionalitet over for delebilklubben i forbindelse med aflevering af en delebil. Implementeringen er testet løbende, men der er ikke gennemført en endelig test af hele systemet, da systemet på nuværende tidspunkt stadig er i elaborationsfasen.

Ud over gennemgang af de centrale metoder, vil der afslutningsvis være en gennemgang af *DTMFdecoder* klassen. Denne medtages da afkodningen af DTMF-toner, er den mest avancerede del af kildekoden. Det er desuden et krav fra projektoplæggets side at denne del skal behandles, og derfor beskrives implementeringen af denne i rapporten.

7.1 Implementering af `removeReservation`

Metoden `removeReservation()` benyttes i forbindelse med annullering af reservationer, og kaldes på *Member*, for at annullere en reservation. `removeReservation()` kaldes med et argument, `reservationID`, der er en *String* indeholdende ID'et på den reservation som ønskes annulleret.

Når en reservation annulleres, slettes den ikke fra systemet, men den vil altid være tilknyttet *Member*. Dette skyldes at det skal være muligt at genskabe reservationer, hvis de annulleres ved en fejl. `removeReservation()` henter derfor referencen til den pågældende reservation ud fra `reservationID`, gemmer referencen i listen over annullerede reservationer, `cancelledReservations`, og sletter den til sidst fra `reservations`.

Reservationer er tilknyttet *Member* ved brug af et *Map* af typen *HashMap*, kaldet `reservations`. Dette gør det hermed muligt at foretage direkte opslag i *HashMap*'et ved brug af en *key*, i modsætning til eksempelvis en *ArrayList*, hvor det er nødvendigt at gennemse listen, for at finde det ønskede element. *HashMap*'s benyttes flere andre steder i kildekoden, hvor der er nødvendigt at kunne foretage direkte opslag efter elementer, for eksempel til listen over *Member*'s i *CarClub* og til listen over *Reservation*'s i *Car*. Som *key* benyttes en *String*, som refererer til de enkelte elementer. *keys* i `reservations` indeholder i dette tilfælde reservationens ID, som er på formen `"rs###"`.

Det er muligt at hente, gemme og slette elementer i *HashMap*'s henholdsvis ved hjælp af metoderne `get()`, `put()` og `remove()`. Disse metoder benyttes i `removeReservation()`, som vist i kildekode 7.1.

Herunder er vist kildekoden for `removeReservation()`:

Member.removeReservation()

```
213     public void removeReservation(String reservationID) {
214         Reservation cancelledReservation = getReservation(reservationID);
215         cancelledReservations.put(reservationID, cancelledReservation);
216         reservations.remove(reservationID);
217     }
```

Kildekode 7.1: Implementering af `removeReservation()`.

7.2 Implementering af checkForOverlap

Metoden `checkForOverlap()` benyttes til at kontrollere hvorvidt en *Reservation* overlapper en given periode. Metoden kaldes med to argumenter `startStr` og `endStr` af datatypen *String*, som angiver start- og slutdato for perioden. De to argumenter skal angives på formen: ”ÅÅÅÅ-MM-DD-TT:mm”.

`checkForOverlap()` kontrollerer om den angivne periodes start- og slutdato begge ligger før reservationens startdato eller begge ligger efter reservationens slutdato. Hvis dette er tilfældet overlapper reservationen ikke og der returneres *false*, ellers overlapper reservationen og *true* returneres.

I *Reservation* gemmes start- og slutdato for den enkelte reservation i form af kalender-objekter og i dette tilfælde benyttes klassen *GregorianCalendar*, som understøtter den gregorianske kalender. *GregorianCalendar* findes i *java.util* og hvert *GregorianCalendar*-objekt indeholder informationer om en enkelt dato, samt metoderne `before()` og `after()` til sammenligning med andre *Calendar*-objekter, hvilket gør den særdeles velegnet i dette tilfælde.

For at kunne sammenligne den angivne periode med reservationens start- og slutdato er det derfor nødvendigt at ændre argumenterne fra *String* til *GregorianCalendar*-objekter, hvilket sker ved at gemme `startStr` og `endStr` i `reqStartDate` og `reqEndDate`. Dette foregår med metoden `set()`, som kaldes på disse to *GregorianCalendar*-objekter. Metoden `set()` modtager fem argumenter af datatypen *integer*, for henholdsvis år, måned, dag, time og minut, og derfor skal `startStr` og `endStr` skal derfor opdeles og omdannes til *integers*. Det er her nødvendigt at være opmærksom på at *GregorianCalendar* benytter, at januar er angivet med 0, og det er derfor nødvendigt at subtrahere 1 fra månedens værdi, når datoen sættes.

Omdannelsen af `startStr` og `endStr` foretages henholdsvis med metoderne `substring()` og `parseInt()`. `substring()` kaldes på *String*-objektet og returnerer en del af den pågældende *String*. `parseInt()` kaldes herefter på klassen *Integer*, med denne *String* som argument, og dermed returneres det pågældende tal som argumentet indeholder, som en *integer*.

Herunder er vist kildekoden for `checkForOverlap()`:

Reservation.checkForOverlap()

```
37     public boolean checkForOverlap(String startStr, String endStr) {
38         boolean overlap = true;
39
40         this.reqStartDate.set(Integer.parseInt(startStr.substring(0, 4)),
41                               //GregorianCalendar benytter Januar = 0
42                               Integer.parseInt(startStr.substring(5, 7))-1,
43                               Integer.parseInt(startStr.substring(8, 10)),
44                               Integer.parseInt(startStr.substring(11, 13)),
45                               0);
46
47         this.reqEndDate.set(Integer.parseInt(endStr.substring(0, 4)),
48                             //GregorianCalendar benytter Januar = 0:
49                             Integer.parseInt(endStr.substring(5, 7))-1,
50                             Integer.parseInt(endStr.substring(8, 10)),
51                             Integer.parseInt(endStr.substring(11, 13)),
52                             0);
53
54         if ((reqStartDate.before(startDate) && reqEndDate.before(startDate)) ||
55             (reqStartDate.after(endDate) && reqEndDate.after(endDate))) {
56             overlap = false;
57         }
58
59         return overlap;
60     }
```

Kildekode 7.2: Implementering af `checkForOverlap()`.

7.3 Implementering af `getPreviousReservation`

Metoden `getPreviousReservation()` er placeret i klassen `Car`, og benyttes i forbindelse med annullering af reservation. `getPreviousReservation()` kaldes med argumentet `startDate`, som er et `GregorianCalendar`-objekt indeholdende en bestemt dato. Metoden returnerer den `Reservation` i den pågældende `Car`, som ligger lige før den angivne `startDate`. Metoden benyttes derved til at hente den `Reservation` som ligger forud for den reservation, som ønskes annulleret. Dette gøres ved at kalde `getPreviousReservation()` med `startDate` fra den reservation som ønskes annulleret.

`getPreviousReservation()` gennemløber de reservationer der ligger i `Car`, og tester for hver reservation, om denne ligger før den angivne `startDate`, og om den ligger tættere på denne, end den nuværende `previousReservation`. Hvis dette er tilfældet sættes `previousReservation` til den `Reservation` der testes på. Til slut returneres `previousReservation`, som enten er en tom reference, såfremt der ikke ligger nogen reservation forud for `startDate`, og ellers indeholder en reference til en `Reservation`.

For at gennemløbe det `HashMap` over de reservationer, der ligger i `Car`, benyttes klassen `Iterator`. Først hentes en collection over `HashMap`'ets `keys`, ved hjælp af `keySet()` og herefter oprettes et `Iterator`-objekt, på denne `Collection`. Dermed kan hver enkelt `key`, og dermed hver enkelt `Reservation` i `HashMap`'et, hentes.

Herunder er vist kildekoden for `getPreviousReservation()`:

`Car.getPreviousReservation()`

```
136     private Reservation getPreviousReservation(GregorianCalendar startDate) {
137         Reservation previousReservation = null;
138         Iterator reservationIterator = reservations.keySet().iterator();
139
140         while(reservationIterator.hasNext()) {
141             String currentKey = (String) reservationIterator.next();
142             Reservation currentReservation = reservations.get(currentKey);
143
144             if (previousReservation == null && startDate.after(currentReservation.getEndDate())) {
145                 previousReservation = currentReservation;
146             } else {
147                 if (startDate.after(currentReservation.getEndDate()) &&
148                     currentReservation.getEndDate().after(previousReservation.getEndDate())) {
149                     previousReservation = currentReservation;
150                 }
151             }
152         }
153
154         return previousReservation;
155     }
```

Kildekode 7.3: Implementering af `getPreviousReservation()`.

7.4 Implementering af getKmPrice

Denne metode benyttes til at returnere forbrugsprisen for en bestemt *CarType*, ud fra medlemmets medlemskabstype. *getKmPrice()* kaldes med argumentet *membershipType*, som er en *String* der indeholder navnet på den *MembershipType*, som forbrugsprisen skal bestemmes for.

Til hver *CarType* er tilknyttet en *ArrayList*, kaldet *kmPrices*, med et *KmPrice*-objekt for hver *MembershipType* der findes i systemet. *KmPrice*-objekterne, som er tilknyttet de enkelte *CarType*-objekter er placeret i et *ArrayList*, da det hermed er muligt at tilføje flere forbrugspriser, såfremt der oprettes flere medlemskabstyper i systemet.

getKmPrice() kører en *for*-løkke over listen af *KmPrice*-objekter tilknyttet den pågældende *CarType*, hvor medlemskabstypen for hvert *KmPrice*-objekt hentes og sammenlignes med den angivne *membershipType*, ved hjælp af metoden *equals()*. Stemmer disse to overens hentes den pågældende *kmPrice* fra *KmPrice*-objektet, og denne værdi returneres herefter.

Herunder er vist kildekoden for *getKmPrice()*:

CarType.getKmPrice()

```
78     public double getKmPrice(String membershipType) {
79         double kmPrice = 0;
80
81         for (int count = 0; count < kmPrices.size(); count++) {
82             if (membershipType.equals(kmPrices.get(count).getMembershipType())) {
83                 kmPrice = kmPrices.get(count).getKmPrice();
84             }
85         }
86
87         return kmPrice;
88     }
```

Kildekode 7.4: Implementering af getKmPrice().

7.5 Implementering af saveFile

Metoden gemmer data fra formandens analyse af delebilklubbens drift i en fil. Metoden kaldes med to argumenter, hvor det første argument er en *ArrayList* der indeholder analysedata og det andet argument er en *String* der indeholder det indtastede filnavn, *fileName*.

Først oprettes et *FileWriter*-objekt, som opretter en fil hvortil der kan skrives. *FileWriter* oprettes med *fileName* som argument. *fileName* tilføjes filefternavnet *.xls*, som angiver at den oprettede fil er et Microsoft Office Excel dokument. Herefter oprettes et *PrintWriter*-objekt med dette *FileWriter*-objekt som argument. *PrintWriter* benyttes til at skrive til filen, hvilket foregår med metoden *println()*, som skriver en linje til filen.

Når data skrives til filen køres der en *for*-løkke over elementerne i *ArrayList*'en. Her hentes antallet af linjer med metoden *size()*. Når al data fra *ArrayList*'en er skrevet til filen, gemmes og lukkes denne. Lukkes filen ikke, vil data skrevet til filen ikke blive gemt.

Til sidst returneres meddelelsen, *result*, der indeholder kvittering for at filen er gemt samt selve filnavnet.

Herunder er vist kildekoden for *saveFile()*:

CarClub.saveFile()

```
355     public String saveFile(ArrayList<String> carInfo, String fileName) throws IOException {
356
357         FileWriter file = new FileWriter(fileName + ".xls");
358         PrintWriter out = new PrintWriter(file);
359
360         for (int i = 0; i < carInfo.size(); i++) {
361             out.println(carInfo.get(i));
362         }
363
364         out.close();
365
366         String result = "Filen er gemt som excelfil med navnet: " + fileName + ".xls";
367
368         return result;
369     }
```

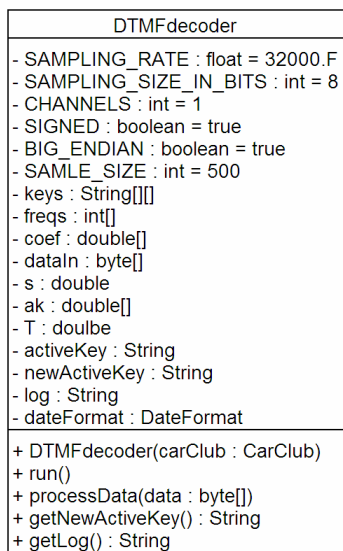
Kildekode 7.5: Implementering af *saveFile()*.

7.6 Implementering af DTMFdecoder klassen

Som det sidste i dette kapitel, redegøres der for klassen *DTMFdecoder*, der hører under *technicalServices*. De andre klasser i *technicalServices* bliver ikke beskrevet, da de kun er midlertidige klasser, og ikke vil være en del af det færdige system. I beskrivelsen fokuseres der ikke på de matematiske beregninger, men hvordan det indgående signal behandles.

For at forstå den matematiske del af *DTMFdecoder*, er det en forudsætning, at læseren har sat sig ind i Goertzels algoritme. Beskrivelsen af Goertzels algoritme findes i ”Appendiks 1 – Teori for afkodningen af DTMF-signaler”. I dette afsnit bliver der ikke gået i dybden med den matematiske del af implementeringen.

Designklassediagram for *DTMFdecoder*-klassen er illustreret herunder:



Figur 7.1: *DTMFdecoder*-klassen.
Enkelte attributter er udeladt.

Afkodning af DTMF-signaler fra lydindgangen skal foretages under hele programafviklingen. Samtidig skal systemet have en god responstid, dels så brugeren ikke oplever at systemet afvikles sløvt, og dels så registreringen ikke forsinkes og der ikke sker datatab. Der benyttes derfor en tråd, som oprettes ved programstart, og tager sig af metoderne i *DTMFdecoder*.

Klassen *DTMFdecoder* implementerer interfacet *Runnable* og dermed metoden *run()*, som kaldes idet en tråd startes med *DTMFdecoder*. Den kode som har ansvar for at foretage afkodningen placeres derfor i *run()*. Metoderne *processData()* og *getNewActiveKey()* tager sig af det tunge matematiske regnearbejde, og metoden *getLog()* returnerer log over lydinputtet.

Før data kan afkodes fra lydindgangen, skal dette først opsættes. Der benyttes et *AudioFormat*-objekt til at specificere lydformatet, som line-in skal behandle. *DataLine.Info* og *AudioSystem* kontrollerer om lydkortet understøtter det valgte lydformat, og hvis dette er tilfældet, åbnes og startes den specificerede *line* (line-in), som er et *TargetDataLine*-objekt. Når dette er foretaget, er systemet klar til at behandle data fra lydindgangen.

I kildekoden til højre ses første halvdel af *run()*, hvor lydindgangen opsættes. Der er indført løbende kommentering på dansk, for at gøre kildekoden mere forståelig.

DTMFdecoder.run() – del 1

```
65     public void run() {
66         // Line in hvorfra data læses
67         TargetDataLine line;
68
69         // Formatet som data læses i
70         AudioFormat format = new AudioFormat(SAMPLING_RATE, SAMPLING_SIZE_IN_BITS, CHANNELS,
71                                               SIGNED, BIG_ENDIAN);
72
73         // Opretter et info-objekt som beskriver specifikationerne på den line in der skal læses fra
74         DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);
75
76         //Kontrollerer om line in understøtter de givne specifikationer
77         if (!AudioSystem.isLineSupported(info)) {
78             Date timestamp = new Date();
79             log += dateFormat.format(timestamp) + ": Line In er ikke understøttet: ";
80             log += info + "\n\n";
81             return;
82         }
83
84         try {
85             // Opsætter en line in med de givne specifikationer
86             line = (TargetDataLine) AudioSystem.getLine(info);
87
88             // Åbner line
89             line.open(format);
90             Date timestamp1 = new Date();
91             log += dateFormat.format(timestamp1) + ": Line In er åben.\n";
92
93             // Starter line
94             line.start();
95             Date timestamp2 = new Date();
96             log += dateFormat.format(timestamp2) + ": Line In er startet.\n";
97
98         } catch (LineUnavailableException exception) {
99             Date timestamp = new Date();
100            log += dateFormat.format(timestamp) + ": Line In er ikke tilgængelig.\n";
101            return;
102        }
103    }
```

Kildekode 7.6: Implementering af run(), del 1.

Efter opsætningen af lydindgangen, kan data hentes ind og databehandlingen påbegyndes. Metoden *read()*, i *TargetDataLine*, benyttes til at læse data fra *line* og gemme dette i et *Array*, *dataIn*. Når 500 samples¹⁵ er gemt i *dataIn*, benyttes metoden *processData()* til at finde ud af hvilket DTMF-signal der blev modtaget. I *ProcessData()* beregnes Goertzels algoritme på det samlede data i henhold til appendiks 1. Slutresultatsværdier for hver frekvenskomponent gemmes i et *Array*, *ak*.

Når alle slutresultatsværdier er fundet, benyttes metoden *getNewActiveKey()* til at sammenligne værdierne i *ak*, og dermed bestemme hvilket DTMF-signal der er modtaget. Der kontrolleres hvilke to frekvenser signalet indeholder hvormed tastetrykket kan bestemmes i henhold til tabel A1.1.

Når det nye tastetryk er fundet, kontrolleres om det er forskellig fra forrige, da det kun i tilfælde af en ny værdi, der reelt er tale om et nyt tastetryk. Hvis dette er tilfældet, og det nye tastetryk er en firkant, forøges *countSquare*. Dette variabel benyttes til at separere *memberID*, *reservationID* og *drivenKm*. Hvis det nye tastetryk er et tal, kontrolleres værdien af *countSquare* for at bestemme, hvilken oplysning der afkodes på det givende tidspunkt. Tastetrykket tilføjes til en af de tre variabler.

Når antallet af firkanter når op på 3, betyder det at alle oplysninger er modtaget og afkodet, og dermed kan reservationen gemmes ved at kalde *endReservation* på den pågældende reservation via *CarClub* og *Member*.

I kildekoden til højre ses anden halvdel af *run()*, indhentet data behandles. Der er ligeledes indført løbende kommentering på dansk, for at gøre kildekoden mere forståelig.

¹⁵ Med en samplingsfrekvens på 32kHz, svarer 500 samples til ca. 15ms, derfor er det umuligt at misse et DTMF-signal. 500 samples er samtidig rigeligt til at afkode signalet.

DTMFdecoder.run() – del 2

```
104     // Sampling starter
105     while(line.isOpen()) {
106         // Et antal (SAMPLE_SIZE) samples indlæses fra line in
107         line.read(dataIn, 0, dataIn.length);
108
109         // Processerer de samlede data og gemmer Ak værdier for hver Goertzel algoritme i ak[]
110         processData(dataIn);
111
112         // Henter den aktive key ud fra det processerede data
113         newActiveKey = getNewActiveKey();
114
115         // Kontrollerer om den nye aktive key er forskellig fra den forrige
116         if (!activeKey.equalsIgnoreCase(newActiveKey)) {
117             // Kontrollerer om den før aktive key er forskellig fra "None"
118             if (!activeKey.equalsIgnoreCase("None") && count > 2) {
119                 input += activeKey;
120
121                 // Opdeler signalet i memberID, reservationID og drivenKm ud fra antallet af '#'
122                 // Det ingående signal er på formen: memberID # reservationID # drivenKm #
123                 if ((countSquare == 0) && (activeKey != "#"))
124                     memberID += activeKey;
125
126                 if ((countSquare == 1) && (activeKey != "#"))
127                     reservationID += activeKey;
128
129                 if ((countSquare == 2) && (activeKey != "#"))
130                     drivenKm += activeKey;
131
132                 if (activeKey == "#")
133                     countSquare++;
134
135                 // Opdaterer log såfremt et nyt signal er indgået:
136                 if (countSquare == 3) {
137                     carClub.endReservation(memberID, reservationID, Integer.parseInt(drivenKm));
138
139                     input += ">";
140                     Date timestamp = new Date();
141                     log += dateFormat.format(timestamp) + ": DTMF input: " + input + "\n";
142                     log += "           Medlem " + memberID;
143                     log += " har afsluttet reservation " + reservationID;
144                     log += ". Kørte km.: " + drivenKm + "\n";
145
146                     input = "<";
147                     countSquare = 0;
148                     memberID = "mb";
149                     reservationID = "rs";
150                     drivenKm = "";
151                 }
152             }
153             // Aktiv key sættes til ny aktiv key
154             activeKey = newActiveKey;
155             count = 0;
156         }
157         else
158             count++;
159     }
160 }
```

Kildekode 7.7: Implementering af run(), del 2.

8 Resterende arbejde

I dette kapitel gennemgås det resterende arbejde for de seks brugsmønstre. Det primære resterende arbejde vil være de lavt prioriterede krav der ikke er bearbejdet endnu. Det skal desuden påpeges, at de brugsmønsterbeskrivelser der er udarbejdet på baggrund af de udvalgte krav, er analyseret, designet og implementeret uden væsentlige mangler. Dette skyldes at der er fokuseret på at få implementeret de centrale dele af systemet, og der er derfor ikke, i bearbejdning af brugsmønstrene, gået i dybden med dele som ikke er centrale.

Først i kapitlet vil der være en generel beskrivelse af fælles mangler for de seks brugsmønstre, og efterfølgende vil der være en specifik beskrivelse af mangler for de enkelte brugsmønstre.

8.1 Generelt resterende arbejde

På nuværende tidspunkt er der ikke udviklet en færdig grafisk brugergrænseflade til systemet, dog er der et udarbejdet et midlertidigt *presentation*-lag der indeholder en brugergrænseflade, vist i terminalvindue, hvorfra systemet styres. Der er derfor ikke lavet nogen form for kontrol af de oplysninger der indtastes, da kontrollen afhænger af opbygningen af brugergrænsefladen.

Det skal påpeges, at der på nuværende tidspunkt ikke er taget højde for alle de alternative forløb i brugsmønstrene, disse er kun medtaget i brugsmønster ”Annullering af reservation (B6)”.

Der er derudover ikke taget højde hvem man logger ind som, men dette ændres når brugsmønsteret ”Adgangskontrol (B1)” behandles. Det betyder blandt andet at man skal være logget ind for at ansøge om medlemskab og at medlemmer kan foretage vedligeholdelsesreservationer. Det er dog ikke vigtigt med denne begrænsning på nuværende tidspunkt, for at kunne præsentere de forskellige funktionaliteter i systemet.

I analysen er nævnt en liste, *unpaidUsage* der skal indeholde betalte regninger, denne liste er ikke medtaget i design og implementering, da denne liste ikke skal benyttes før brugsmønster ”Opgørelse af bilforbrug (B16)” behandles.

8.2 Resterende arbejde i brugsmønstrene

De oplyste krav i tabellen herunder er ikke bearbejdet på nuværende tidspunkt, og er derfor en del af det resterende arbejde for de seks brugsmønstre.

Brugsmønstre	Krav ID	Beskrivelse
B3	F03	NDB skal ved indmeldelse sikre at de gældende betingelser for medlemskab er opfyldt, såsom alderskrav ol.
	F19	NDB skal registrere oplysninger om medlemmer svarende til de oplysninger der fremgår af nuværende indmeldelsesblanket.
B4	F23	NDB skal gøre det muligt at bestille biler hos tredjepart.
	K01	NDB skal gøre det muligt for medlemmer at reservere delebiler døgnet rundt.
B14	F24	NDB skal oplyse berørte medlemmer såfremt deres reservation annulleres af administrator.
	F25	Hvis et medlems reservation annulleres skal NDB tilbyde medlemmet en ny bil i den pågældende reservationsperiode. Hvis der ikke er nogen biler til rådighed hos debilklubben tilbydes en bil fra tredjepart.
B17	F28	NDB skal gøre det muligt for formanden at få vist bilernes kørte km.
	F30	NDB skal gøre det muligt for formanden at få vist antallet af tilkommende medlemmer i en periode.
	F31	NDB skal gøre det muligt for formanden at få vist reservationer pr medlem i en periode.
	F32	NDB skal gøre det muligt for formanden at få vist ulykker pr. år.

Tabel 8.1: Beskrivelse af de krav som hører under resterende arbejde.

Det resterende arbejde der fremgår af tabellen herunder, er punkter som på nuværende tidspunkt ikke er behandlet, men som bør behandles når IT-systemet videreudvikles.

Brugsmønster	Resterende arbejde
B3	På nuværende tidspunkt er det ikke muligt: - For det potentielle medlem, ved ansøgning, at angive alle de oplysninger der fremgår af indmeldelsesblanketten, der forefindes i projektoplægget.
B4	På nuværende tidspunkt er det ikke muligt: - At opliste biler efter biltyper og udleveringssted. - At undgå at reservere oven i en eksisterende reservation. - For systemet at opliste biler fra dags dato til et år frem i tiden, i første omgang oplistes der fra "2008-01-01-00:00" til "2010-01-01-00:00". Alternative forløb: - Administratoren skal kunne foretage reservationer på vegne af et medlem. - Den ønskede bil er reserveret.
B6	På nuværende tidspunkt er det ikke muligt: - For systemet at kontrollere om medlemmet har påbegyndt reservationen der ønskes annulleret. Alternative forløb: - Administrator skal kunne annullere vedligeholdelsesreservationer.
B8	På nuværende tidspunkt er det ikke muligt: - For systemet at kontrollere datatab ved afkodning af DTMF-signal.
B14	På nuværende tidspunkt er det ikke muligt: - At undgå at reservere oven i en eksisterende vedligeholdelsesreservation. - For systemet at opliste biler fra dags dato til et år frem i tiden, i første omgang oplistes der fra "2008-01-01-00:00" til "2010-01-01-00:00". - At administrator har mulighed for at ændre berørte reservationer. - For systemet automatisk at give de berørte medlemmer besked om ændringer i deres reservationer.
B17	På nuværende tidspunkt er det ikke muligt: - For formanden at hente analysedata for andet end bilforbrug.

Tabel 8.2: Beskrivelse af det resterende arbejde under de enkelte brugsmønstre.

I forbindelse med behandlingen af de udvalgte brugsmønstre er der i udviklingsforløbet opstået nogle alternative forløb, som er så omfattende at det er vurderet at de afføder nye brugsmønstre. De alternative forløb er opsummeret i tabellen herunder, på baggrund af brugsmønsterbeskrivelserne i kapitel 4 og det kan så senere vurderes hvorvidt det er nødvendigt at behandle disse i fremtidige elaborationsiterationer.

Brugsmønster	Alternativt forløb
B4	- Reservation af bil hos tredjepart.
B8	- Delebil afleveres for tidligt. - Delebil afleveres for sent.

Tabel 8.3: Beskrivelse af alternative forløb der afføder nye brugsmønstre.

Hermed afsluttes resultatdelen, og dermed dokumentation for det udviklede IT-system, som henvender sig til kunden og systemudviklere. Den resterende del af rapporten konkluderer på den opstillede problemformulering, og der vil derudover foretages en perspektivering, hvor det videre forløb i systemudviklingsprocessen beskrives.

9 Konklusion

Dette projekt beskæftiger sig med udviklingen af et IT-system til Midtby Delebilklub. Delebilklubben ønsker, at det udviklede IT-system skal give en aflastning af administratorens arbejde og en bedre service overfor medlemmerne, i forbindelse med udlejningen af klubbens debiler.

Formålet med denne rapport er at dokumentere resultatet af systemudviklingsforløbet, så det er muligt at præsentere den overordnede struktur af IT-systemet overfor delebilklubbens interessenter. Med den udarbejdede dokumentation vil det desuden være muligt at fortsætte det videre udviklingsarbejde af IT-systemet.

IT-systemet skal benyttes af de fire primære aktører henholdsvis potentielle medlemmer, medlemmer af delebilklubben, samt af delebilklubbens administrator og formand. Først er der derfor udvalgt brugsmønstre, som dækker over de centrale områder af IT-systemet og som tilgodeser de fire primære aktørers tilgangsvinkel. Disse områder indbefatter ansøgning om medlemskab for det potentielle medlem, reservation og aflevering af delebil, samt annullering af reservationer for klubbens medlemmer. Vedligeholdelsesreservation og analyse af delebilklubbens drift er behandlet for henholdsvis klubbens administrator og formand.

Ud fra den udleverede dokumentation¹⁶ og ved interview med delebilklubbens repræsentant¹⁷ har det været muligt at udfærdige kravmodel og brugsmønsterbeskrivelser for IT-systemet. På baggrund af disse har det herefter været muligt i samråd med delebilklubben, at fastlægge klubbens ønsker til IT-systemets funktionaliteter.

Ved hjælp af analyse- og designmodeller, udviklet på baggrund af kravmodel og brugsmønsterbeskrivelser, har det været muligt at implementere den overordnede struktur af IT-systemet, som overholder kundens ønsker.

På nuværende tidspunkt er systemet implementeret med en konsolbaseret brugergrænseflade, da det ikke har været et mål for projektet at udvikle en grafisk brugergrænseflade. I designet er der taget højde for at den nuværende brugergrænseflade kan udskiftes, uden der skal ændres i domænelaget, hvilket letter den videre udvikling af IT-systemet.

Fra projektoplæggets side er det fastlagt, at registrering af enten afhentning eller aflevering af debiler, via en telefon, skal behandles. Derfor er brugsmønsteret omhandlende aflevering af delebil implementeret og det er muligt, ved hjælp af en simulering, at illustrere overfor delebilklubben hvordan registreringen af en aflevering vil foregå.

På nuværende tidspunkt tilbyder systemet de væsentligste funktionaliteter i de seks udvalgte brugsmønstre, og det er hermed muligt at illustrere de fire primære aktørers tilgangsvinkel til IT-systemet overfor delebilklubben. Det udviklede IT-system giver desuden mulighed for at illustrere hvordan administrationsarbejdet lettes, idet det er muligt for medlemmer af delebilklubben at foretage reservationer på egen hånd. Derudover illustrerer IT-systemet hvordan der tilbydes en bedre service overfor klubbens medlemmer, idet det er muligt at registrere en aflevering af en delebil automatisk.

¹⁶ Projektoplægget indeholder inceptions-, case- og elaborationsiterationsdokument og dertilhørende kildekode.

¹⁷ Repræsentanten for delebilklubben har været gruppens vejleder.

10 Perspektivering

I perspektiveringen gives der forslag til det videre forløb for udviklingen af IT-systemet. Derudover gives en beskrivelse af den videre plan for udviklingsprocessen samt et overblik over de elementer, som bør ændres i det udviklede IT-system.

På nuværende tidspunkt foreligger der dokumentation for den i projektet udviklede del af system og det vil derfor være naturligt at foretage et review af systemet, dels med kunden og dels med eksterne konsulenter, for at sikre kvaliteten og at systemet lever op til kundes forventninger.

Efter der er foretaget review og systemet er blevet tilpasset så begge parter er enige, kan det videre udviklingsforløb fortsætte. I den næste elaborationsiteration vil det være naturligt at bearbejde brugsmønstrene: ”Forlængelse af reservation (B5)”, ”Administrer medlemmer (B12)” og ”Opgørelse af bilforbrug (B16)”, da der er taget højde for disse brugsmønstre i den udarbejdede dokumentation. Det vil derudover være oplagt at medtage nogle af de brugsmønstre der er opstået under den foreløbige udvikling, hvis de bidrager til komplicerede dele af systemet.

Efter elaborationsfasen, når størstedelen af brugsmønstrene er behandlet, påbegyndes konstruktionsfasen, hvor IT-systemet udfærdiges og testes. I konstruktionsfasen behandles *presentation-* og *technicalServices-*lagene, således at der er udarbejdet et færdigt system, som er klar til at blive sat i drift. Der vil være naturligt at udforme en grafisk brugergrænseflade i konstruktionsfasen og ændre klassen *DBManager*, så systemet tilknyttes en database.

Efter konstruktionsfasen påbegyndes transitionsfasen. I transitionsfasen tages det udviklede system i brug, og herefter vil brugere af systemet blive oplært i anvendelsen af systemet. På baggrund af tilbagemelding fra brugerne kan der foretages små justeringer af systemet og desuden laves små tilføjelser hertil, hvis dette er nødvendigt.

11 Litteraturliste

Herunder ses de bøger, internetsider, øvelser samt de programmer der er anvendt under projektet.

11.1 Anvendte bøger

1. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (Addison-Wesley Object Technology Series)
Paperback - 2005, 2. udgave, Addison-Wesley
ISBN-13 978-0-321-32127-5
2. Java Software Solutions
Paperback - 2008, 6. udgave, Pearson Education (Us)
ISBN-13 978-0-321-54934-1
3. Digital signalbehandling
Paperback - 1998, 1. udgave, Teknisk Forlag
ISBN-13 978-87-571-0874-3
4. DIGITAL SIGNAL PROCESSING - Fundamentals and Applications
ISBN-13 978-012-374090-8
Forefindes på cd'en.

11.2 Anvendte internetsider

5. Sun Developer Network (SDN): <http://java.sun.com/j2se/1.5.0/docs/api/index.html>

11.3 Anvendte øvelser

6. Signal Processing First
Lab 09: Encoding and Decoding Touch-Tone Signals
Forefindes på cd'en.

11.4 Programliste

Diagram Designer v1.21 – *Forefindes på cd'en.*

BlueJ v2.2.1 – *Forefindes på cd'en.*

Java SE Runtime Environment (JRE) 6 Update 11 – *Forefindes på cd'en.*

Java SE Development Kit (JDK) 6 Update 11 – *Forefindes på cd'en.*

MATLAB R2007b

CCPdf converter v0.7 – *Forefindes på cd'en.*

PDF Tool Kit v1.12 – *Forefindes på cd'en.*

VidShot Capturer v1.0.0.1 – *Forefindes på cd'en.*

Notepad++ v5.11 – *Forefindes på cd'en.*

Appendiks 1 – Teori for afkodning af DTMF-signaler

Signalet fra telefonlinjen, der skal afkodes, består af to frekvenskomponenter, som vist i tabel A1.1¹⁸.

Frekvens:	1209 Hz	1336 Hz	1477 Hz	1633 Hz
679 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Tabel A1.1: Tastetryk fordelt på DTMF-frekvenser.

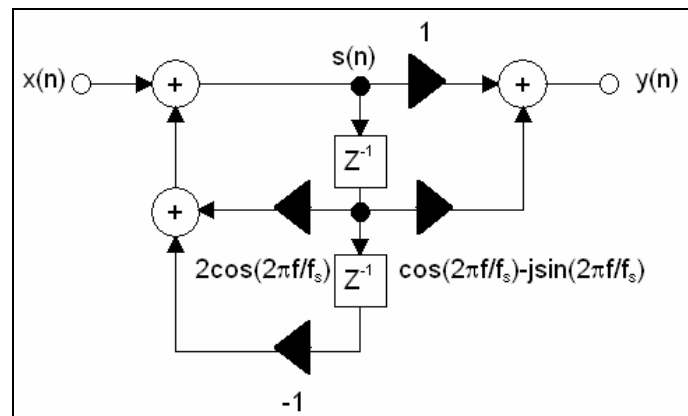
For at aflæse frekvenskomponenterne, skal signalet sendes igennem 8 IIR-filtre, hvert designet efter en af de mulige frekvenser. Det filter der repræsenterer den korrekte frekvens, vil have det største udgangsrespons, og de to filtre med den største udgangsrespons, vil dermed repræsentere et tastetryk. Disse 8 filtre kan designes ved anvendelse af Goertzel-algoritmen.

2.-ordens Goertzel filter har en overføringsfunktion defineret som:

$$H_f(z) = \frac{Y_f(z)}{X(z)} = \frac{1 - \left(\cos\left(\frac{2\pi f}{f_s}\right) - j \sin\left(\frac{2\pi f}{f_s}\right) \right) z^{-1}}{1 - 2 \cos\left(\frac{2\pi f}{f_s}\right) z^{-1} + z^{-2}} \quad (1)^{19}$$

hvor f er frekvensen, som filteret skal designes efter og f_s er samplingsfrekvensen.

Filteret realiseres som "Direkte type 2 struktur"²⁰, da den har færrest additioner og multiplikationer, dermed mindskes regnetiden. Blokdiagrammet over realisationsstrukturen kan ses på fig. A1:



Figur A1.1: Blokdiagram over Goertzels realisationsstruktur.

Udgangssekvensen bestemmes som:

$$y_f(n) = s_f(n) - \left(\cos\left(\frac{2\pi f}{f_s}\right) - j \sin\left(\frac{2\pi f}{f_s}\right) \right) \cdot s_f(n-1) \quad 0 \leq n \leq N \quad (2)^{21}$$

$s_f(n)$ er mellemresultat-sekvensen, der bestemmes som:

$$s_f(n) = 2 \cos\left(\frac{2\pi f}{f_s}\right) \cdot s_f(n-1) - s_f(n-2) + x(n) \quad 0 \leq n \leq N \quad (3)^{21}$$

Det er den sidste udgangsrespons værdi, $y_f(N)$, der benyttes til at sammenligne filtrene.

¹⁸ Kilde 4 – side 395

¹⁹ Kilde 4 – side 386

²⁰ Kilde 3 – side 315

²¹ Kilde 4 – side 387

Ifølge (2), afhænger udgangsresponsen kun af de sidste to $s_f(n)$ -værdier, og derfor skal de være beregnet først. Det vil altså sige, at $s_f(n)$ skal beregnes for alle samples, men det er kun de sidste to værdier der benyttes i (2) til at finde $y_f(N)$.

For at fjerne det komplekse led i (2), multipliceres $y_f(N)$ med den komplekse konjugerede. produktet fra udledningen, som udelades her, er:

$$|y_f(N)|^2 = s_f^2(N) + s_f^2(N-1) - 2 \cos\left(\frac{2\pi f}{f_s}\right) \cdot s_f^2(N) \cdot s_f^2(N-1) \quad (4)^{22}$$

Der tages herefter kvadratroden af (4), for at fjerne potensen:

$$A_k = \sqrt{y_f(N)^2} \quad (5)$$

Dermed er det A_k værdier for hvert filter der skal sammenlignes.

Det skal bemærkes at A_k skal være større end en grænseværdi, T , som sørger for at sammenligningen kun sker i det tilfælde, hvor der kommer et givet DTMF-signal på telefonlinjen, og at det ikke er støjfrekvenser der bliver sammenlignet. En passende grænseværdi kan være summen af alle A_k divideret med 4:

$$T = \frac{1}{4} \sum_{k=1}^8 A_k \quad (6)^{23}$$

²² Kilde 4 – side 387

²³ Kilde 4 – side 392

Bilag 1 – Tidsplaner

Herunder ses tidsplaner for projektet.

Overordnet tidsplan

Uge	40	41	42	43	44	45	46	47	48	49	50	51
Aktivitet	Projektgrundlag				Iteration 2			Iteration 3			Afslutning	
Projektoplæg udlevering	3/10											
Gennemlæsning af projektoplæg	P	P	P									
Projektgrundlag	P	P	P	P/U								
Præsentation af projektgrundlag				23/10								
Overordnet tidsplan				P/U								
Tidsplan for iteration 2				P/U								
Tidsplan for iteration 3							P/U					
Rapportskrivning				P	P	P	P	P	P	P	U	
Korrekturlæsning											P	P/U
Projektaflevering												18/12
Vejledermøder		9/10			30/10	7/11		21/11				
Grupperepræsentantmøde							14/11					
Forkortelser:	P står for påbegynd og U står for udført											

Tabel B1.1: Overordnet tidsplan for projektet.

Tidsplan for det aktuelle forløb i iteration 2

Aktivitet	Iteration 2														
	44					45					46				
Uger															
Dage	M	T	O	T	F	M	T	O	T	F	M	T	O	T	F
Kravmodel		B3/B4	B3/B4	B3/B4	B3/B4		B3/B4					B14	B14		
Brugsmønstermodel				B3/B4	B3/B4		B3/B4					B14	B14		
Interview				B3/B4	B3/B4		B3/B4								
Analyseklassediagram				B3/B4	B3/B4		B3/B4	B3/B4				B14	B14		
Analyseinteraktionsdiagrammer				B3/B4	B3/B4		B3/B4	B3/B4				B14	B14		
Designklassediagrammer							B3/B4	B3/B4	B3/B4	B3/B4	B3/B4	B4		B14	B14
Designinteraktionsdiagrammer								B3/B4	B3/B4	B3/B4	B3/B4	B4		B14	B14
Implementering										B3	B3	B3	B3/B4	B4	B14
Vejledermøde				30/10						7/11					
Grupperepræsentantmøde															14/11
Tidsplan for iteration 3															
Forkortelser:	P står for påbegyndt og U står for udført														
Brugsmønsteransvarlige	Alexey og Agge står for B3, Jakob og Frantz står for B4, og B14 behandles fælles														

Tabel B1.2: Tidsplan for iteration 2.

Tidsplan for det aktuelle forløb i iteration 3

Aktivitet	Iteration 3														
	47					48					49				
Uger															
Dage	M	T	O	T	F	M	T	O	T	F	M	T	O	T	F
Kravmodel		B6/B8		B6/B8			B17								
Brugsmønstermodel		B6/B8		B6/B8			B17								
Interview				B17	B17										
Analyseklassediagram				B6/B8	B6/B8		B17	B17							
Analyseinteraktionsdiagrammer				B6/B8	B6/B8		B17	B17							
Designklassediagrammer							B6/B8				B6/B8	B17	B17		
Designinteraktionsdiagrammer							B6/B8		B6/B8	B17					
Implementering											B8	B6	B17	B17	
Vejledermøde					21/11										
Brugsmønsteransvarlige	Alexey, Agge, Jakob og Frantz står fælles for B6, B8 og B17														

Tabel B1.3: Tidsplan for iteration 3.

Bilag 2 – Mockup for ”Reservation af delebil (B4)”

I dette bilag behandles mockup for brugergrænseflade til brugsmønster B4. Den angivne mockup er kun en skitse for brugergrænsefladen, og er på ingen måde bindende. Den er blot skitseret for at give et overblik over hvilke informationer systemet skal levere, så det senere er muligt at udvikle en brugergrænseflade til IT-systemet.

I tabellen herunder er selve mockup'en illustreret med tilhørende beskrivelse og de datakrav systemet skal opfylde, for at brugergrænsefladen kan realiseres:

<p>Udleveringssted: <input type="text" value="Alle"/></p> <p>Biltype: <input type="text" value="Alle"/></p> <p>◀ Forrige uge Uge 2 Næste uge ▶</p> <table border="1"><thead><tr><th></th><th>Mandag</th><th>Tirsdag</th><th>Onsdag</th><th>Torsdag</th><th>Fredag</th><th>Lørdag</th><th>Søndag</th></tr></thead><tbody><tr><td>Bil A Biltype Udleveringssted</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Bil B Biltype Udleveringssted</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Bil C Biltype Udleveringssted</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Bil D Biltype Udleveringssted</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Bil E Biltype Udleveringssted</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table> <p>Reservationer</p> <p>Reserver Delebil Annuller</p>			Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag	Bil A Biltype Udleveringssted								Bil B Biltype Udleveringssted								Bil C Biltype Udleveringssted								Bil D Biltype Udleveringssted								Bil E Biltype Udleveringssted							
	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag																																										
Bil A Biltype Udleveringssted																																																	
Bil B Biltype Udleveringssted																																																	
Bil C Biltype Udleveringssted																																																	
Bil D Biltype Udleveringssted																																																	
Bil E Biltype Udleveringssted																																																	
Aktør:	Medlem																																																
Beskrivelse:	<p>Et medlem ønsker at reservere en delebil og får derfor først oplistet alle biler i delebilklubben og deres tilgængelighed i en fastlagt periode.</p> <p>Medlemmet har herefter mulighed for at begrænse visningen af delebilklubbens biler ved at angive en bestemt biltype eller et bestemt udleveringssted. Det er desuden muligt for medlemmet at skifte mellem forskellige perioder ved at skifte til næste eller forrige uge.</p> <p>Når medlemmet har besluttet sig for en bil skal periode og den pågældende bil angives, hvilket kan ske enten ved at markere perioden i kalenderen, eller mere simpelt ved at have medlemmet til at indtaste disse oplysninger. Når periode og bil er angivet trykker medlemmet derefter på 'Reserver Delebil' og reservationen gemmes i systemet.</p>																																																
Data som hentes fra systemet ved oplistning af delebiler:	<p>De informationer som skal hentes for hver delebil, såfremt biltype og udleveringssted stemmer overens med de angivne værdier, er:</p> <ul style="list-style-type: none">- Indregistreringsnummer.- Biltype.- Udleveringssted.- Reservationer inden for en angivet periode.																																																
Data som sendes til systemet ved registrering af reservation:	<p>De informationer medlemmet sender til systemet ved oprettelse af en reservation er:</p> <ul style="list-style-type: none">- Dato for reservationens start.- Dato for reservationens slutning.																																																

Tabel B2.1: Beskrivelse af mockup for ”Reservation af delebil (B4)”

Bilag 3 – Mockup for ”Vedligeholdelse af bil (B14)”

I dette bilag behandles mockup for brugergrænseflade til brugsmønster B14. Den angivne mockup er kun en skitse for brugergrænsefladen, og er på ingen måde bindende. Den er blot skitseret for at give et overblik over hvilke informationer systemet skal levere, så det senere er muligt at udvikle en brugergrænseflade til IT-systemet.

I tabellen herunder er selve mockup’en illustreret med tilhørende beskrivelse og de datakrav systemet skal opfylde, for at brugergrænsefladen kan realiseres:

<p style="text-align: center;"> Udleveringssted: <input type="text" value="Alle"/> </p> <p style="text-align: center;"> Biltype: <input type="text" value="Alle"/> </p> <p style="text-align: center;"> ◀ Forrige uge Uge 2 Næste uge ▶ </p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>Mandag</th> <th>Tirsdag</th> <th>Onsdag</th> <th>Torsdag</th> <th>Fredag</th> <th>Lørdag</th> <th>Søndag</th> </tr> </thead> <tbody> <tr> <td>Bil A</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Biltype</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Udleveringssted</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste rengøring</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste service</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Bil B</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Biltype</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Udleveringssted</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste rengøring</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste service</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Bil C</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Biltype</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Udleveringssted</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste rengøring</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste service</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Bil D</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Biltype</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Udleveringssted</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste rengøring</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Sidste service</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"> ■ Rengøringsreservationer ■ Servicereservationer ■ Medlemsreservationer </p> <p style="text-align: center;"> <input type="button" value="Reserver Delebil"/> <input type="button" value="Annuller"/> </p>			Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag	Bil A								Biltype								Udleveringssted								Sidste rengøring								Sidste service								Bil B								Biltype								Udleveringssted								Sidste rengøring								Sidste service								Bil C								Biltype								Udleveringssted								Sidste rengøring								Sidste service								Bil D								Biltype								Udleveringssted								Sidste rengøring								Sidste service							
	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag																																																																																																																																																																		
Bil A																																																																																																																																																																									
Biltype																																																																																																																																																																									
Udleveringssted																																																																																																																																																																									
Sidste rengøring																																																																																																																																																																									
Sidste service																																																																																																																																																																									
Bil B																																																																																																																																																																									
Biltype																																																																																																																																																																									
Udleveringssted																																																																																																																																																																									
Sidste rengøring																																																																																																																																																																									
Sidste service																																																																																																																																																																									
Bil C																																																																																																																																																																									
Biltype																																																																																																																																																																									
Udleveringssted																																																																																																																																																																									
Sidste rengøring																																																																																																																																																																									
Sidste service																																																																																																																																																																									
Bil D																																																																																																																																																																									
Biltype																																																																																																																																																																									
Udleveringssted																																																																																																																																																																									
Sidste rengøring																																																																																																																																																																									
Sidste service																																																																																																																																																																									
Aktør:	Administrator																																																																																																																																																																								
Beskrivelse:	<p>Administrator skal reservere en bil til enten serviceeftersyn eller rengøring og får derfor først oplyst samtlige biler i delebilklubben, med deres tilhørende reservationer af forskellig type.</p> <p>Administratoren har derefter mulighed for at begrænse visningen af biler ved at angive biltype eller udleveringssted. Desuden kunne man forestille sig at det ville være praktisk at kunne sortere bilerne efter datoen for sidste rengøring eller for sidste serviceeftersyn.</p> <p>Når administratoren har besluttet sig for hvornår der skal foretages en vedligeholdelsesreservation angives periode samt den pågældende bil, og reservationen registreres idet der trykkes 'Reserver Delebil'.</p> <p>Herefter skal administratoren, såfremt vedligeholdelsesreservationen overlapper med nogle medlemsreservationer, have mulighed for at kontakte de berørte medlemmer og ændre på deres reservationer.</p>																																																																																																																																																																								
Data som hentes fra systemet ved oplystning af delebiler:	<p>De informationer som skal hentes for hver delebil, såfremt biltype og udleveringssted stemmer overens med de angivne værdier, er:</p> <ul style="list-style-type: none"> - Indregistreringsnummer. - Biltype. - Udleveringssted. - Dato for sidste rengøring. - Dato for sidste serviceeftersyn. - Medlemsreservationer inden for en angivet periode. - Servicereservationer inden for en angivet periode. - Rengøringsreservationer inden for en angivet periode. 																																																																																																																																																																								
Data som sendes til systemet ved registrering af reservation:	<p>De informationer administratoren sender til systemet ved oprettelse af en reservation er:</p> <ul style="list-style-type: none"> - Dato for reservationens start. - Dato for reservationens slutning. - Reservationstype: service eller rengøring. 																																																																																																																																																																								

Tabel B3.1: Beskrivelse af mockup for ”Vedligeholdelsesreservation (B14)”