
P2: Administration af skolebåde og bådudlån i en sejlklub.

- Programmering og modelløsning. -

Projektrapport
SW2A305

Caspar Rosgaard Kuchartik

Marc Tom Thorgersen

Thomas Pilgaard Nielsen

Nikolaj Møller Smed

Søren Hvidberg Frandsen

Tristan Carl Benjamin Bendixen

Troels Beck Krøgh



AALBORG UNIVERSITET
STUDENTERRAPPORT

Første Studieår — Software

Strandvejen 12-14

9000 Aalborg

<http://tnb.aau.dk>

Titel:

Administration af skolebåde og bådudlån i
en sejlkлуб.

Tema:

Programering og modelløsning.

Projektperiode:

P2 (Forårssemestret 2014)

03-02-2014 - 21-05-2014

Projektgruppe:

Gruppe SW2A305

Gruppemedlemmer:

Caspar Rosgaard Kuchartik

Marc Tom Thorgersen

Nikolaj Møller Smed

Søren Hvidberg Frandsen

Thomas Pilgaard Nielsen

Tristan Carl Benjamin Bendixen

Troels Beck Krøgh

Abstract:

Mangler / Missing

Vejleder:

Jacob Nørbjerg

Oplagstal: 1

Rapport sideantal: ???

Appendiks sideantal: ???

Total sideantal: 109

Projekt klaret den:

???

Ord/Tegn (Cirka): 13350 /68200

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Indholdsfortegnelse

Kapitel 1 Indledning	1
1.1 Initierende Problem	1
Kapitel 2 Fritidsklubber	3
2.1 Sportsklubber	3
2.2 Haller	4
2.3 Ungdomsklub	4
2.4 Sejlklub	4
2.5 Afgrænsning	4
Kapitel 3 Struktur af rapporten	7
3.1 Informationssystem	7
3.2 Rapportens opbygning	8
A Problemanalyse	9
Kapitel 4 Interessenter for en sejlklub	11
4.1 Interessenter for sejlklubbers administrative opgaver	11
Kapitel 5 Organisation	13
5.1 Medlemmer	13
5.2 Sejlerskole	13
5.3 Administration	14
5.4 Konklusion af organisationsanalyse	15
Kapitel 6 Teknologianalyse	17
6.1 Teknologier	17
6.2 State of the art	19
6.3 Konklusion af teknologianalyse	20
Kapitel 7 Problemformulering	21
7.1 Interessenterne for sejlklubber	21
7.2 Organisation	21
7.3 Teknologi	22
7.4 Afgrænsning for problemløsning	23
Kapitel 8 Produktkrav	25

8.1 Funktionelle krav	25
8.2 Datalogiske krav	25
B Problemløsning	27
Kapitel 9 Arbejdsprocess	29
9.1 Agile software development	29
9.2 Test driven development	29
Kapitel 10 Grafisk teori	31
10.1 Grafisk brugergrænseflade	31
10.2 Windows Presentation Foundation	32
Kapitel 11 Programopbygning	35
11.1 Klasser	35
11.2 Medlemsdata	36
Kapitel 12 Klasser	37
Kapitel 13 Database	43
13.1 Det “gamle” — Bare i tilfælde af, at noget af det skal bruges.	43
Kapitel 14 Programmets brugergrænseflade	45
14.1 Primære brugergrænseflade	45
14.2 UserControls	46
14.3 Vinduer	49
Kapitel 15 Program test	61
15.1 Funktionalitet	62
15.2 Effektivitet	62
15.3 Tilfredshed	63
Kapitel 16 Diskussion	65
Kapitel 17 Konklusion	67
Kapitel 18 Perspektivering	69
C Referencer	70
Figurer	71
Tabeller	71
D Appendiks	72
Bilag A Informationer påkrævet af sejlklubben Sundet	75

Bilag B	Screenshots af SailingClubManager	77
Bilag C	Interview	81
Bilag D	Interview transskribering	83
Bilag E	Interview Spørgsmål	89
Bilag F	UML-diagram	91
Bilag G	Use Cases	93
Bilag H	Bruger tests af program	95
Bilag I	Database	101

Indledning

1

Denne rapport er den skriftlige del af et projekt, udført af en gruppe studerende på Aalborg Universitet. Nærmere bestemt er det et P2 projekt på Softwareingeniørstudiet. Formålet med projektet er, ifølge projektoplægget, “[...] at opnå færdigheder i problemorienteret projektarbejde i en gruppe samt viden om sammenhænge mellem problemdefinition, modeldannelsers rolle i forståelse og konstruktion af programmer, og programmer som løsning på et problem i en problemstilling kontekst.” [?] Helt konkret skal der produceres et program i C#, der skal virke som en hel eller delvis løsning på det problem, opstillet i det valgte emne.

Projektets emne er “Administration af skolebåde og bådudlån i en sejlklub”, dette er et meget specifikt emne, derfor vil muligheden for at udvide problemstillingen til andre sports- og fritidsklubber undersøges.

I moderne tider er computere og smartphones i højere grad blevet en integreret del af størstedelen af personers hverdag. [?] Dog er det ikke altid, at foreninger og fritidsklubber anvender disse nye værktøjer på en effektiv og optimal måde. Dette er det problem, som gruppen vil undersøge, samt forsøge at løse.

Rapporten er opdelt i to hoveddele. Først undersøges problemet i problemanalysen og dernæst udarbejdes en løsning i løsningsdelen. Problemanalysen har til formål at opnå et overblik over hvilke interessenter der er, deres organisation og den teknologi der allerede findes indenfor det valgte emne. Dette ender ud i en problemformulering, som leder projektet ind i løsningsdelen. Her skal konstruktionen af selve programmet dokumenteres, arbejdsprocessen under arbejdet, forklaringer på valg foretaget i koden, programmets struktur og valg og overvejelser i forbindelse med brugergrænsefladen. Desuden vil der være en programspecifikation, som vil

FiXme Note! være de krav der er opstillet, på baggrund af analysen. ¹

1.1 Initierende Problem

Fritidsklubber benytter ofte frivillige i klubben til at hjælpe med administrativt arbejde. I en sejlklub er der flere logbøger over sejlture, som skal håndteres, eventuelt skal der også holdes styr på en sejl-skole, klubbens interne begivenheder og andre funktioner relateret til klubben. Hvis dette ikke organiseres effektivt kan det medføre, at arbejdet kan blive uoverskueligt, tager længere tid end nødvendigt og risikoen for at miste informationer bliver større.

¹FiXme Note: Her skal tilføjes lidt ang. hvad løsningsdelen kommer til helt konkret at indeholde. - Er blevet udbygget en lille smule nu; Caspar -

Denne problemstilling er ikke unikt knyttet til sejlkлубber, men et generelt dilemma med administrativt arbejde i fritidsklubber, specielt klubber der bruger samme faciliteter til både udløjning og undervisning, såsom sejlkлубbens fartøjer.

Rapportens analyse vil bygges ud fra følgende initierende problem:

Er det muligt at lave en softwareløsning, der kan gøre frivillige i fritidsklubbers administrative arbejde nemmere, som stadig er let at benytte uden nødvendigvis at have meget erfaring med anvendelse af computere, og i så fald hvordan?

I analysen vil disse problemstillinger blive undersøgt:

- Hvilke fritidsklubber har administrative opgaver, såsom udløjning, skemalægning af begivenheder mm., der skal håndteres af frivillig arbejdskraft?
- Hvilken information skal diverse fritidsklubber håndtere?
- Er det muligt at organisere de frivilliges arbejde med en softwareløsning?
- Findes der andre løsninger på markedet til at løse problemet? I så fald, hvorfor benyttes de ikke?

Fritidsklubber 2

I følgende afsnit vil der blive **kigget** nærmere på forskellige fritidsklubber, for at afdække de administrative poster som kunne forekomme. Følgende afsnit er skrevet ud fra gruppens egne erfaringer. Det er taget i betragtning, at disse erfaringer kan være specifikke for den enkelte klub hvor erfaringen stammer fra. Dog vurderes der at de forekommende administrative opgaver, er generelle opgaver, og altså ikke enkelttilfælde. Fritidsklubber dækker over klubber, hvor unge og voksne kan bruge deres fritid, om det er socialt, sportslig eller begge. Fritidsklubberne er valgt ud fra det faktum, at de ofte er drevet af frivilligt arbejde og har faciliteter der udlejes. Denne type frivilligt arbejde, er meget individuel fra klub til klub, om det er undervisning, håndtering af det økonomiske eller noget helt andet.

Dog er der nogle administrative opgaver, som går igen blandt fritidsklubberne. Det er primært opgaver, som omhandler medlems håndtering, da klubber har en medlemsbase, som skal administreres. De generelle administrative opgaver omkring medlems håndtering kan være følgende:

- Generel medlems håndtering
- Medlemsbetaling

2.1 Sportsklubber

I sportsklubber kan der forekomme mange forskellige administrative opgaver. Det er forskelligt fra sportsklub til sportsklub, hvilke opgaver der er tale om, men der er nogle fællestræk, som går igen blandt klubberne. Disse fællestræk er bl.a.:

- Udlejning af **materiel**, f.eks. baner, sportsudstyr o.l.
- Arrangering og pointgivning ved sportsarrangementer
- Arrangering af kørsel til og fra arrangementer

Sportsklubber kan også have specielle forhold, som spiller ind, f.eks. ved en skytteklub gælder der specielle regler ved våbenlicenser og våbentransport. Golfklubber har også specielle forhold omkring deres baner da de, rent arealmæssigt, er meget større end almindelige sportsbaner.

2.2 Haller

Sportshaller lægger lokale til mange forskellige sportsklubber. Fodbold bruger omklædningen og de udendørs baner til træning, badminton bruger de indendørs baner, som de skal dele med f.eks. håndbold, indendørs fodbold, indendørs hockey og mange flere. Sportshaller har også ofte lokale arrangementer, f.eks. foredrag, ungdomsklub osv. Haller har også ofte en kiosk eller café, hvor sportsudøvere eller arrangementsgæster kan komme og få noget at spise og drikke. Haller kan også have et motionscenter, hvor idrætsklubber og almindelige personer kan købe adgang[?]. Eksempler på administrative opgaver kunne være:

- Administration af haludlejning og omklædning
- Information fra kiosken
- Informationsdeling til sportsklubber og andre interesserede

2.3 Ungdomsklub

En ungdomsklub ses som **en** sted, hvor unge kan bruge deres fritid på sociale aktiviteter. Det er meget forskelligt, hvordan ungdomsklubber drives. Nogle drives af frivillige, hvor andre drives af skolevæsenet[?], f.eks. ungdomsskoler. Ofte holder ungdomsklubber til i skolebygninger. Hvad end de drives af frivillige eller skolevæsenet, har de stadig nogle administrative opgaver som skal løses. Eksempler på sådanne administrative opgaver kunne være:

- Arrangementsplanlægning
- Bestilling af varer (hvis der sælges sodavand, slik o.l.)

2.4 Sejlklub

En sejlklub kan, alt efter størrelsen, have flere administrative opgaver. Disse opgaver kunne bl.a. bestå af administration af deres både, hvem har reserveret en båd, hvornår er båden ledig osv. Nogle sejlklubber har også sejlundervisning, hvor der bl.a. skal kunne holdes styr på elevers niveau. Andre eksempler på administrative opgaver i en sejlklub kunne være:

- Af- og tilmelding af undervisning
- Reservering og udlån af både
- Oversigt over hvem der skylder penge for lån af båd

2.5 Afgrænsning

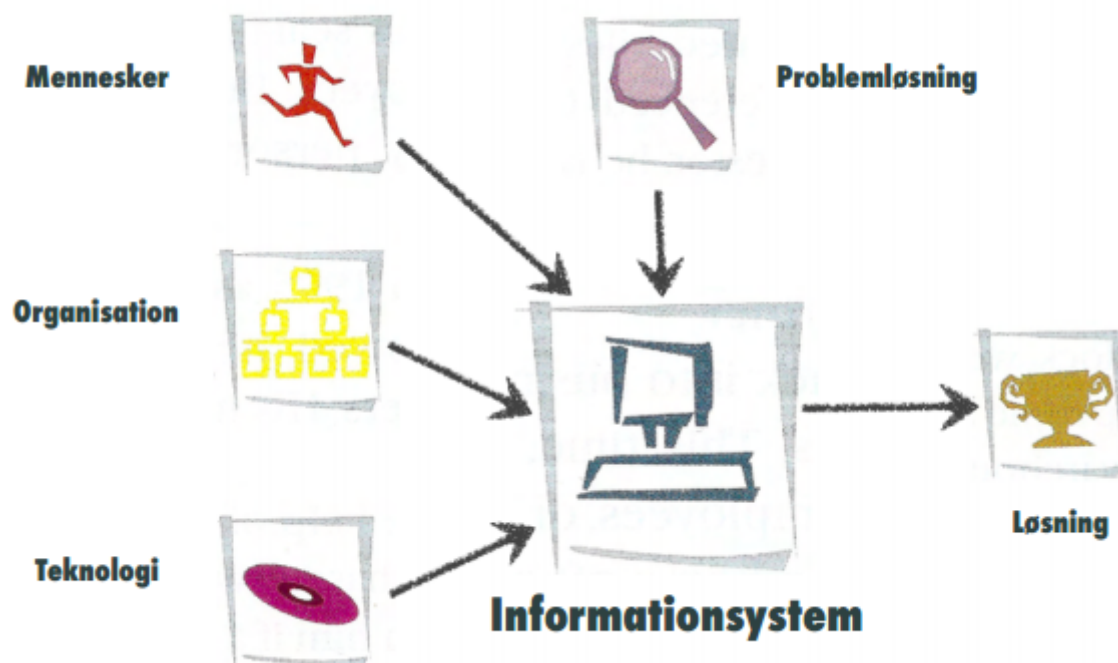
Ud fra de fundne fritidsklubber kan man se, at der er nogle træk som går igen. Dette er bl.a. **medlemshåndtering og medlemsbetaling, som stort set er en generel administrativ opgave i alle ovenstående fritidsklubber.** Udlån/udlejning af **materiel** er også en generelt administrativ opgave som går igen. Nogle klubber har også specielle administrative poster, som f.eks. skytteklubber, der har med våbenhåndtering af gøre.

Ud fra ovenstående fritidsklubber afgrænses der til udelukkende at beskæftiges med sejlklubber. Denne afgrænsning finder sted, da det vurderes, at der er et potentiale til at gøre arbejdet i

en sejlkub nemmere og ligeledes findes sejklubber specielle, da nogle sejklubber har en undervisningsfunktion, og derfor har specielle administrative funktioner.

Struktur af rapporten 3

I dette afsnit, beskrives strukturen af rapporten. Strukturen er baseret på beskrivelsen af et informationssystem fra ?. Komponenterne i et informationssystem er illustreret i figur 3.1.



Figur 3.1. Illustration af elementerne i et informationssystem. Kilde: ?

3.1 Informationssystem

Et informationssystem bruges til at effektivisere en arbejdsproces og hjælper med at holde fokus, så arbejdet bliver gjort tilfredsstillende. For at udvikle et godt informationssystem, skal man indsamle viden om mennesker, organisation og teknologi. Når informationssystemet er udviklet består de af tre processer: Indsamling af data, behandling af dataene og formidling af dataene. De tre emner bruges til at finde ud af, hvad der skal tages hensyn til under problemløsningsdelen, og vil blive beskrevet nedenunder.

3.1.1 Mennesker

Elementet *mennesker* handler om personer/persongrupper, som har en interesse i, at en given problemstilling løses. Det kan være brugeren af det program, der bliver lavet, samt andre, som får gavn af en løsning. Man undersøger bl.a. brugerens evner, da programmet skal laves på en sådan måde, at brugeren har den fornødne kunnen, til at kunne betjene programmet. Brugerens behov undersøges også, så man får alle de funktioner med, som er nødvendige for at programmet er brugbart.

3.1.1.1 Organisation

Under organisationsafsnittet undersøges hvor og hvordan problemet opstår, derudover undersøges der hvilke regler og værdier organisationen har, for at kunne tage disse med til problemløsningen.

3.1.2 Teknologi

Teknologidelen omhandler de teknologier, som anvendes til at løse en informationssystemsrelevant problemstilling. Ofte vil denne teknologi være en computer, en server og/eller internet. Dette er altså de byggeblokke hvorpå systemet bygges.

3.2 Rapportens opbygning

Først bliver menneskedelen behandlet, i form af en interessentanalyse, herefter vil organisationselementet blive berørt. Derefter bliver der skrevet om teknologier, hvorefter de tre elementer munder ud i en problemafgrænsning og en problemformulering. Til sidst skrives der om problemløsningsdelen.¹

FiXme Note!

¹FiXme Note: Sidste sætning skal udvides så der står hvad der vil blive skrevet om - Er det nødvendigt når det er nævnt i indledningen?

Del A

Problemanalyse

Interessenter for en sejlkлуб 4

I denne interessentanalyse ses der på mennesker og grupper, der har interesse i en specifik softwareløsning, konkret til sejlkлубber med sejl-skole. Der vil herudover blive set på, hvad deres interesse er i projektet, samt hvilken indflydelse en løsning ville kunne have på deres tilværelse.

Der vil for interessenterne vedkommende blive set på, hvad de hver især kunne få gavn af i forhold til projektet. Det er værd at notere, at hver enkelt klub har forskellige behov, så dette er en generel forståelse af interessenterne behov. Forståelsen for de forskellige grupper af interessenter er dannet ud fra et interview med den tidligere skolechef fra sejlkлубben Sundet. Interviewet kan findes i appendiks C på side 81.

4.1 Interessenter for sejlkлубbers administrative opgaver

4.1.1 Dansk Sejlunion

Dansk Sejlunion er et forbund, som blev dannet i 1913. Deres mission er at være det nationale samlingspunkt for alle sejlere. Dansk Sejlunion er tilsluttet Danmarks Idrætsforbund, International Sailing Federation og andre lignende organisationer inden for sejlsport [?]. Et af deres mål er, som forbund, at hjælpe sejlkлубber med service, rådgivning mm. og derfor mener det, at Dansk Sejlunion også vil være interesseret i et system, der vil kunne hjælpe de frivillige i deres arbejde. De vil ikke komme til direkte at anvende informationssystemet, men vil være stadig relevante at tage med i projektets overvejelser.

4.1.2 Medlemmer

Hver sejlkлуб bestemmer selv deres organisatoriske opbygning af, hvilke type medlemmer de har i klubben. Dette vil blive undersøgt nærmere i kapitel 5 på side 13.

Medlemmerne i sejlkлубben er de personer, som betaler kontingent og bruger sejlkлубbens faciliteter. Da medlemmerne betaler kontingent er det også her klubben får deres indtægter til at kunne holde sig i gang, og investere i eventuelt nye både, begivenheder el. Det er derfor vigtigt, at klubberne gør medlemmernes tid ved sejlkлубben så god som mulig. Et system der kunne hjælpe medlemmerne med at finde informationer, leje både og som har medlemmernes

interesse i højsædet, vil være i deres interesse. De førnævnte funktioner i et system til en sejlkлуб, er netop funktioner, der sandsynligvis ville blive brugt mest af medlemmer.

Medlemmer i en sejlkлуб kan være en meget bred gruppe fra den yngre befolkningsgruppe, til den noget ældre. Det er derfor vigtigt at lave et design, som er intuitivt i anvendelse, for alle brugere.

4.1.3 Undervisere

En sejlkлуб med sejlerskole skal have undervisere til sejlerskolen, og disse kan også drage nytte af et system til at hjælpe med administrative opgaver. Underviserne vil ved hjælp af et system, eventuelt kunne sende afbud via e-mails eller SMS-beskeder ud til alle på undervisningsholdet. Det vil muligvis kunne gøre det lettere for underviseren at holde styr på den enkelte elevs fremskridt ved undervisningen mm., hvilket medfører at underviserne bør have andre funktioner i et system til en sejlkлуб end medlemmerne ville have. Underviserne ville eventuelt kunne stå for at registrere hvem der har et førerbevis i klubben, ol. Underviserne kan også være frivillige i sejlkлубben, men de er beskrevet for sig selv, da de har mere specifikke opgaver i sejlkлубben. Det antages, at hvis et system ville kunne hjælpe undervisere med deres opgaver, så de bruger mere af deres arbejdstid på at undervise, ville det give en positiv respons.

4.1.4 De frivillige i sejlkлубben

Der er, udover undervisere i en sejlkлуб, også andre frivillige der arbejder i sejlkлубben. Dette kan være lige fra en formand, til en sekretær eller måske endda nogen som gør rent i sejlkлубben. Disse frivillige kan alle have forskellig gavn af et system, der kan håndtere deres anliggender i sejlkлубben, og er derfor taget med som interessenter. Nogle af dem kan have til opgave at holde styr på ind- og udbetalinger for medlemmerne, hvilket et medlemssystem ville kunne hjælpe med. En sekretær ville kunne give nyheder eller referater videre til klubbens medlemmer igennem sådan et system. Så der er mange forskellige muligheder et administrativt system ved en sejlkлуб ville kunne hjælpe med, og herudover også mange mennesker der ville kunne få gavn af en løsning.

4.1.5 Konklusion af interessentanalyse

Disse fire interessenter, har hver deres interesse for projektet, hvor underviserne og de andre frivillige smelter en smule sammen. Underviserne, samt de andre frivillige, vil gerne have det gjort lettere at udføre deres frivillige arbejde. Medlemmerne vil gerne have informationer fra klubberne angående begivenheder, der måtte foregå, samt muligheder for at deltage i disse, eller leje sejlbåde. Dansk Sejlunion er en mindre direkte interessent, men kan dog have interesse i at hjælpe nye eller mindre sejlkлубber med at opsætte et godt administrativt netværk, for at danne en velfungerende sejlkлуб, der giver gode oplevelser for medlemmerne.

Organisation 5

I dette kapitel vil der blive set på sejlklubber som organisation. Der vil blive set på, hvordan medlemmer kategoriseres, undervisningen organiseres og hvordan sejlklubber er organisatorisk opbygget. Under den organisatoriske opbygning kigges der på hvordan klubberne bestyres, hvordan klubbernes data håndteres samt hvilke regler de enkelte klubber har. Undersøgelsen omhandler tre forskellige klubber: Sejlklubben Sundet beliggende i København, Aalborg Sejlklub som har til huse i Aalborg og Bådklubben Valby som også er beliggende i København. Der vil primært blive fokuseret på Sejlklubben Sundet, da der er bedst information til rådighed om dem. Dette i form af et interview som blev nævnt i kapitel 4 på side 11, samt en mere informativ hjemmeside, i forhold til de to andre klubber. Aalborg sejlklub og Bådklubben Valby er med i undersøgelsen for sammenligningens skyld, og har ikke været tilgængelige for interviews.

5.1 Medlemmer

Det er meget forskelligt fra sejlklub til sejlklub hvordan medlemmer kategoriseres, hvis dette bliver gjort. Sejlklubben Sundet kategoriserer deres medlemmer således: voksen-, bådejer-, gaste-, mini-kølbåd-, ungdoms-, passiv- og støttemedlem. Desuden har de en særlig æresmedlemskategori. Denne status opnås ved at have ydet en mangeårig, ekstraordinær indsats for klubben og sejlsporten generelt. [?].

Til sammenligning har Aalborg sejlklub kategoriseret deres medlemmer som følger: aktive, passive, junior- og æresmedlemmer [?].

Bådklubben Valby kategoriserer som A-, B- og C-medlemmer, som indikerer hvilken bådtype de har [?].

Det formodes på baggrund af appendiks D på side 83, at fælles for medlemmerne er, at de gerne vil have informationer fra klubberne, så de ved hvilke begivenheder der finder sted. Det kan være kapsejladser, foredrag o.l.

5.2 Sejlerskole

Sejlklubben Sundet og Aalborg sejlklub har en sejlerskole, hvor man kan opnå et førerbevis til båd. Bådklubben Valby adskiller sig fra de to andre, ved ikke at have en decideret sejlerskole, men de har kurser man kan tage hen over vintersæsonen, for at opnå et duelighedsbevis [?].

I Sejlklubben Sundets sejlerekole er der undervisning i hverdagene fra kl. 18 til 21, i månederne maj, juni, august og september. Uddannelsen varer to år, og man lærer at sejle i bådtyperne **Drabant 24** og Gaffelrigger. I løbet af en sæson sejler man typisk op til 18 gange. Er man forhindret i at møde op, skal man ringe og melde afbud, hvis man ikke møder op tre gange, uden at have meldt afbud, mister man sin plads på sejlerekolen. Undervisningsforløbet afsluttes med en praktisk prøve, hvorpå man opnår sit førerbevis, i så fald man består [?]. Aalborg sejlklubs sejlerekole minder meget om Sejlklubben Sundets.

5.3 Administration

Den organisatoriske opbygning i de tre klubber er meget ens, altså bestyrelsessammensætningen og hvordan denne vælges. Herunder vil Sejlklubben Sundets organisatoriske opbygning blive beskrevet, samt nogle administrative opgaver.

5.3.1 Organisatoriske opbygning

Sejlklubben Sundet vælger sin ledelse ved generalforsamlinger, her bliver der også valgt, regnskabsfører, øvrige bestyrelsesmedlemmer, samt en revisor og revisorsuppleant. Kun æresmedlemmer eller aktive medlemmer over 18 år har stemmeret ved generalforsamlingen. Ved aktive medlemmer forstås der, medlemmer som ikke er passive eller støttemedlemmer. Stemmeretten udøves ved personligt fremmøde til forsamlingen. Medlemmerne kan senest seks uger før generalforsamlinger sende forslag til diskussioner, såsom vedtægtsændringer, eksklusioner af medlemmer o.l. Det er altså demokratiske valg, der håndterer større beslutninger i klubben.

5.3.2 Bådudlån og andre administrative opgaver

Man kan i sejlklubben Sundet, låne både til sejlads for et mindre beløb, når de ikke er i brug til undervisning. Prisen for leje af både er højere i weekenderne end i hverdagene, men hvis en af skolens elever er med på sejladsen, er det gratis at leje en båd. For at kunne leje en båd, skal der være mindst en bådfører med, som man bliver ved at fuldføre to års undervisning, som nævnt i afsnit 5.2 på foregående side. Hver gang der skal sejles ved sejlklubben Sundet, er der meget data der skal skrives ned, af forskellige årsager. Det er vigtigt at vide, hvem der er med på båden både for at sikre sig, at besætningen er i stand til at føre båden, men også af sikkerhedsmæssige årsager, hvis der skulle ske en ulykke. En liste, over hvilke data der skal skrives ned, er udformet og kan findes i appendiks A på side 75. Ved et kig på Sejlklubben Sundets hjemmeside [?] ses det, at forskellige værktøjer, bl.a. en doodle, er taget i brug, i et forsøg på at øge brugervenlighed og interaktion mellem deltagere. Som Jacob Nørbjerg fortalte i interviewet (se appendiks C på side 81), skulle man før i tiden ned i klubhuset, for at se om der var en ledig båd man kunne reservere, men dette kan altså nu gøres gennem doodlen. Jacob Nørbjerg giver også udtryk for i interviewet, at det er svært, at få viden omkring klubbens foretagende hjemmefra. Nuværende praksis er, at man skal ned i klubben for at se og melde sig på diverse begivenheder, såsom 24-timers sejlads, eller for at se om der er en ledig plads, på et sejlads hold.

Håndtering af information omkring klubbens medlemmer foregår på en lokal computer i Sejlklubben Sundets klubhus. Endvidere bruges denne computer til at udprinte girokort til

betaling af bådlån. Da informationerne er gemt lokalt, betyder det, at de frivillige, som er ansvarlige for udprintningen af girokort, skal ned i klubhuset for at udføre denne opgave.

5.4 Konklusion af organisationsanalyse

Sejlkлубben Sundet har mange administrative opgaver som håndteres manuelt og til dels på en ustruktureret måde. Som Jacob Nørbjerg giver udtryk for i interviewet, er det svært at holde sig opdateret med hvad der sker i klubben. Ud fra dette kan der konkluderes at en softwareløsning, hvor der implementeres en kalender, som kan holde styr på alle begivenhederne, vil være et godt supplement, til administrationen af klubben. For at give eleverne bedre mulighed for at kunne af- og tilmelde sig, så de har mindre risiko for at miste deres plads i sejl-skolen, kunne der med fordel implementeres en sådan funktion i en softwareløsning. Efter egen erfaring, virker doodlen til udlån af både ustruktureret og rodet. Derfor vil en forbedret udlånsfunktion med fordel kunne udvikles.

Da en softwareløsning, som ikke har forbindelse til nettet, blot vil være et system ligesom det gamle, hvor man skulle ned i klubben for at låne både og holde øje med begivenheder, tilstræbes det at udvikle et web implementeret system.

Da der har været flest informationer til rådighed for Sejlkлубben Sundet, som dermed også har haft bedst konkretiserede krav til funktioner, vil der fremover blive fokuseret på at lave en løsning til netop dem. Dog skal der siges, at systemet der udvikles til Sejlkлубben Sundet, let vil kunne bruges af de andre sejlkлубber, enten ved at ændre minimalt, eller at helt slette funktioner.

Teknologianalyse 6

I det følgende kapitel vil forskellige teknologier, som kunne anvendes i forhold til at hjælpe med administrationen i en sejlklub, blive undersøgt. Derefter findes en analyse af de programmer, som på nuværende tidspunkt er state of the art inden for administration af sejlklubber.

6.1 Teknologier

6.1.1 Internet

En fordel for et system til administration i alle typer af klubber og organisationer ville være at kunne tilgå det fra flere steder end bare et sted. Ved at gøre et system netværksbaseret kunne medlemmer og frivillige tilgå det fra hjemmet, deres arbejdspladser eller måske endda deres mobile enheder, og derved ville en tur ned til klubhuset, f.eks. for at undersøge om der var en båd ledig den kommende uge, undgås, hvilket der i afsnit 5.4 på side 15 blev fremlagt som et problem. Anvendelsen af et system, som kunne operere over internettet, kunne altså øge informationsspredning i klubben. Herved gøre det muligt at lave papirarbejde, som udfyldelse af log-filer hjemmefra i stedet for i klubben og tilmelde sig sejlads og andre arrangementer hjemmefra.

6.1.2 Server

For at alle medlemmer og frivillige arbejdere i en klub kan tilgå systemet over internettet, skal systemet opretholdes af en server, hvorfra andre computere kan forbinde sig til og bruge systemet. En mulighed for opsætning af en server ville være at få en computer i selve klubhuset til at agere server og have systemet, og al information, til at ligge der. Dette er dog en dårlig løsning, da klubben selv skulle stå for opdatering af serversoftware, tage backup af al data og det ville kræve mere af klubbens internetforbindelse. En anden mulighed for at en klub ikke selv skulle stå med at administrere en server, er at købe sig ind hos et hosting firma. Her kan man have systemet kørende på en server, og firmaet varetager samtidigt al vedligeholdelse af server og serversoftware og sørger for backup af al data. Dog kan forskellige firmaer have forskellige måder at gøre tingene på, men de firmaer der kigges på i de følgende afsnit, *abakomp* og *Softcom*, sørger begge for vedligeholdelse og overvågning af serverne, og for at tage backup.

De typer af servere, der bliver udbudt af hosting firmaer, kan deles op i dedikerede servere og i virtuelle servere.

6.1.2.1 Dedikerede servere

Ved en dedikeret server lejer man hele maskinen, som skal køre systemet, og man får derfor mere ud af maskinen og har mere indflydelse på komponenterne i maskinen og hvilket operativsystem der køres. En dedikeret server ville umiddelbart være bedre for større klubber/organisationer, hvor mængden af informationer serveren skulle kunne håndtere er større og trafikken til serveren også ville være højere. [?]

6.1.2.2 Virtuelle servere

Alternativet til at leje en hel maskine til at køre et system, er at leje en virtuel server, hvor der på den samme fysiske maskine findes flere separate virtuelle servere. At leje en virtuel serverplads er billigere end at leje en dedikeret server, da en virtuel serverplads kan lejes for helt ned til 850 kr. i måneden[?], hvorimod en dedikeret server kan lejes fra 1062 kr. i måneden [?]. Dog variere prisen alt efter hvilket hardware der bruges og efter, hvor meget serverplads man ønsker. Grunden til at en virtuel serverplads er billigere, er at flere forskellige virtuelle servere kan køre på samme fysiske maskine, og derved er der flere til at betale for vedligeholdelsen. Men en virtuel server har til gengæld ikke lige så meget plads og kraft som en dedikeret server.

En virtuel serverplads ville umiddelbart være nok til at køre et administrativt system for en sejlklub, og det ville desuden være billigt at leje plads hos et hosting firma. Det er derfor fornuftigt at antage at sejlklubber har midler til at leje en serverplads, hvilket ville kunne anvendes til at køre et administrativt system med alle informationer, som kunne være nødvendigt for systemet.

6.1.3 Brugergrænseflade

I et program som skal anvendes af en bruger eller brugergruppe, ønskes det, at programmet har et udseende, som brugeren forstår og som hjælper med at bruge programmet. Her menes der programmets interface, på dansk, brugergrænsefladen. Det er vigtigt, at brugergrænsefladen henvender sig til modtagergruppen, hvilket i dette projekt er personer med tilknytning til en sejlklub. Der skal desuden tages højde for de forskelle i teknisk snilde, der er i modtagergruppen, da det ikke er effektivt at lave et system, som halvdelen af modtagerne ikke kan bruge. Forskellige undergrupper i modtagergruppen skal bruge forskellige funktioner i systemet, og derfor skal interfacet tilpasses, så de korrekte funktioner bliver tilgængelige for de rigtige brugere.

En mulighed for at skabe en brugergrænseflade i samarbejde med C#-kode, er ved at anvende *Windows Presentation Foundation*. Det findes også andre muligheder for at anvende C#-kode til at skabe brugergrænseflader, men grundet begrænset tid og viden vil *Windows Presentation Foundation* blive anvendt til at skabe brugergrænsefladen i dette projekt.

6.1.4 Management systemer

Et system som med fordel kunne anvendes til administrere en fritidsklub, som en sejlklub, er et management system. Et management system er et framework af processer og procedurer, som bruges til administration af frivillige organisationer og virksomheder. Eksempler på management systemer findes i kapitel 6 på foregående side. Mere præcist er SailingClubManager et management system, hvilket er yderligere dokumenteret i appendiks B på side 77. Det ma-

nagement system har en grafisk brugergrænseflade (GUI) i web browseren. Dette er i kontrast til en konsol applikation, eller en nativ GUI.

Ved at anvende et management system kan en klub eller forening, forudsat at systemet er succesfuldt, mindske den tid, der bruges på papirarbejde. Ifølge [?] er 26% af de arbejdsopgaver, som findes for frivillige, sekretariatsarbejde og administrativt arbejde. Med et godt management system kan de administrative opgaver derved bringes ned på mindre tid, og derved kan de frivillige bruge deres tid på mere interessante ting for organisationen, eller bruge den ekstra tid på noget andet.

6.2 State of the art

For at kunne udvikle et godt produkt, der skal kunne bruges i en bådklub, er det vigtigt, at se på hvilke produkter der allerede er på markedet, altså hvad er “state of the art”. I denne forbindelse er der fundet forskellige produkter, som har nogle af de features der efterspørges i et system til en bådklub.

BoatCloud

Der er f.eks. et program udviklet af Anderson Software, der hedder BoatCloud.[?] BoatCloud består af 3 applikationer, StackTrack, VesselValet og Service Request. StackTrack benyttes, når medlemmerne selv sejler deres både, hvorimod VesselValet, er når der skal tjenere og passagerer med ombord på bådene. BoatCloud applikationerne er derfor mere designet til en bådklub, som passer på medlemmernes både, fremfor klubbens egne både. I applikationerne kan medlemmerne melde, at de vil sejle på et bestemt tidspunkt. Medlemmet kan få klubben til at vaske båden, tanke den, og fylde den op med diverse snacks, alt sammen registreres igennem denne webbaserede applikation. Applikationen tager imod alle disse bestillinger i realtime, og 24 timer i døgnet. Herudover bliver der sendt e-mails ud til medlemmerne, når de har lavet en bestilling. Man kan logge ind som administrator, og her kan man se alle reservationer, der er lavet, samt have mulighed for at se yderligere detaljer om hver enkelt reservation.

Sailing Club Manager

En anden applikation der findes er Sailing Club Manager [?]. Denne applikation er også web-baseret, og gør det muligt for en bådklub at tilføje hvilke både der er, samt hvor det er fortojet i havnen. Man kan i en kalender lave begivenheder og reservationer, samt medlemstilmeldelse til forskellige begivenheder. Applikationen kan også bruges som kontaktmedium for klubben til deres medlemmer. Klubben kan sætte et e-mail system op, samt tilføje et template som bliver sendt med hver enkelt e-mail. Applikationen kan endvidere holde styr på økonomien, man kan tilføje en bankkonto, samt holde styr på fakturaer, og endda sende dem og håndtere det online. Man kan tilføje medlemsskaber, med forskellige oplysninger omkring medlemmerne der har netop dette medlemsskab i klubben.

Ud fra denne undersøgelse er flg. features altså fundet i applikationerne:

- Medlemmer kan reservere afgang
- Medlemmer kan melde sig på begivenheder i bådklubben

- Medlemmer kan betale igennem hjemmesiden
- Medlemmerne kan bede klubben om at gøre deres egen båd klar, med forskellige aftaler klubben håndterer før den aftalte tid
- Bådklubben kan vise hvilke både der er i klubben
- Bådklubben kan oprette begivenheder, og afsætte hvilke medlemmer der kan melde sig på begivenheden
- Bådklubben kan oprette forskellige medlemsskaber og opkræve betalinger igennem hjemmesiden
- Bådklubben kan sende e-mails med klubbens egen template gennem hjemmesiden, herunder påmindelser om en reservation nogen tid før.

I bilag appendiks B på side 77 findes der screenshots af udvalgte features der findes i SailingClubManger.

ForeningLet

ForeningLet er en applikation, som adskiller sig fra de andre ved ikke at henvende sig specifikt til én type klub, desuden er det en dansk applikation. Applikationen er web-baseret og har mange nyttige funktioner for en sejlkлуб som Sundet. Applikationen har en medlemsdatabase, et regnskabssystem samt en kontaktfunktion som kan benytte både e-mail og SMS. Der kan også opkræves betalinger fra medlemmerne gennem systemet. Applikationen har også en reservationsfunktioner. Desuden kan der oprettes begivenheder i applikationen, som medlemmerne også kan tilmelde sig gennem applikationen. ForeningLet giver også foreningen mulighed for at oprette deres egen hjemmeside samt deres egen online butik.

6.3 Konklusion af teknologianalyse

6.3.1 Konklusion for teknologianalyse

Ud fra teknologianalysen er de teknologiske aspekter i projektet blevet belyst. Muligheden for at gøre systemet netværksbaseret, gør det muligt for medlemmerne og administrationen i en sejlkлуб, at foretage handlinger vedrørende klubben fra hjemmet eller arbejdspladsen. For at undgå at en lokal server skulle opsættes og vedligeholdes af klubben selv, kunne en virtuel server, hostet af et hosting firma, varetage denne opgave. Vigtigheden i en god brugergrænseflade blev gjort klart, og metode for at opsætte en brugergrænseflade blev valgt i form af *Windows Presentation Foundation*. Fra state of the art delen blev nuværende systemer på markedet undersøgt, og udvalgte features tages med videre til løsningsdelen, som eventuelle tilføjelser til det administrationssystem der skal udvikles.

Problemformulering 7

Indtil nu er problemet om fritidsklubbers administrative opgaver blevet belyst og analyseret vha. Laudon og Laudons model beskrevet i kapitel 3 på side 7. Det er nu ud fra denne analyse muligt at komme frem til en problemformulering, som vil hjælpe til at finde en løsning for fritidsklubber og ikke mindst sejklubberne.

Der blev i kapitel 2 på side 3, omhandlende fritidsklubber, afgrænset til sejklubber, da disse viste sig at have flere specifikke administrative opgaver, sammenlignet med andre fritidsklubber. Denne afgrænsning blev foretaget da et system der kan håndtere generelle, samt specifikke opgaver til en sejlklub, kan bruges af andre former for fritidsklubber ved at tilføje, fjerne og modificere specifikke funktioner. Derfor ved at analysere videre på de direkte forhold vedrørende en sejlklub, ville der kunne dannes en forståelse for, hvilke problemer en sejlklub med sejlere har, som kan løses vha. et administrativt system.

7.1 Interessenterne for sejklubber

Der var forskellige interessenter for sejklubberne, og de havde varierende mængder af interesse i projektet. Dansk Sejlunion er den som skiller sig mest ud. De kan have interesse i projektet for at kunne hjælpe klubber med at få et godt administrativt system, så klubberne kan give deres medlemmer en bedre oplevelse og service.

Medlemmerne, underviserne og de frivillige vil alle være personer der skal bruge det udviklede system. De har derfor en mere direkte indvirkning på, hvordan system skal opbygges. Hvis systemet ikke har de funktioner som de efterspørger, vil det ikke være den gode løsning, som de gerne vil have. Dette gælder for alle de tre nævnte interessenter. Det skal dog understreges, at de ikke vil bruge samme dele af systemet. Dette skyldes, at medlemmerne ikke skal kunne oprette undervisningsdage mm., det er kun underviserne. Dette er bl.a. en af grundene til, at de er delt op i de tre forskellige grupper.

7.2 Organisation

I forbindelse med organisationsafsnittet blev der undersøgt, hvordan sejklubberne håndterer forskellige opgaver i klubben, samt hvilke opgaver de beskæftiger sig med.

Det viste sig, at klubberne har individuelle medlemstyper, og det kan derfor være relevant at lade klubberne selv oprette forskellige medlemstyper i systemet. Desuden efterspørges

det, at man kan tilkoble sig systemet hjemmefra, for således at kunne få informationer om begivenheder og undervisning i klubben, og måske endda tilmelde sig disse, uden at tage turen ned til klubben.

Grundet den store mængde af information, der skal nedskrives, i forbindelse med en sejlads, giver det mening at hjælpe med at organisere denne opgave, samt at gøre det lettere at registrere informationerne for diverse frivillige og undervisere. Desuden kunne det hjælpe hvis hvert medlem havde en saldo over udgifter ved sejlklubben, således det er nemmere at håndtere brugerbetaling.

Følgende er altså en liste over emner som kan dækkes af et system for sejlklubberne:

- Tilkobling hjemmefra, via internettet.
- Mulighed for at få informationer vedr. begivenheder, samt at tilmelde sig disse.
- Organisering af informationer der foretages i form af sejlads.
- Organisering af betalinger, samlet for det enkelte medlem, samt mulighed for online betaling.

Man har fundet frem til, at systemet, der efterspørges, er et af typen management systemer, som blev beskrevet i afsnit 6.1.4 på side 18. For at kunne implementere disse funktioner til et management system, skal der være forskellige teknologier til rådighed. Disse blev beskrevet i kapitel 6 på side 17.

7.3 Teknologi

For at kunne tilkoble sig systemet igennem internettet, skal man have opsat en server til systemet. Serveren bør ikke være lokaliseret i klubhuset, men man kan købe sig til forskellige servere i datacentre rundt om i verden. For at systemet skal kunne fungere optimalt for de frivillige, er det stadig vigtigt med en computer i klubhuset, og der skal derfor være en internetopkobling i klubhuset.

Foruden en serveropsætning er det vigtigt at se på designet af systemet, for at de frivillige kan udnytte det optimalt. Her tænkes der på brugergrænsefladen. Som nævnt i kapitel 4 på side 11 er det en blandet gruppe af mennesker, der kan være frivillige i en sejlklub. Derfor bør brugergrænsefladen være nem at finde rundt i, således de fleste vil kunne bruge systemet uden større problemer. Dette forventes at blive gjort ved brugen af Windows Presentation Foundation.¹

FiXme Note!

Sejlklubben Sundet, viste sig at have mange flere forhold at organisere end de andre klubber der er undersøgt, og derfor afgrænses projektet til konkret at designe et IT-system til Sundet. Dette gøres da hvis et system kan hjælpe Sundet, vil der på baggrund af analysen kunne konkluderes, at det også kan hjælpe andre sejlklubber, med færre forhold at holde styr på.²

FiXme Note!

¹FiXme Note: Skal dette være der inden diskussionen mellem WPF, WinForms og Webpage?

²FiXme Note: Kan ikke helt finde ud af starten af sætning

Ud fra denne afgrænsning er flg. problemformulering formuleret:

Det er et problem at frivillige i fritidsklubber med specielle udlejningsmuligheder, så som Sundet, benytter unødvendig arbejdskraft på fysisk dokumenthåndtering af faciliteternes udlån, undervisning og begivenhedsorganisation. Hvordan kan et system hjælpe med at danne overblik over sådanne opgaver?

7.4 Afgrænsning for problemløsning

Da dette projekt er foretaget i løbet af forårssemestret på Aalborg Universitet, er der ikke uanede mængder af tid. Der afgrænses derfor fra at udvikle hele management systemet til sejlklubben Sundet, til i stedet at lave enkelte dele af systemet.

Grundet de manglende ressourcer og tid, vil der altså ikke blive lavet et system, der kører over internettet, men i stedet et system, som kan håndtere de forskellige emner lokalt. Denne afgrænsning finder sted, **da** hvis funktionerne for klubben kan fungerer på computeren, skal **FiXme Note!** det blot tilkobles en server, for dermed at kunne virke på flere forskellige computere.³ Derfor ser projektgruppen altså funktionerne for sejlklubben som værende de vigtige ting at udvikle, selvom systemet mister meget af sin effektivitet uden internetopkoblingen. Men dette ses som værende okay, da systemet ikke skal tages i brug af klubben, da dette er et projekt lavet i forbindelse med uddannelsen på Softwareingeniørstudiet ved Aalborg Universitet.

Hvis tiden er til det i slutningen af projektet, vil der muligvis ændres i systemet så det virker igennem en internetopkobling.⁴ **FiXme Note!**

De resterende kapitler i rapporten, vil beskæftige sig med udviklingen af et management system, henvendt imod sejlklubben Sundet, baseret på problemformuleringen. Som det første vil der komme en kravspecifikation for systemet, baseret på baggrund af denne afgrænsning.

³FiXme Note: Underlig sætning, er lidt i tvivl om hvad der menes

⁴FiXme Note: Skal sådanne gisninger omkring projektforløbet med?

Produktkrav 8

FiXme Note! I problemanalysen blev det gjort klart, at der findes et problem ved fritidsklubber, nærmere specificeret sejlskoler, når det kommer til management af deres ressourcer samt dokumenthåndtering. Problemet kan muligvis løses ved et informationssystem¹, til at håndtere papirarbejde og gøre det overskueligt, i dette afsnit vil det blive set på, hvilke krav et sådant system har.

8.1 Funktionelle krav

Papirarbejde indebærer logbøger, bådrreservationer, samt hvem der er ombord, tilmeldinger til begivenheder m.m. Som grundlag for at alt dette kan fungere, kræves der en metode til at sætte sig selv på en aktivitet. For at gøre dette muligt kræves der således to ting: En database af medlemmer, samt en form for begivenhedskalender. Foruden dette kræves der information om de forskellige begivenheder, en log over hvem der har betalt for at være med og eventuelt hvem der mødte op. Der kræves også et system til udlejning af både samt undervisning, hvor nævnte både også benyttes.

Produktet har krav til følgende elementer i sejlklubben:

- Medlemmer
- Login system
- Begivenhedsinformation
- Begivenhedskalender
- Bådrreservation
- Oversigt og kommunikation for undervisning

8.2 Datalogiske krav

(Der ligger mere bag de ovenstående elementer end afsnit 8.1 giver udtryk for, disse vil her blive set nærmere på fra en datalogisk synsvinkel.

¹FiXme Note: skal der ikke stå management system?

8.2.1 Medlemmer

For at kunne integrere medlemmer i programmet, skal der være en overordnet klasse, der danner base for alle medlemmer. Da der også kan komme gæster i klubben, er det relevant at lave en personklasse over dette. Yderligere specifikationer kan så laves igennem subklasser af personklassen og på denne måde få lavet forskellige instanser af f.eks. elever, undervisere, administratorer osv. Da Sundet allerede har en database med deres medlemmer, skal der tages højde for at importere dette data over i en anden database², så Sundet ville kunne have denne mulighed, hvis behovet skulle opstå. FiXme Note!

8.2.2 Login system

For at medlemmerne skal kunne få en personlig oplevelse i programmet, er det vigtigt at have et loginsystem, som vil være relevant ved f.eks. bådreservation og begivenheder. Hvis medlemmet skal kunne se hvilke begivenheder de deltager i, eller kunne reservere en båd, er det smart at reservationen kan kobles på objektet af medlemmet. Dette kan registreres på en nem måde, ved at medlemmet er logget ind på deres personlige bruger i systemet.

8.2.3 Begivenhedsinformation

Begivenhederne skal også gemmes i en database, således de kan tilgås og gemmes fra runtime til runtime af programmet. Derfor er det også et krav at håndtere objekter af begivenheder i en database. Her skal der være information omkring den pågældende begivenhed, såsom dato, start tidspunkt, og andre informationer relevante for medlemmernes deltagelse til begivenheden.

8.2.4 Begivenhedskalender

Når begivenhederne kan findes i databasen, skal den kunne forbinde dem med kalenderen, som også skal indbygges i programmet. Her skal det være muligt for programmet at vise, hvornår der sker begivenheder og opsætte begivenhederne i den rigtige kronologiske rækkefølge så det er nemt at finde rundt i for klubbens medlemmer.

8.2.5 Bådreservation

Der skal oprettes objekter af bådene, som findes i klubben og på denne måde reservere disse objekter, på den pågældende dag. Bådene, samt reservationerne, skal også gemmes i databasen, så de kan findes fra runtime, til runtime af programmet. Foruden dette skal bådene også have en log omkring de forskellige sejlads, der er blevet foretaget i bådene. Disse logs er blevet omtalt og uddybet i appendiks A på side 75

8.2.6 Oversigt og kommunikation for undervisning

Det skal være muligt for underviserne og eleverne at kunne holde styr på deres undervisningsdage, se hvilke typer undervisning de har haft hvornår, samt at se hvilke typer undervisning de mangler for at kunne bestå. Underviserne skal også i den forbindelse kunne se hvilke elever der er på de forskellige hold, og samtidig være i stand til at skrive hvem der er deltaget hvornår.

²FiXme Note: Menes der vores database eller hvad?

Del B

Problemløsning

Arbejdsproces 9

I dette afsnit vil udviklingsprocessen beskrives, hvilket metoder der er blevet benyttet, hvordan de er udført samt hvordan udviklingen er planlagt. Udviklingsmæssigt baserer vi vores proces på to primære begreber værende, agile software development og test driven development(TDD).

9.1 Agile software development

Agile software development benyttes som et grundlag for programmeringen af produktet. Her ligger der eksempelvis fokus på et virkende produkt frem for en teoretisk snak. Dette anvendes i udviklingen for at holde programmet realistisk, begynd ikke på noget der ikke er tid til eller kun kan laves delvist. Hellere have et mindre og godt program, end et større fejlfyldt program.

Agile software development ligger yderligere fokus på “co-location”, således at arbejde bliver udført på en arbejdsplads med ens kollegaer, medstuderende, snarere end en isoleret process, heri ligger “pair programming”. Denne metode er baseret på at to programmører programmerer på en enkelt computer, her med en fører og en observatør, for øget kodekvalitet. Denne metode er blevet delvist anvendt, mens der nogle tidspunkter har været benyttet pair-programming, er metoden ikke blevet brugt hele tiden. Således er der skrevet mindre dele af kode isoleret, som så efterfølgende er gennemgået og samlet sammen.

Agile software development ligger også fokus på, at det benyttes på en dynamisk udviklingsproces over planlagt udvikling. Hermed menes der naturligvis at planlægning benyttes, men at hvis der opstår problemer eller andre uventede dilemmaer undervejs i udviklingsprocessen, så

FiXme Note! håndteres dette med det samme.¹

9.2 Test driven development

TDD baserer sig på test til udvikling af kode. TDD anvender en simpel iterativ arbejdsproces for udvikling af kode (Insert image?)². Processen baserer sig på 5 skridt, add a test, run all tests and see if the new one fails, write some code, run tests, refactor code og så gentages processen³. Denne udviklingsproces er blevet delvist benyttet i denne løsningsudvikling.⁴ For at tests ikke

FiXme Note!

FiXme Note!

FiXme Note!

¹FiXme Note: Marc, find din kilde

²FiXme Note: Insert image?

³FiXme Note: Var der ikke nogle vise ord fra Jacob omkring denne sætning?

⁴FiXme Note: Husk uddybning i testafsnit

bliver uoverskueligt store, er der foruden et andet princip, som anvendes i løsnings arbejdet, “keep the unit small”. For at reducere tid benyttet på debugging, samt at øge læsbarheden af tests, programmeres der i mange små methods og classer, snarere end enkelte store.⁵

FiXme Note!

⁵FiXme Note: Diskuter/konkluder på det

10.1 Grafisk brugergrænseflade

I forbindelsen med udviklingen af systemet, skal man sørge for at designe det, så brugerne kan finde rundt i de forskellige funktioner. Da sejlklubben Sundets medlemmer kan bestå af en meget mangfoldig gruppe af mennesker, er det svært at indskrænke brugerne i en enkelt gruppe. Brugerne af systemet konkluderes derfor at kunne have meget forskellige evner inden for brugen af IT.

10.1.1 Designet

Programmets brugergrænseflade er designet ud fra principperne ifølge. [?]

- Minimalisme
- Konsistente

Det betyder at der skal være så få forstyrrelser på skærbilledet som muligt. Det skal være enkelt og simpelt at navigere rundt og finde de funktioner i programmet, man skal bruge. Der skal ikke være mange forskellige farver, og generelt skal programmet følge samme struktur til at navigere rundt igennem hele programmet. Herudover skal der ikke være en dyb menustruktur; helst ikke et dybere hierarki end 3.¹

FiXme Note!

Disse forskellige principper eller retningslinjer, er forsøgt implementeret i systemet.

10.1.2 Implementation

Menustrukturen består af tabs, som er store og lette at se. De forskellige funktioner i programmet er delt op i deres tilhørende tabs, og man kan altid gå ind i en ny tab uanset hvor, man befinder sig i programmet. Dette betyder, at strukturen ikke har et hierarki, da uanset hvor man befinder sig, kan der navigeres ind i alle programmets funktioner. Der er valgt blå nuancer, da det er en sejlklub, og dermed virkede som en logisk beslutning. Grunden til der ikke er flere farver rundt omkring i programmet, skyldes den minimalistiske, samt den konsistente, tankegang.

FiXme Note!

¹FiXme Note: Er dette nu også korrekt?

Når brugergrænsefladen er lavet, skal den dermed også testes. Der vil blive forklaret hvordan dette vil blive gjort i et senere kapitel.³

FiXme Note!

10.2 Windows Presentation Foundation

Windows Presentation Foundation (WPF) er en grafisk brugergrænseflade på Windowsbaserede applikationer. WPF gør brug af bl.a. vektorbaseret rendering af grafik, databinding, til nem redigering af data igennem den grafiske brugergrænseflade, og har også inkluderet Extensible Application Markup Language (XAML), hvilket er en nem måde at skabe WPF-grafik på.[?]

I dette projekt bruges WPF til at skabe den grafiske brugergrænseflade. Valget stod mellem WPF og Windows Forms (WinForms), hvilket er forgængeren til WPF til generering af grafik til Windows. Valget faldt på WPF, da XAML, som WinForms ikke har, gør det let at hurtigt lave en grafisk brugergrænseflade. Microsoft har også informeret om, at der ikke længere tilføjes nye funktioner til WinForms, men kun udelukkende rettelser af fejl.[?]

En helt anden mulighed, som blev diskuteret, var at bruge en hjemmeside, som ville have den fordel, at den kan køre på stort set alle enheder, som har adgang til internettet via en browser. En hjemmeside, som gør brug af C#, vil kunne laves lidt på samme måde som ved WPF; med HTML (HyperText Markup Language) og CSS (Cascading Style Sheets) som frontend og C#-klasser som backend. En hjemmeside blev fravalgt, da det blev vurderet at WPF var nemmere at oprette end en hjemmeside, og at en grafisk brugergrænseflade ikke er en essentiel del af opgaven.

10.2.1 Ofte anvendte WPF-controls

Det er ofte de samme WPF-controls, som anvendes til opbygningen af funktionaliteterne i den grafiske brugergrænseflade. Disse vil blive beskrevet her. På figur 10.2.1 vises de beskrevne elementer (Note: De 3 sidste elementer er henholdsvis DataGrid, ComboBox og ListBox).

ComboBox

I WPF er en ComboBox det element, som andre steder omtales som en dropdown menu. Dens indhold kan indstilles enten i XAML-koden eller i Code-Behind koden (her ment C#-koden). Hvis dette indhold skal være dynamisk, vil Code-Behind ofte være anvendt, eksempelvis hvis man kun vil have de medlemmer af en liste, som opfylder et givent prædikat.

Button

En Button er en knap, som kalder en metode i Code-Behind koden, når den trykkes. Den metode udfører det kode, som er angivet i den.

²FiXme Note: Her skal tilføjes nogle screenshots, så man kan se hvordan det helt konkret ser ud. Dette synes jeg dog skal vente til programmet når et mere færdigt stadie.

³FiXme Note: Tilføj en kilde til kapitlet omhandlende test af programmets GUI

DataGrid

Et DataGrid er et grafisk element, som kan vise data dynamisk. Ofte vil det vise en liste af objekter. Dets layout er opdelt i rækker og koller, hvor hver kollone indeholder en bestemt type data, og kan sorteres ved at klikke på labellet. Det er også muligt at konstruere en søgefunktion, altså kan et DataGrid bliver filteret.

CheckBox

En CheckBox er en kvadratisk boks, som enten er "Checked" eller ikke "Checked". Dens stadie er binært, og (som standard) uafhængig af andre elementer.⁴

FiXme Note!

RadioButton

En RadioButton er en cirkelformet CheckBox. Dog vil RadioButtons ofte optræde i serie, altså 2 eller flere sammen. Et oplagt brug af dem er til ja, nej (og måske) situationer, hvor kun er en af dem, man skal vælges.

StackPanel

Et Stackpanel er ikke synligt for slutbrugeren, men bruges til at organisere de andre grafiske elementer. Princippet er, at alle elementer i et StackPanel bliver skubbet sammen i mod en given retning. Dette gør det nemmere at få de grafiske elementer til at flugte. Derudover sikrer det en konsistent bredde af elementer deri, med mindre andet er angivet.

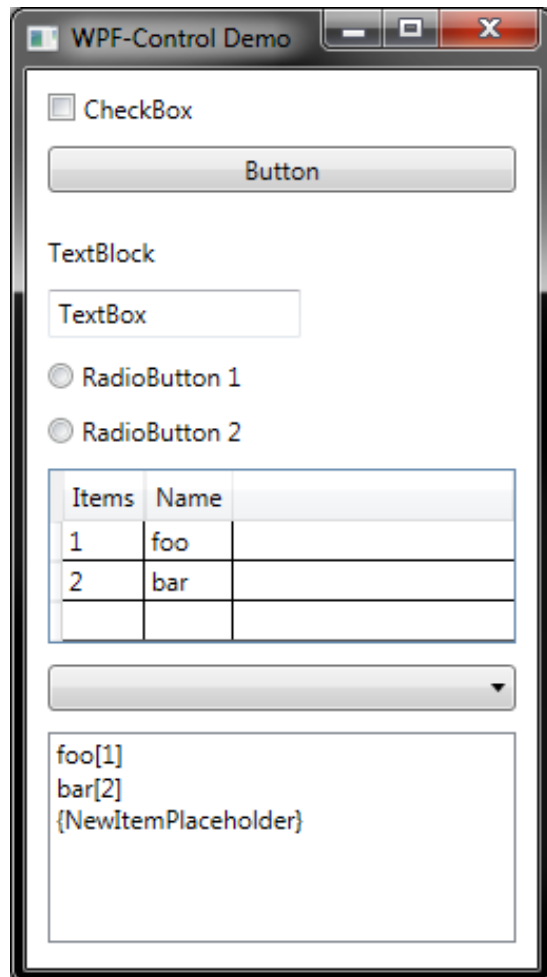
TextBlock

En TextBlock bruges til afbildning af tekst, som ikke kan redigeres. Det vil typisk være en forklarende tekst op ad et andet grafisk element. Det er også muligt at ændre en TextBlock via Code-Behind. Hvis man vil føre en dialog med brugeren, eksempelvis til fejlbeskeder.

TextBox

En TextBox er et tekstfelt, hvori brugeren kan skrive infomation. Dette kan dermed ses som et inputfelt. Det er også muligt at sætte sådan

⁴FiXme Note: Måske give et eksempel på dette



Figur 10.1. Demonstration af WPFs Controls

et felt til at være skrivebeskyttet, samt ændre dets indhold.

ListBox

En ListBox er et tabel, med en kolonne og flere rækker, hvori en bruger kan vælge en eller flere elementer. ListBoxen er i stand til at indeholde samlinger af data på enhver form, oftest vil det være en streng, men et billede er også en mulighed. Den vil i nogen tilfælde have samme brugsscenario som en ComboBox eller et DataGridView. Forskellen fra en ComboBox er, at der kan være flere synlige elementer i en ListBox på samme tid, samt der ikke er nogen dropdown menu. Et DataGridView kan indeholde flere informationer på kolonner, mens en ListBox kun kan indeholde en information.

UserControls

Det er mulig at konstruere sine egne grænsefladeelementer fra et eller flere af WPFs indbyggede eller 3. parts Controls. Dette kaldes en UserControl, som indeholder både en grafisk del og en Code-Behind del. Det bruger-skabte element kan derefter genbruges flere steder i programmet. Dette er et eksempel på genanvendelse, hvilket kan bidrage til højere programkvalitet og større konsistens gennem programmet.

I dette kapitel vil programmets opbygning blive beskrevet ud fra UML-diagrammet, som kan findes i appendiks F på side 91. Dataene indeholdt i klasser kan ses ud fra UML-diagrammet. De fleste af dataene indeholdt i klasserne giver sig selv, men nogle vil blive uddybet. Sidst i kapitlet vil der blive argumenteret for dataene der skal være på medlemmerne.

11.1 Klasser

For at holde styr på medlemmer i sejlklubben, er der lavet en overordnet person klasse. Klassens “PersonId” er et nummer som kan bruges til at skelne de enkelte personer fra hinanden. “BoatDriver” er en boolsk værdi som afgør om personen har et førerbevis eller ej. “SailClubMember” klassen arver fra person klassen og har et “MemberId”, et “Username” og “PasswordHash”. De to sidstnævnte skal bruges til logge på i programmet. Klassen har også et “Position” medlem. Denne afgør hvilke slags medlem man er. Medlemstyper er som følger:

- SupportMember
- Student
- Member
- Teacher
- Admin

Programmet indeholder også en båd klasse. Dataene indeholdt i denne klasse giver sig selv ud fra UML-diagrammet. Der er også en overordnet klasse for sejlture, som er abstrakt. Denne arver klasserne skoletur og almindelig tur fra. Der skal for hver sejltur, registreres hvilken båd der bliver brugt samt tidspunkt for afgang og ankomst. Vejrforholdene skal registres og der kan tilføjes de kommentarer der muligvis kan være. Når der er tale om en almindelig tur, registreres der hvem kaptajnen er og de øvrige besætningsmedlemmer bliver tilføjet til en liste. Der skal desuden registreres, hvornår man regner med at være tilbage fra turen, hvad formålet med turen er og hvor man sejler henne. Dette skal gøres, så det er lettere at finde båden og besætningen, f.eks. i tilfælde af at en ulykke forekommer. Ved en skoletur er der registret et skolehold, som er defineret i en klasse. Skoleholds klassen indeholder lærer og elever. Man kan tilføje yderligere elever til en skoletur, som normalt ikke er med, samt gæster som ikke er elever. Der registreres også hvem der deltager på turen, da der kan være elever fra det normale hold, som ikke møder op. Til sejlturene er der tilknyttet en skadesrapport, som skal, som navnet angiver, bruges til at registrere hvis der kommer skader på båden under sejlladsen.

11.2 Medlemsdata

“PersonId” og “MemberId” bliver, som nævnt, brugt til at skelne de forskellige personer og medlemmer fra hinanden, ligesom “FirstName” og “LastName” bliver brugt til identificering. Adresse dataene er almindelige for en klub at have på sine medlemmer, men bliver ikke brugt til noget i programmet. Ligeledes bliver “DateOfBirth”, “Gender”, “PhoneNumber” og “Email” ikke brugt. “BoatDriver” fortæller, som sagt, hvorvidt medlemmet har førerbevis, og bestemmer i programmet om medlemmet har lov til at være “Captain” på en sejltur. “Position” afgør hvad medlemmet har adgang til i programmet, f.eks. har et medlem der er “Student” adgang til undervisningsfunktionerne i programmet, hvorimod et almindeligt “Member” ikke har. “Username” og “PasswordHash” bruges, som sagt, til at logge på.¹

FiXme Fatal!

¹FiXme Fatal: Ændre tekst til generel programopbygning

Klasser 12

I dette kapitel vil funktioner i programmet blive forklaret.

Navn: Person

Felter: I personklassen findes felter, som indeholder informationer vedrørende personen såsom navn, alder, adresse, postnummer og kontaktinformation. Desuden indeholder klassen også et felt, **Boat-Driver**, som angiver om personen har lov til at føre en båd.¹

FiXme Note!

Metoder: Personklassen indeholder en metode, **ToString()**, som overrider den eksisterende ToString metode og returnerer personens fulde navn. Denne metode er oprettet for at gøre det lettere andre steder i programmet at hente personers fulde navne.

Anvendelse: Personklassen fungerer som baseklasse for SailClubMember klassen. Grunden til at personklassen og SailClubMember klassen ikke er slået sammen til en klasse er, at personklassen også er grundlag for en gæst i systemet. Ved en booking af en sejltur angiver man hvilket hold man skal have med, og i tilfælde af at man skal have en gæst med, som ikke er medlem af klubben, oprettes en instans af personklassen, med de informationer som man angiver i bookingen.

Person
+PersonId: Int
+FirstName: String
+LastName: String
+Address: String
+PostCode: String
+CityName: String
+DateOfBirth: String
+BoatDriver: Bool
+Gender: Gender
+PhoneNumber: String
+Email: String
+FullName: String

Figur 12.1. Person

Navn: SailClubMember

Felter:

Basis medlemsklassen for systemet er SailClubMember. Klassen er en underklasse til den mere generelle Person klasse. Udbygelsen i denne klasse er, at et medlem har et medlemsid, en position, et brugernavn og et password.

SailClubMemberId bliver brugt i hele systemet til at identificere hvilken bruger der gør

SailClubMember
+SailClubMemberID: Int
+Position: Positions = Member
+Username: String
+PasswordHash: String
+SetMemberId(memberId:String): Void

Figur 12.2. SailClubMember

¹FiXme Note: Har vi overhovedet anvendt det felt til noget?

hvad.

Positions feltet angiver at personen er et medlem i klubben. Positions feltet kan også sættes til at være administrator, i tilfælde af at personen er administrator. Positions feltet angiver desuden hvilke rettigheder de forskellige brugere har, f.eks. har en administrator lov til at tilgå alle funktioner i programmet, mens et medlem ikke har lov til at gå ind i undervisnings delen af programmet.²

FiXme Note!

Username og **PasswordHash** felterne anvendes i loginsystemet, som sikrer at hver bruger kan logge ind på systemet og sikre at de kan tilgå funktionerne.

Anvendelse:

SailClubMember klassen er central i programmet, da alle brugere/medlemmer af sejlklubben skal oprettes som et SailClubMember, og deri skal deres position i klubben angives, så de kan tilgå de funktioner i programmet, som de kunne have gavn af. Derudover findes der i RegularSailTrip-klassen og Logbook-klassen referencer til SailClubMember. Dette gøres således, at information om hvem der har foretaget en booking af en båd, og hvem der har skrevet logbogen til den sejltur, der blev foretaget. Disse referencer er også vist på det fulde UML-diagram, som findes i appendiks F på side 91

Navn: StudentMember

Felter:

StudentMemberklassen arver fra SailClubMemberklassen og har derfor alle de felter SailClubMemberklassen også har, men StudentMemberklassens **Position** bliver altid sat til at være 'Student'. **AssociatedTeam** henviser til det skolehold, den pågældende elev hører til. De seks bool-værdier: **RopeWorks**, **Navigation**, **Motor**, **Drabant**, **Gaffelrigger** og **Night** repræsenterer hver et læringsområde, og de bliver sat til 'true', når området er lært.

Anvendelse:

Denne klasse fungerer som en elev på et skolehold og bruges til at holde styr på ens frem-

StudentMember
+Position: Position = Student +AssociatedTeam: Team +RopeWorks: Bool +Navigation: Bool +Motor: Bool +Drabant: Bool +Gaffelrigger: Bool +Night: Bool

Figur 12.3. StudentMember

²FiXme Note: evt. ref til tabel over rettigheder

skridt mod at blive bådfører. I løbet af en to-årig periode bør man have lært alle områderne, og herefter bliver man “forfremmet” til SailClubMember og får et bådførerbevis.

Navn: Team

Felter:

TeamId bruges til at identificere de enkelte hold fra hinanden. **Teacher** er den lærer der er tilknyttet holdet, og **Name** er navnet på holdet. **Level** fortæller om holdet er et første eller andet års hold. Klassen har en “collection” (**TeamMembers**), som der bliver sat *StudentMembers* ind på, hvilket repræsenterer de elever der er på skoleholdet. “Collectionen” **Lectures** indeholder de lektioner,

FiXme Note! som klassen har haft og skal have.³

Anvendelse:

Teamklassen fungerer som et skolehold. Den bliver brugt til at holde styr på eleverne og bruges som besætningen på en båd, når

FiXme Note! denne reserveres.

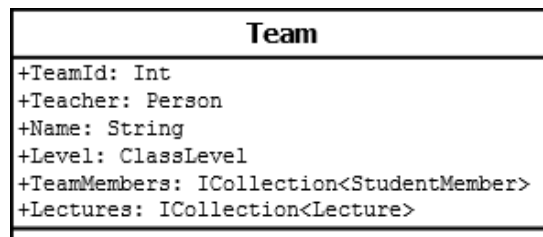
4

Navn: SailTrip og RegularSailTrip

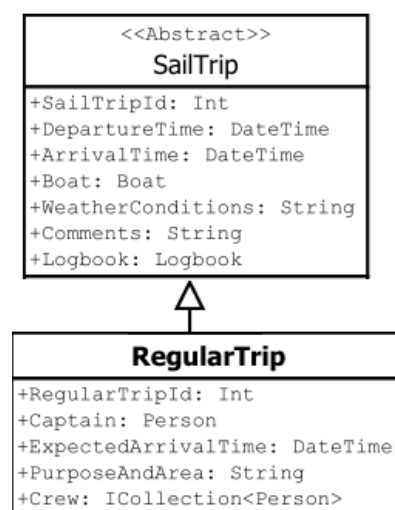
I en tidlig designstruktur var det tænkt at programmet skulle have indeholdt flere forskellige typer af sejlture, og derfor blev SailTrip-klassen skabt som værende baseklasse for disse andre klasser. Senere blev ideen om at have flere typer sejlture ændret, og i stedet blev der valgt, at der kun skulle være én sejlturstype. SailTrip-klassen er dog bibeholdt i tilfælde af videre udbygning af programmet. Dog findes der enkelte felter i klasserne, som overlapper hinanden lidt, men felter er også bibeholdt i tilfælde af udbygning til flere typer sejlture.

Felter:

SailTrip og RegularSailTrips felter udfyldes med informationer vedrørende en sejltur. Felterne **Departure-**



Figur 12.4. Team



Figur 12.5. SailTrip

³FiXme Note: Er ikke sikker på dette, det skal omskrives ud fra hvad der bliver skrevet om 'Lecture'-klassen

⁴FiXme Note: Er ikke sikker på dette, det skal omskrives ud fra hvad der bliver skrevet om 'Lecture'-klassen

Time og **ArrivalTime** indeholder informationer om, hvornår sejlturen starter og slutter.

Der findes en reference til et object af typen **Boat** i **SailTrip**. Denne reference kobler en sejltur sammen med en bestemt båd således, at der holdes styr på, hvilke både der er optaget hvilke tidspunkter.

Desuden findes der i begge klaser et **Id**, **SailTripId/RegularTripId**, som bruges til at adskille de forskellige sejlture fra hinanden.

I **RegularSailTrip** findes der desuden en samling af personer i feltet **Crew**. Her er det værd at bemærke, at besætningen er en samling af typen **Person**, hvilket gør det muligt at vælge besætningsmedlemmer, som ikke er medlemmer af klubben, da de kan oprettes som gæster og derved tilføjes til besætningen.

I **SailTrip** er der også en reference til en **Logbook**. Når en sejltur oprettes, i form af en booking, bliver logbogen sat til at være tom. Først når turen er omme og vedkommende, som foretog bookingen, logger ind i programmet og udfylder logbogen for turen, vil den indeholde data.

Anvendelse:

SailTrip-klassen anvendes kun som baseklasse for **RegularSailTrip**-klassen. **RegularSailTrip** er derimod en central del i programmet, da det ene hovedformål for programmet er at gøre det muligt at foretage bookings gennem et program. Det andet hovedformål, undervisningsdelen, gør også brug af sejlturene.⁵

FiXme Note!

Navn: Event

Felter: Klassen består af 7 felter: **EventDate**, **EventTitle**, **SignUpReq**, **Description**, **SignUpMsg**, **Created** og så en liste af klassen **Person**, som hedder **Participants**.

Description er selve begivenhedsbeskrivelsen.

SignUpReq bruges til registrering af, om en begivenhed kræver tilmelding og **SignUpMsg** bruges til at vise beskeden "Tilmelding krævet!", så man kan se, at der kræves tilmelding til begivenheden.

⁵FiXme Note: nogen der har styr på undervisnings delen må gerne lige skrive ind hvordan sejlturene bliver anvendt der

Feltet **Created** bruges internt i programmet, til at tjekke om en begivenhed oprettes korrekt.

Den eneste metode Eventklassen indeholder en override af ToString, som returnerer

Description.

Anvendelse:

Eventklassen er den klasse om bruges til oprettelse af begivenheder, nødvendigvis ikke sejlbegivenheder, men generelle begivenheder. Et eksempel kunne være en grillaften eller natsejlsads.

Participantslisten bruges når folk skal tilmeldes til begivenhederne; hver begivenhed har sig egen liste med tilmeldte.

Navn: Logbook

Felter:

Logbook klassen indeholder et **LogbookId**, som bruges til at adskille de forskellige logbøger, der bliver oprettet efter sejlturene.

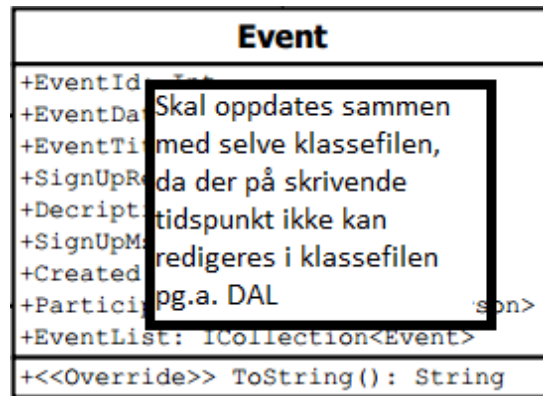
Derudover findes der desuden to tids felter, **ActualDepartureTime** og **ActualArrivalTime**, som bruges til at gemme det faktiske afgang- og ankomsttidspunkt. På samme måde findes der den reelle besætning, **ActualCrew**, hvor den besætning som var på turen indsættes.⁶ Disse 3 felter er oprettet, da der imellem bookingen og sejlturen kunne være sket ændringer i besætningen og, hvornår sejlturen faktisk blev påbegyndt og afsluttet.

Feltet **DamageInflicted** bruges til at angive, om båden tog skade under sejlturen. I tilfælde af at båden faktisk har taget skade, bliver det muligt at beskrive skaden, hvilket bliver gemt i feltet **DamageDescription**. Envidere kan en administrator på systemet svare på skaden i feltet **AnswerFromBoatChief**.

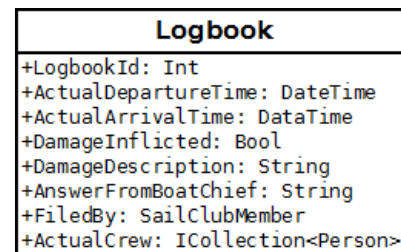
Feltet **FiledBy** bruges til at registrere hvilket medlem af klubben, som har udfyldt den pågældende logbog.

Anvendelse:

I programmet har oprettede sejlture(bookinger) en reference til den logbog, der hører til turen. Logbogen selv har



Figur 12.6. Eventklasse



Figur 12.7. Logbook

FiXme Note!

⁶FiXme Note: Lidt sjov sætning, ... der den... kan ikke helt tyde meningen

referencer til de personer, som besætningen bestod af, og et bestemt medlem af klubben som har udfyldt logbogen.

Lecture-klassen

FiXme Note!

⁷ **Navn: Lecture** 'Lecture' klassen benyttes som en property i 'Team' klassen og indeholder en 'DateTime' for lektionens start, seks booleans som benyttes for en indikator af hvad der er blevet undervist i, samt en ICollection af 'StudentMember' klassen hvori hver fremmødte elev til lektionen er indeholdt. Klassens funktionalitet i programmet er at holde styr på et holds samlede lektioner, hvad der er undervist i samt at holde styr på hver enkelt elevs fremskridt i undervisningsområderne.

FiXme Note!

8

⁷FiXme Note: indsæt UML diagram

⁸FiXme Note: Formatet er stadig ikke 100 % konsistent: Eventklassen vs event-klassen og andre fremmedeord om der skal være bindestreg eller bare sammensat og Lecture klassens generelt

Database 13

I forbindelse med programmets data var det nødvendigt at overveje, om programmet skulle have foruddefineret data, som ville blive nulstillet ved hvert programopstart, eller om dataene skulle gemmes mellem kørsler.

Efter nogen overvejelse blev det bestemt, at der skulle benyttes et såkaldt persistenslag. Efter at have overvejet SQLite og Entity Framework (EF) faldt valget på EF, der er populært blandt C#-udviklere, da det tillader en hurtig start på kodeprocessen.

EF er et såkaldt Object Relational Mapping (ORM) framework, der sammenkæder tabeller i en database med objekter i et program. Det kan benyttes på flere måder, der kort fortalt afhænger af, om man starter med en defineret database, eller en samling af klasser. Sidstnævnte mulighed kaldes for “Code First” og blev den valgte metode, da det tillod en – for gruppen – logisk arbejdsproces, hvori klassestrukturen blev opbygget, og databasen blev automatisk tilpasset denne.

Der blev truffet afgørelse i gruppen om at benytte et såkaldt Data Abstraction Layer (DAL) til at give programmet yderligere robusthed. Et DAL er en samling af interfaces, samt implementationer af disse, som lægger sig mellem programmet og det valgte persistenslag. Hermed opnås mulighed for at udskifte persistenslag, eller sågar benytte flere forskellige i det samme program. Ligeledes bliver det muligt at have et lag specifikt til at teste med. I yderste konsekvens, hvis det skulle vise sig, **af EF fejlede**, og der ikke kunne rettes op på det i tide, så ville det være muligt at skrive en såkaldt “mock implementation”, hvilket ville give den først overvejede mulighed for at have data, som ikke gemmes i noget lager.

Det er også værd at bemærke, at mange udviklere vælger at benytte EF Code First under udviklingen, med en passende DAL-implementation, for derefter, når programmet skal distribueres, at kode en implementation der benytter en database eller lignende, som de har fuld kontrol over. EF har et højt abstraktionsniveau, hvilket simplificerer arbejdsprocessen, men fratager udvikleren en del kontrol.

13.1 Det “gamle” — Bare i tilfælde af, at noget af det skal bruges.

For at kunne holde på data, imellem kørsler af programmet, skal der været en form for persistens. Dette opnåes ved at lagere data.

Til lagring af data findes der flere måder, den primære overvejelse er mellem en database og

tekstfiler. Flade filer, eller såkaldte “flat files” på engelsk, er tekstfiler, som bruges til datalagring. Flade filer er en meget simpel løsning, som vil være nemt at få i gang, men den skalerer ikke særligt godt. En database er et stykke software som opbevarer data, på en hensigtsmæssig måde. Der findes en lang række af database management systemer (DBMS) hver med fordele og ulemper, mange af dem understøtter Structured Query Language (SQL) standarden.

Med databaser kan der dog være flere muligheder for datahåndtering, og der er også løsninger til C#, som skulle være til at overkomme at implementere.[?] Af den grund blev der valgt at bruge almindelig database fremfor flade filer.

Gruppen kiggede nærmere på SQLite og Entity Framework (Code-First). SQLite er en serverløs, selvstændig, konfigurationsløs database. Dette gør den meget simpel i anvendelse, samt ville resourceforbruget være meget lavt. Entity Framework er en object-relational mapping (ORM) framework til .NET platformen. Altså får hvert objekt, der ønskes lagret, en plads i databasen. Med “Entity Code First” arbejdsmodellen, så laves klasser først, hvorefter Entity Framework håndterer integrationen med dens database.

Gruppen adspurte vores underviser i Objek Orienteret Programmering, og han anbefalede Entity Framework (Code-First), til vores problemstilling. Valget faldte derfor på Entity Framework, med Code-First mønsteret.

Programmets brugergrænseflade

14

I dette kapitel vil programmet grafiske brugergrænseflade blive beskrevet.

14.1 Primære brugergrænseflade

Det primære vindue tilgås via loginvinduet, som er det der starter ved programstart eller ved at trykke på logudknappen inde fra selve programmet. Der findes 4 vinduer, forskellen mellem dem er, hvilke tabs der er aktive. I hver tab findes der en funktionalitet, samlet set findes følgende tabs:

- Forside
- Undervisning
- Begivenheder
- Medlemmer
- Både

Via hver af disse tabs, vil der være adgang til programmets forskellige funktionaliteter. Der er også en logud knap, som bruges til at vende tilbage til loginvinduet, således en anden bruger

	Gæst	Støttemedlem	Medlem	Elev	Underviser	Administrator
Personlig forside			✓	✓	✓	✓
Se begivenheder	✓	✓	✓	✓	✓	✓
Tilmeld begivenheder			✓	✓	✓	✓
Opret begivenheder					✓	✓
Se sejlture	✓	✓	✓	✓	✓	✓
Opret sejltur			✓	✓	✓	✓
Se logbøger	✓	✓	✓	✓	✓	✓
Opret logbog			✓	✓	✓	✓
Svar på logbog						✓
Se undervisningstimer				✓	✓	✓
Opret undervisningstimer					✓	✓

Tabel 14.1. Tabel over alle brugerniveauer og deres tilladte funktioner.

¹**FiXme Fatal:** Måske skal denne tabel lige laves lidt om, enten indholdet, eller hvordan det virker i programmet.

kan anvende systemet. Programmet er lavet til at køre i opløsningen 1024x720 pixels. Denne opløsning er valgt for at understøtte alt fra bærbare, med opløsninger som 1366x768 pixels i et vindue, og op til FullHD (1920x1080 pixels) op og efter. Stort set alle computerskærme har en opløsning større end 1024x720 pixels[?]. Programmet har et lyst farveskema; med farverne hvid og en lys blå som hovedfarver (#87D4EE).

14.2 UserControls

Der anvendes Usercontrols til at kode både brugergrænsefladen og den tilhørende code behind.

14.2.1 DateTimePicker

Formål: Denne UserControl er lille i forhold til de andre, der findes i programmet. Den bruges, når der skal vælges et tidspunkt, som både indeholder en dato og et tidspunkt. Der



Figur 14.1. DateTimePicker

findes en DateTimePicker i extended WPF toolkit, men da der blev opdaget en underlig fejl ved denne², blev det besluttet at en usercontrol, som gruppen selv kunne programmere, ville være bedre.

FiXme Note!

BrugerGrænseflade: Den består af en DatePicker, og en TimePicker som findes i extended WPF toolkit.

Code-Behind: De to værdier man skriver i henholdsvis DatePickeren og TimePickeren skal samles i en enkelt værdi, som kan tilgås vha. usercontrollen. Dette er gjort, ved at lave et property der kaldes for Value. Det kan ses på listing 14.1, hvordan dette er blevet implementeret. Man kan ved at assigne value igennem en instans af usercontrollen, sætte både usercontrollens TimePicker og DatePicker, og dermed give de to GUI elementer en værdi som brugerne ser. Herudover kan man kalde dens get, hvilket resulterer i at begge værdier bliver sat sammen, hvor TimePickeren bruger TimeOfDay, for at få et TimeSpan, som kan tilføjes direkte på den DateTime, der laves fra en DatePicker, vha. + operatoren.

```

1 public DateTime Value
2 {
3     get
4     {
5         return (DatePicker.SelectedDate.GetValueOrDefault() +
6             TimePicker.Value.GetValueOrDefault().TimeOfDay);
7     }
8     set
9     {
10        DatePicker.SelectedDate = value;
11        TimePicker.Value = value;
12    }
13 }
```

Listing 14.1. DateTimePicker Value

²FiXme Note: evt. uddybe?

14.2.2 Forside

Formål: Formålet med forsiden er at vise aktuel infomation på en overskuelig måde for brugeren. Det er den første side som medlemmer, og op, ser³. Fra forsiden kan man ændre og slette sine bookings, samt starte oprettelsen af en logbog.

BrugerGrænseflade: Brugergrænsefladen på forsiden består primært af to datagrids: Det venstre viser de kommende sejlture, og det højre viser de sejlture personen mangler at udføre logbøger for. Under dem er der knapper, som bliver aktive efter en markering er udført ved, at trykke på en af rækkerne i det tilhørende datagrid.

Code-Behind: For at kun vise de sejlture hvor personen, som er logget, ind deltager i, anvendes der standard query operatorer. I listing 14.2 er der et udsnit af koden, nærmere bestemt den del som vælger de rigtige sejlture. Det første udtryk finder de ture, som er i fremtiden, hvor personen indgår i besætningslisten.

Det andet udtryk finder de ture, hvor den nuværende person er kaptajn, som er forventet at være returneret til havn og ikke har udfyldt logbog.

Begge udtryk returnerer en IEnumerable, som derefter assignes som DataGridenes ItemSource.

```

1 UpcommingTripsDataGrid.ItemsSource =
2   sailTripList.Where(t => t.Crew.Select(p =>
3     p.PersonId).Contains(usrId))
4     .Where(t => t.DepartureTime > DateTime.Now);
5 LogbookDataGrid.ItemsSource =
6   sailTripList.Where(t => t.Captain.PersonId == usrId &&
7     t.ArrivalTime < DateTime.Now && t.Logbook == null);

```

Listing 14.2. Forsidens Code-Behind

14.2.3 Boat user control

³FiXme Note: Medlemmer og hvad ser? (btw. OP is a faggot)

Velkommen Magic User!

Herunder er dine kommende sejlture og til højre er dem, du mangler at udfylde logbog for.

Dine kommende sejlture:

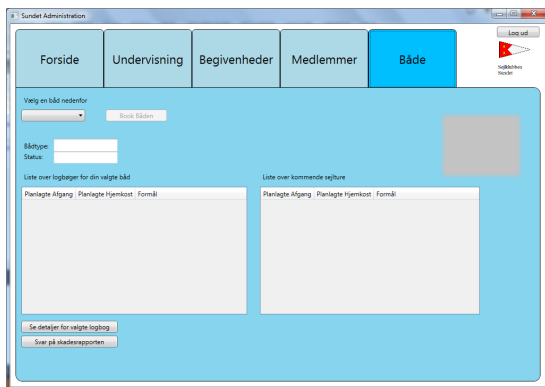
Båd	Kaptajn	Planlagte Afgang	Formål
Anna	John Filbert	Fri 09/05/2014 18:25	Terrorisme

Du mangler at udfylde logbøger for:

Båd	Kaptajn	Planlagte Afgang	Formål
Anna	Magic User	Sun 04/05/2014 18:25	Ass 'n' Titties 2

Ændre booking Slet booking Opret logbog

Figur 14.2. FrontPage-Usercontrol



Figur 14.3. Boat screenshot

Formål:

Under Boat kan man få et overblik over hvilke både som, der er til rådighed i sejlklubben inklusiv bådtype og status på båden (om den f.eks. er operationel). Man kan også booke en båd, se liste over logbøger for en valgt båd og ligeledes se en liste over kommende sejlture (bookings).

Brugergrænseflade:

På siden findes der flere forskellige controls.

Man starter først med en dropdown menu,

hvor man kan vælge, hvilken båd man vil se informationer om. Til højre for den er der en knap "Book Båden", som man kan trykke på, hvis man vil booke båden. Når den trykkes på, så åbner der en ny fane, hvor man kan angive bookingsinformationerne. Helt ude til højre er der et billede af hver enkelt båd. Under dropdown menuen bliver der vist i en textbox bådtypen og bådens status. Til visning af logbøger og kommende sejlture er der to listbokse ved siden af hinanden, hvor hhv. logbøger og kommende sejlture vises efter dato. I selve listboksene kan man se "Planlagte afgang", "Planlagte hjemkomst" og "Formål" ved både logbøger og kommende sejlture. Til sidst er der to knapper: "Se detaljer for valgte logbøger" og, hvis man er logget ind som administrator, "Svar på skadesrapporten". Ved at først vælge en logbog, i listen over logbøger, og derefter klikke på "Se detaljer for valgte logbøger", så kan man se detaljer omkring den valgte logbog. Hvis der dobbeltklikkes på en logbog i listboksen, så opnår man samme effekt. Hvis man er logget ind som administrator, så man kan vælge en logbog og trykke på "Svar på skadesrapporten", hvor der åbnes et nyt vindue, hvor der kan svares på skadesrapporten.

Code-behind:

Metoden `BoatComboBox_OnSelectionChanged` aktiveres ved at vælge en båd i dropdown menuen. Hvis der ikke er valgt en båd, så er `Book Button` ("Booking") ikke aktiv.

Først noget hentning af data fra dal og sortering⁴

FiXme Note!

Efter dataet er hentet fra databasen, så tjekkes der, at hvis brugeren ikke er supportmedlem, så er `BookButton` ("Booking") aktiv.

Derefter laves der et tjek på om `ImagePath` er null, hvis ikke, så er billedestien `ImagePath`, så hver båd har sit eget billede. Hvis `ImagePath` er null, så bliver der vist et gråt billede.

Derefter tjekkes der om båden er operationel dernæst hentes bådtypen og begge vises i hver deres textbox.

Til sidst så bliver dataet, som hentes fra databasen afbildet i hvert sit datagrid.⁵

FiXme Note!

⁴FiXme Note: Skal skrives når database oprettes

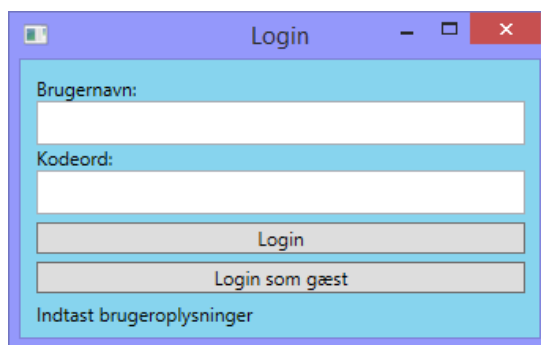
⁵FiXme Note: sæt kode ind

14.3 Vinduer

14.3.1 Login brugergrænseflade

Formål: Login brugergrænsefladen har til formål at verificere en brugers identitet. Det er det første vindue brugeren ser når programmet åbnes, og det er også det vindue, som leder brugeren ind i selve programmet.

BrugerGrænseflade: Brugergrænsefladen til loginvinduet er meget simpel og ligetil. Der er en textbox til brugernavnet og en passwordbox til kodeordet. En passwordbox er en textbox, hvor de indtastede karakterer vises som en sort prik, i stedet for de skrevne tegn, for at beskytte brugeren. Loginknappen trykkes efter brugeren har indtastet sine brugeroplysninger; den åbner hovedvinduet hvis infomationen er korrekt. "Login som gæst"-knappen kræver ingen brugeroplysninger og åbner en let udgave af hovedvinduet, uden særlige tilladelser. Til sidst er en textblock, som fortæller brugeren, hvis der er skrevet forkert.



Figur 14.4. Login interface

Code-Behind: Den centrale del af det Code-Behind som er i forbindelse med loginvinduet, er det der verificerer at en bruger findes og koden er korrekt. I listing 14.3 vises det stykke kode, som tjekker brugeroplysningerne. Først sikres det, at der findes medlemmer i databasen af typen `SailClubMember`. Herefter hentes den anmodede bruger, udtrykket sammenligner det indtastede brugernavn med alle dem i databasen; dette er case insensitive (altså vil "FoOBar" være lig med "foobar" og "FOOBAR"). Hvis brugen findes, så sammenlignes det indtastede kodeord med det i databasen. Her anvendes en hashing algoritme, hvilket sikrer at hvis noget får adgang til databasen, kan de ikke se brugernes kodeord. Er alt infomation gyldigt så kaldes "LoginCompleted"-metoden med argumentet "usr" som er brugeren.

```

1 private void DoLogin(object sender, RoutedEventArgs e)
2 {
3     // If usernamebox or password is empty display an error message.
4     [...]
5
6     if (sailClubMembers != null)
7     {
8         SailClubMember usr =
9             sailClubMembers.FirstOrDefault(
10                x => String.Equals(x.Username, UsernameBox.Text,
11                                   StringComparison.CurrentCultureIgnoreCase));
12
13         // Check if user exists (Case insensitive)
14         if (usr != null &&
15             String.Equals(usr.Username, UsernameBox.Text,
16                           StringComparison.CurrentCultureIgnoreCase))
17         {
18             // Check if the password is correct (Case sensitive)

```

```

17         if (usr.PasswordHash ==
18             EncryptionHelper.Sha256(PasswordBox.Password))
19         {
20             LoginCompleted(usr);
21         }
22         // Code to show errors for wrong infomation
23         [...]
24     }
25 }
26 }

```

Listing 14.3. DoLogin

14.3.2 CreateBoatBookingWindow

Formål: Dette vindue opretter, eller ændrer, en **RegularTrip** (en bådbooking). Dette vindue anvendes fire steder i programmet: Fra **Boats**-usercontrollen, **NewLecture**-vinduet (under Undervisning), og fra **FrontPage**-usercontrollen eller **Boats**-usercontrollen (til at redigere en booking).

BrugerGrænseflade: Vinduet er opbygget af et vertikalt stackpanel. Først vælges en båd, her anvendes en combobox, da der skal vælges en værdi fra en liste. Herefter anvendes **DateTimePicker**-usercontrollen til at vælge et start- og sluttidspunkt. Der vises den nuværende besætning, samt muligheden for at ændre den ved at trykke på en Button med labellet "Ændre Besætning". Når en besætning er valgt, kan der vælges en kaptajn ud fra besætningslisten. Til sidst er der en textbox, hvori brugeren kan angive formålet med turen. Derudover er der to buttons, en til at gemme og en til at annullere.

Code-Behind: Til vinduet er der tre constructors. Dette er således de fire steder, hvorfra vinduet åbnes, hver kan behandle det på sin måde, da to af dem er ens. De tre constructors er i listing 14.4. Den første constructor skaber et nyt vindue, herunder henter den bådede fra databasen og bliver kaldt fra Boats-usercontrollen. Den vælger også den båd, som er angivet i dens indeksparemeter. De to datetimerpickere sættes også til det nuværende tidspunkt, dette gør det nemmere at vælge et passende tidspunkt for den kommende booking, ellers ville der blive valgt defaultværdien, hvilket er 1/1/0001.

Fornavn	Efternavn	Rank	ID
Jonna	Gored	Admin	
Magic	User	Admin	
John	Filbert	Member	
Jens	Hansen		

Figur 14.5. CreateBoatbookingWindow

FiXme Fatal!

De to andre constructors kalder den første, da den er grundlæggende for at kunne bruge vinduet. Constructoren med forskriften “CreateBoatBookingWindow(RegularTrip rt) : this(-1)”, ændrer teksten på gem knappen fra “Gem Booking” til “Ændre Booking”,⁶ samt den ændrer hvilken metode den kalder, således der ikke forsøges at oprette en ny sejltur, men derimod opdaterer en eksisterende. Fælles for de to metoder, som kaldes af knappen, er at de skal verificere gyldigheden af en tur. Dette udføres i “CreateSailTrip()” som returner en gyldig instans af SailTrip-klassen, hvis turen er gyldig eller null og en fejlbesked, hvis den ikke er. Dette null bliver håndteret af kaldermetoden, således der ikke opstår exceptions.

```

1 // Called to Initialize the window, from the other constructors and
  // from the Boats-UserControl
2 public CreateBoatBookingWindow(int index)
3 {
4     InitializeComponent();
5
6     // Initialize ComboBoxes and Databaseforbindelsen
7     [...]
8
9     // Set DateTimePickers to the current time.
10    [...]
11 }
12
13 // Called to edit a trip from the FrontPage
14 public CreateBoatBookingWindow(RegularTrip rt) : this(-1)
15 {
16     // Sets all the value from the input RegularTrip
17     [...]
18
19     // Change the text and behaveour of the buttons
20     [...]
21 }
22
23 // Called from NewLecture
24 public CreateBoatBookingWindow(DateTime departure, DateTime arrival,
  Team currentTeam) : this(-1)
25 {
26     // Sets values to match the parameters given
27     [...]
28
29     // Set the Captain to be the tracher
30     [...]
31
32     // Add a description of the class in the PurposeTextBox.
33     [...]
34
35     // Call the SaveFunction
36     [...]
37 }

```

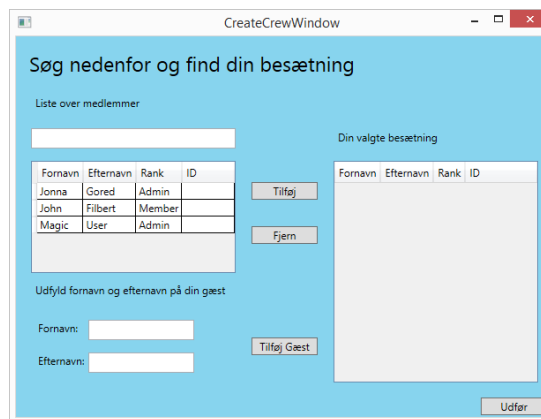
Listing 14.4. De tre constuctoreres forskrifter

⁶FiXme Fatal: Skal der være nutids-r på ændre(r)? Det skal der vel ikke, når det er i bydeform - Thomas

14.3.3 CreateCrewWindow

Formål: Dette vindue åbnes op to steder i programmet: **CreateLogbookWindow** og i **CreateBoatBookingWindow**. Det bruges, når der skal laves en besætning til en RegularSailtrip.

BrugerGrænseflade: Der er to datagrids, som indeholder to lister. Listen til venstre består af SailClubMembers, som bliver hentet ind fra databasen. Listen til højre indeholder det Crew, som man er i gang med at udforme til enten RegularSailTrip eller Logbook. Der er også tilføjet tekstfelter, så man kan skrive navnet på en gæst, man tog med på sejlturen. Der er knapper, som tilføjer de forskellige personer til Crewlisten. Øverst findes også et tekstfelt til at søge listen over medlemmer igennem. Når man har lavet sin liste, kan man trykke udfør, for at komme tilbage til vinduet, der kaldte CreateCrewWindow.



Figur 14.6. CreateCrewWindow

Code-Behind: På listing 14.5 kan man se koden der sker, når man trykker på knappen med teksten Tilføj Gæst. Der er blevet brugt et **regular expression** til at tjekke, om den string brugeren angiver i de to tekstbokse for gæstens navne, er **lovlige**. Det er blevet valgt, at man må bruge hele det danske alfabet samt mellemrum, så navne såsom: Lars Peter Østergaard, er mulige. Hvis begge tekstbokse er lovlige, kommer man ind i det inderste if-statement, hvor der laves en ny person med det pågældende FirstName og LastName. Derefter tilføjes personen til listen, der vises i datagriddet til højre, og til sidst kaldes RefreshDatagrid, som man kan se på listing 14.6 på modstående side.

```

1  {
2      if (Regex.IsMatch(FirstNameBox.Text, "[A-ZÆØÅa-zæøå ]*$") &&
        FirstNameBox.Text.Trim() != String.Empty)
3      {
4          if (Regex.IsMatch(LastNameBox.Text, "[A-ZÆØÅa-zæøå ]*$") &&
            LastNameBox.Text.Trim() != String.Empty)
5          {
6              var p = new Person();
7              p.FirstName = FirstNameBox.Text;
8              p.LastName = LastNameBox.Text;
9              CrewList.Add(p);
10
11              RefreshDatagrid(CurrentCrewDataGrid, CrewList);
12
13              FirstNameBox.Clear();
14              LastNameBox.Clear();
15          }
16      }
17      else
18      {
19          MessageBox.Show("Ugyldigt navn. \nPrøv venligst igen");
20      }
    }

```

21 }

Listing 14.5. Add Guest Button

FiXme Note! Her modtages der et datagrid, som skal have dets Itemssource refreshet⁷, og en ICollection, som er det data, der skal sættes ind i datagridet. Det gøres ved at assigne dets Itemssource til null, og derefter assigne det tilbage til den ICollection, der blev sendt med.

```

1 private void RefreshDatagrid(DataGrid Grid, ICollection<Person> list)
2 {
3     Grid.ItemsSource = null;
4     Grid.ItemsSource = list;
5 }

```

Listing 14.6. Refresh Datagrid

Knappen til at tilføje et medlem kan ses på listing 14.7. Inden det valgte medlem tilføjes til listen tjekkes der, om medlemmet allerede findes i den anden liste. Dette gøres ved at anvende standard query operatører. Først et where med et lambda udtryk, hvor der findes alle SailClubMembers i listen. Herefter castes disse om til SailClubMembers, for derefter at kunne tjekke, om det valgte medlems SailClubMemberId er forskelligt fra de andre i listen. Hvis det hele er true, så bliver det valgte medlem tilføjet til listen, og RefreshDataGrid kaldes for at opdatere datagridet.

```

1 private void AddButton_OnClick(object sender, RoutedEventArgs e)
2 {
3     var currentPerson = (SailClubMember) MemberDataGrid.SelectedItem;
4
5     if (
6         CrewList.Where(x => x is SailClubMember)
7             .Cast<SailClubMember>()
8             .All(x => x.SailClubMemberId !=
9                 currentPerson.SailClubMemberId)
10        )
11        CrewList.Add(currentPerson);
12
13        DataGridCollection.Filter = Filter;
14        RefreshDatagrid(CurrentCrewDataGrid, CrewList);
15    }

```

Listing 14.7. Add Member

14.3.4 CreateLogbookWindow

Formål: Dette vindue bruges når et medlem, som har booket en båd skal udfylde sejlturens logbog. Vinduet tilgås fra usercontrollen FrontPage. Medlemmet skal udfylde de forskellige felter der findes i vinduet og trykke udfør for at gemme logbogen i databasen.

⁷FiXme Note: Er det ikke bedre med "opdateret" i stedet for refreshet?

Figur 14.7. CreateLogBookWindow

BrugerGrænseflade: Der er mange elementer på dette skærmbillede. Øverst til venstre finder man en textbox, som udfyldes automatisk, når man vil udfylde sin logbog. Textboxen indeholder navnet på den båd, logbogen skrives for, og dette loades igennem den RegularSailtrip, som logbogen skrives for. Under den findes en combobox, hvor man vælger kaptajnen eller bådføren for sejlturen. Her kan vælges imellem alle de personer som er sat på besætningslisten. Der bliver altså ikke tjekket, om personerne har duelighedsbevis, da gæsterne netop også kan være kaptajnen eller bådføren. Under dette finder man en textbox, hvor man angiver turens formål. Der er desuden tre radiobuttons, som angiver hvilken tilstand båden er i. Hvis man aktiverer enten knappen med teksten: "Ja men skadet" eller "Nej", så påkræves det, at man udfylder dette felt.

Til højre finder man 4 datetimepicker usercontrols, som beskrevet i afsnit 14.2.1 på side 46. De to øverste er ikke enabled, da de også er loaded fra den RegularSailtrip, man udfylder logbog for. De to andre skal dog udfyldes og ændres fra de standardværdier, som de har, når man åbner vinduet.

Under disse datetimepickers, finder man endnu en textbox, som skal udfyldes med en vejrrapport.

Der findes et lignende vindue som hedder ViewSpecificLogbookWindow. Dette vindue indeholder alle de samme felter, men har også et svar fra BoatChief, som findes i en textbox. Alle elementerne i det vindue er sat til readonly eller IsEnabled="False", da man kun skal kunne se logbogen i det vindue og altså ikke udfylde noget.

Code-Behind: Der er ikke meget kode at se på ved dette vindue, da meget af det er simpelt, men man kan se på den kode, der findes bag knappen med teksten: "Udfør". Koden for dette kan ses på listing 14.8. Først tjekkes der om de forskellige felter i vinduet er udfyldt og, hvis alt er udfyldt, kommer man ind i det sidste else if-statement, der ses i koden. Her afsættes der om båden har taget skade under sejlturen, og derefter gemmes alle felterne i de lokale kopier af både RegularSailTrip og Logbook, hvor der til slut kaldes updatedatabase på de to lokale objekter.

```

1 private void FileLogbookButton_OnClick(object sender, RoutedEventArgs
   e)
2 {
3     if (YesRadioButton.IsChecked == false &&
        NoRadioButton.IsChecked == false)
4     {
5         MessageBox.Show("Udfyld venligst om båden blev skadet
            under sejladsen");
6     }
7     ...
8     else if (YesRadioButton.IsChecked == true ||
        NoRadioButton.IsChecked == true
9             || YesButBrokenRadioButton.IsChecked == true)
10    {
11        if (YesRadioButton.IsChecked == true)

```

```

12         {
13             currentLogbook.DamageInflicted = false;
14         }
15         if (NoRadioButton.IsChecked == true ||
16             YesButBrokenRadioButton.IsChecked == true)
17         {
18             currentLogbook.DamageInflicted = true;
19         }
20         RegularSailTrip.PurposeAndArea = PurposeTextBox.Text;
21         currentLogbook.DamageDescription = DamageTextBox.Text;
22         currentLogbook.ActualCrew = CrewList;
23         currentLogbook.ActualArrivalTime =
24             DateTimePickerActualArrival.Value;
25         currentLogbook.ActualDepartureTime =
26             DateTimePickerActualDeparture.Value;
27         currentLogbook.FiledBy = _currentSailClubMember;
28         RegularSailTrip.WeatherConditions =
29             WeatherConditionTextBox.Text;
30         RegularSailTrip.Crew = CrewList;
31         RegularSailTrip.Logbook = currentLogbook;
32     }

```

Listing 14.8. Gem Logbog

FiXme Note!

8

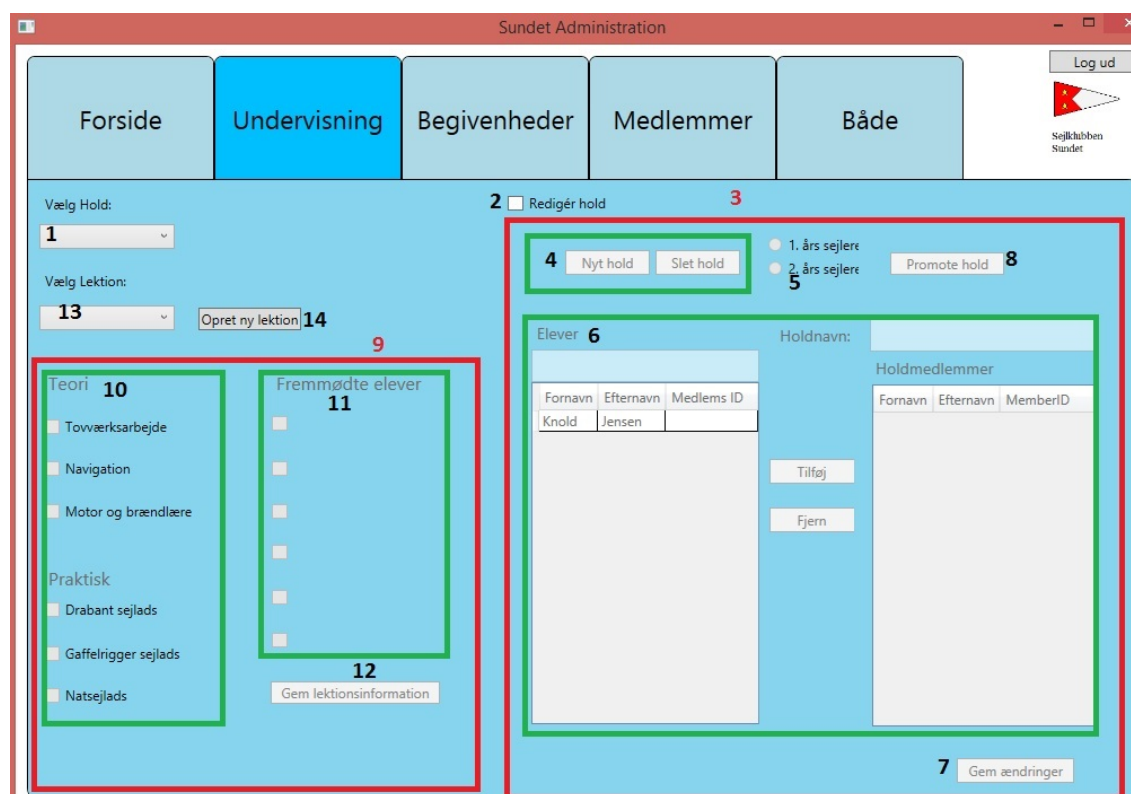
14.3.5 Undervisning

Undervisningsdelen i programmet består af to 'usercontrols' og to 'windows'. Af 'usercontrols' eksistere 'StudyTeacher' som er det et medlem med undervisning eller administratorstatus kan se. Den anden 'usercontrol' er 'StudyStudent', hvilket er den, studenter har adgang til. Hvis man hverken er student, underviser eller administrator, har man således ikke adgang til nogle af disse. 'StudyStudent' er programmeringsmæssigt meget begrænset, idet dens eneste funktion er at repræsentere den enkelte students information, hvilket kun er tilgængelig for student på et read-only niveau. I 'StudyTeacher' delen kan en underviser oprette lektioner og hold, slette hold, ændre på hold samt fuldføre uddannelsesforløbet ved at give studenter deres duelighedsbevis.

14.3.5.1 Brugergænsefladen

StudyTeacher På figur 14.8 på den følgende side ses en 'usercontrol' for 'StudyTeacher'. På de markerede områder ses grupperinger af brugergænsefladens funktioner, som er sammenhængende. **1** referere til en 'combobox', som benyttes til valg af hold. Denne 'combobox' har indflydelse på det meste af undervisningsdelen, da denne information er afhængig af hvilket hold, som er valgt. **2** henviser til en 'checkbox', som har kontrol over **3**, et 'grid' indeholdende funktionalitet til brug af redigering samt kreation af hold. **4** er tilføjelse og sletning af hold, "Slet hold" knappen sletter det hold, som er valgt i **1**, mens "Nyt hold" knappen åbner et 'nested window', 'NewTeam', i programmet, hvor et nyt hold kan blive oprettet. **5** disse 'radio buttons' benyttes for at vælge om holdet er 1. års eller 2. års sejlere.

⁸FiXme Note: Opdater denne listing med Update Database kaldet.

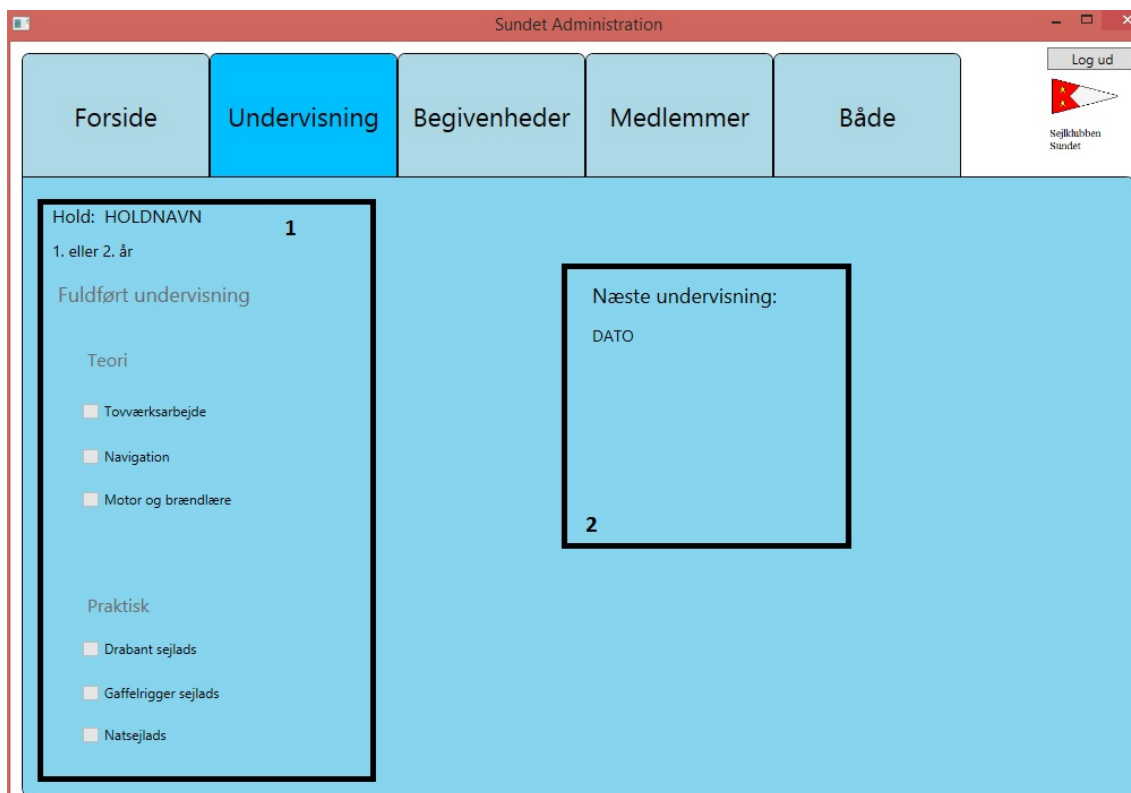


Figur 14.8. UI for undervisning tab som underviser og admin, markeringer bruges til forklaringer nedenfor

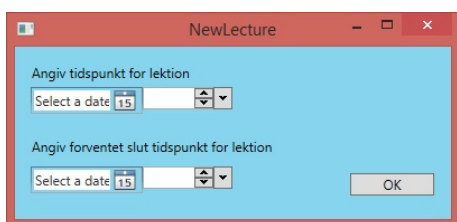
6 dette område benyttes til at tilføje og fjerne medlemmer ved brug af "Tilføj" og "Fjern" knapperne. Det venstre 'datagrid' benyttes til søgning af medlemmer. I dette grid kan alle 'StudentMember' findes, mens i det højre 'datagrid' ses de studerende, som der er på det valgte hold i 1. 7 denne knap gemmer ændringer for holdet. Dette er lavet separat for at undgå kommunikation med database ved hvert klik. 8 denne knap angiver et duelighedsbevis til de medlemmer på det valgte hold i 1, som opfylder alle undervisningskrav. 9 referer til et grid med lektionsinformation. Hvis der ikke er valgt en lektion i 13, er dette 'grid' ikke muligt at benytte. 10 denne række af 'checkbox' bruges til at krydse af hvad der er blevet undervist i på den valgte lektion og 11 benyttes ligeledes til at afkrydse hvilke elever var til stede. 12 fungerer for lektion ligesom 7 for hold og er lavet til samme formål. 13 er en 'combobox' som bruges til valg af lektion. Den sidste funktion 14 åbner et nyt 'window', 'NewLecture', hvor man kan oprette en ny lektion for det givne hold valgt i 1.

StudyStudent På figur 14.9 på næste side ses en 'usercontrol' for 'StudyStudent'. 1 indeholder information omkring personens undervisning, 'checkbox'ene indikerer undervisningsområder for den student som er logget ind, mens teksten ovenfor viser hvilket hold personen er på. 2 viser den næste lektionstidspunkt for det hold, studenten er tilknyttet.

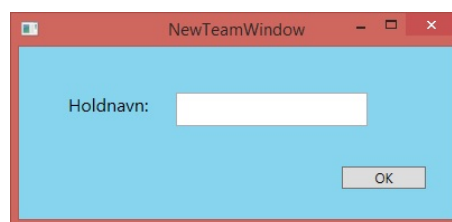
NewLecture og NewTeam På figur 14.10 på modstående side ses et 'window' for 'NewLecture'. I 'NewLecture' vælges der to datoer ud i fremtiden. Disse bruges til at oprette en lektion, hvilket sker når der trykkes 'OK'. På figur 14.11 på næste side ses 'NewTeam'. Her skrives et holdnavn i tekstboksen, og således kan et hold oprettes.



Figur 14.9. UI for undervisning tab som student, markeringer bruges til forklaringer nedenfor



Figur 14.10. UI for NewLecture window



Figur 14.11. UI for NewTeam window

14.3.5.2 Code-Behind

StudyTeacher

StudyStudent

NewLecture 'NewLecture' er for brugeren simpel at benytte, der er dog mere funktionalitet i dette vindue end der bliver vist for brugeren.⁹

FiXme Note!

```
1 private void CompleteLectureCreate_Click(object sender,
    RoutedEventArgs e)
2     {
3         var lecture = new Lecture
4         {
5             DateOfLecture = DateTimePickerPlannedLectureTime.Value
```

⁹FiXme Note: Menes der almindelig bruger set i forhold til administrator?

```
6         };
7         DalLocator.LectureDal.Create(lecture);
8         var Departure = DateTimePickerPlannedLectureTime.Value;
9         var Arrival = DateTimePickerPlannedLectureTimeEnd.Value;
10        var book = new CreateBoatBookingWindow(Departure,
11            Arrival, _currentTeam);
12    }
```

Listing 14.9. Kode for 'OK' knap i 'NewLecture' 'Window'.

I linje 12 ses koden for "OK" knappen i 'NewLecture' vinduet. Datetimepicker bliver aflæst og informationen brugt til at oprette en 'Lecture'. Yderligere bliver informationen også gemt i henholdsvis 'Departure' og 'Arrival'. Disse bliver videresendt i en constructor for booking af både, 'CreateBoatBookingWindow()'. Dette er "OK" knappens indirekte funktionalitet. Funktionaliteten af dette kan ses på linje 27.

```
1 public CreateBoatBookingWindow(DateTime departure, DateTime arrival,
2     Team currentTeam)
3     : this(-1)
4     {
5         List<Boat> boats = new List<Boat>();
6         boats = DalLocator.BoatDal.GetAll().ToList();
7         Boat Anya = new Boat
8         {
9             Type = (currentTeam.Level == Team.ClassLevel.Second)
10                ? BoatType.Gaffelrigger
11                : BoatType.Drabant
12        };
13
14        Boat currentBoat = boats.FirstOrDefault(
15            x => x.Type == Anya.Type);
16
17        BoatComboBox.SelectedIndex = boats.FindIndex(b => b ==
18            currentBoat);
19        CrewList.Add(GlobalInformation.CurrentUser);
20        CaptainComboBox.SelectedIndex = 0;
21        foreach (var member in currentTeam.TeamMembers)
22        {
23            CrewList.Add(member);
24        }
25        DateTimeStart.Value = departure;
26        DateTimeEnd.Value = arrival;
27        PurposeTextBox.Text = "Undervisning af:" +
28            currentTeam.Name;
29        SaveButton_Click(new object(), new RoutedEventArgs());
30    }
```

Listing 14.10. Dette kode bliver indirekte udført når der trykkes på 'OK' knappen, og opretter en bådreservation for lektionen.

Denne constructor nedarver fra 'CreateBoatBookingWindow()' constructeren, som tager en paramter, index. Hertil benyttes ': this(-1)', ses på linje 2, for at sætte combobox'en, der styrer valg af booking, kan ses på <indsæt reference til BoatBooking>¹⁰, til null.¹¹ Alt efter FiXme Note!

FiXme Note! om sejlerholdet er et 1. eller 2. års-hold, skal der bookes henholdsvis en drabant eller en gaffelrigger. Dette håndteres på linje 6 - 15. Der laves en lokal båd, Anya, hvis 'BoatType' bliver sat gennem en 'conditional operator'. Herefter benyttes et 'lambda expression' til at finde den første båd af den korrekte type i databasen og 'boatcombobox', som styrer valg af båd, bliver sat til denne. Efterfølgende bliver den administrator eller underviser der opretter lektionen sat som kaptajn. Det medsendte hold tilføjes 'CrewList', Departure og Arrival sættes også til de medsendte værdier. Formål bliver sat til undervisning og til sidst kaldes det event, som fuldfører bookningen.¹²

¹⁰FiXme Note: look at <>

¹¹FiXme Note: Dårlige sætninger imo

¹²FiXme Note: Enten slettes alle de apostroffer omkring de udvalgte navneord eller tilføjes til alle andre specielle navneord, fra Undervisning og ned passer ikke rigtig sammen med det tidligere skrevet

FiXme Note!

I dette afsnit forklares der om teorien bag programtest samt udførelsen deraf. Foruden ses der nærmere på hvordan resultaterne bruges, hvorfra der kan dannes et overblik over programmet funktionalitet. For at teste programmet anvendes der, foruden unit tests i dette projekt, en række user tests, som tager udgangspunkt i projektets userstories¹. Denne type test ligger fokus på at teste tre primære emner hos brugeren: Funktionalitet, effektivitet og tilfredshed.

For at kunne teste disse tre kriterier udvikles der først en profil af målgruppen. Idet dette program er lavet med hensigt af benyttelse i en sejlkлуб, er denne gruppe rimelig stor. Målgruppen er hele sejlkлубben, hvilket kan være rige som fattige og gamle som unge. Idet det er et computersystem, er det eneste krav til målgruppen, at de har basal kendskab til brugen af en computer.

For at kunne udføre en test, udvikles der således også en række mål for, hvad brugeren ønsker at benytte programmet til. Ud fra disse mål kan der således opstilles scenarier, der kan testes. Programmet er ment som en hjælp med organisering, således at målene for programmet er relateret til klubbens organisatoriske opgaver. Nedenfor ses en liste af mål for den almindelige bruger, hvorudfra der kan udvikles scenarier.

- Følge ens status på duelighedsbevis.
- Se reservationer af både.
- Opret reservation af både.
- Find andre medlemmer.
- Tilmeld sig begivenheder.
- Følg status på begivenheder og undervisning(ændringer/aflysninger).

En administrator af programmet vil have få yderligere mål såsom opdatering af bådstatus og aflysning såvel som oprettelse af klubbegivenheder. Ud fra disse mål er der lavet en række scenarier til user cases, som er vedlagt i bilaget, som kan findes i appendiks G på side 93.

Et eksempel på en user case kunne være: Et medlem af klubben kunne godt tænke sig at komme ud og sejle. Vedkommende logger ind på systemet med sit personlige login. Han navigerer ind på bookingsiden og booker en båd den kommende lørdag. Han logger derefter ud af programmet igen.

¹FiXme Note: myref til userstories

Ud fra disse scenarier kan hver funktionalitet beskrevet omskrives til en række korte opgaver, som berører én bestemt funktion i programmet. Opgaverne kan findes i appendiks H på side 95

Dette bilag vil blive vist til testpersoner af programmet, og de vil blive observeret mens de forsøger at udføre den opgave de har fået. Efter udførelsen af opgaven de er blevet stillet, spørges der ind til brugeres oplevelse af programmet med henblik på funktionalitet, om brugergrænsefladen var forvirrende eller guidede testpersonerne i den rigtige retning og eventuelt andre meninger fra testpersonerne.

Et eksempel på en opgave svarende til eksemplet på user casen fra før ville være:

- Log ind på systemet med følgende brugeroplysninger: INDSÆT BRUGERNAVN og PASSWORD
- Foretag en booking af båden: INDSÆT BÅDNAVN til næste lørdag, med følgende andre oplysninger: INDSÆTOPLYSNINGER
- Log ud af systemet igen

Vi vil imens udfylde et skema med testforløbet, samt skrive kommentarer omkring brugeren færden i programmet hvis noget er bemærkelsesværdigt. Herudover tages der tid på brugernes tidsforbrug på hver enkelte opgave.

I de følgende afsnit beskrives der hvordan disse data vil blive behandlet.

FiXme Note!

2

15.1 Funktionalitet

I denne sektion ses der på produktets funktionalitet og udførelsen af programmets mål for den givne målgruppe. Funktionalitet ser på en overordnet succesrate af programmet og tester således for helt basale fejl i brugergrænsefladen, der resultere i at opgaver ikke bliver løst. Funktionalitet kan udregnes ud fra hvor mange scenarier der er fejlet i forhold til udført. Funktionen for at udregne dette kan ses på ligning (15.1). Her er N antallet af opgaver, R antallet af brugere, n_{ij} er resultatet fra brugeren j ved at udføre opgave i . Her er n_{ij} lig med 0, hvis brugeren ikke klarede opgaven, og det er lig med 1 hvis brugeren klarede opgaven. Funktionaliteten ønskes at ligge over 75 procent.[?]

$$\bar{E} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij}}{RN} \cdot 100\% \quad (15.1)$$

15.2 Effektivitet

Her ses der nærmere på den egentlige udførsel af hvert scenarie og ikke bare om det er udført eller ej. Effektivitet ser på, hvor lang tid en bruger benytter på at blive bekendt med programmet og kunne løse opgaver. Denne værdi kan ses relativt i forhold til hver enkelt bruger, samt sættes

²FiXme Note: Der mangles lige lidt her om hvordan vi vil behandle det data vi får ind, og hvilke data vi får ind

op imod tidsforbrug af en ekspertbruger, en af programmets arkitekter, for at danne et indtryk af hvor let programmet er at gå til. I stedet for at bruges formelen der findes på [?], vil der i stedet findes et gennemsnit af brugernes udførelse af alle opgaverne, og denne værdi sammenlignes derefter med en eksperts tid på at løse opgaverne.

15.3 Tilfredshed

Til sidst laves der en tilfredshedsundersøgelse for at få brugerens meninger omkring programmets design. Dette kan gøres igennem et spørgeskema. Undersøgelsen skal give et indtryk af, hvor godt brugeren synes programmet var i forhold til layout, placering af de krævede objekter i programmet og deres synlighed i programmet. Denne undersøgelse giver et indtryk af, hvor godt programmet er designet, og derved hvor let forskellige funktioner i programmet er at tilgå. Et spørgeskema kan ses i appendiks I på side 101. [?]

I vores produktkrav **var det er** punkt vedrørende medlemmer, hvilket er blevet **tilfredstillende** implementeret i programmet. Medlemmer har fået deres egen klasse, **SailClubMember**, som indeholder de relevante informationer om et medlem. Medlemmerne udgør desuden grundlaget for login-systemet, da der herigennem kan bestemmes hvilken bruger der logger ind, og hvilke funktioner i programmet de skal have lov til at tilgå. Login-systemet fungerer desuden med at brugerne har brugernavne og kodeord. Kodeordene er krypteret med en hashingalgoritme, hvilket er gjort for at gøre dem ulæselige i tilfælde af at databasen skulle blive aflæst af andre.

Et vigtigt punkt i programmet var at sørge for at man kunne oprette en booking af en båd. Dette punkt er blevet gennemført, da der i programmet kan oprettes en booking. Desuden er det indført logik som sikrer at båden skal være ledig og ikke være skadet før at bookingen kan gennemføres. Yderligere er der til en reservation også knyttet en logbog, som skal udfyldes efter sejladsen er gennemført. Udfyldelsen af logbøger blev fremført i interviewet med Jacob Nørbjerg, som værende besværligt i Sejlklubben Sundet, da deres logbøger skal udfyldes på papir. Det menes derfor at programmet her kan gøre det lettere at klubben at holde styr på logbøger, ved at digitalisere dem, holde referencer til hvilken sejltur de hører til, sørge for at ikke alle kan udfylde logbøgerne og selvfølgelig indeholde de krævede informationer.

Et andet vigtigt punkt i programmet var at gøre det muligt at holde styr på undervisningen, både fra elevernes side og underviserens side. I programmet er det blevet konstrueret **et** del som specifikt tager sig af undervisningen, hvorfra underviserne kan notere hvilke hold og elever der har gennemført specifikke opgaver. Tilsvarende kan eleverne se deres egne fremskridt og hvornår de har deres undervisningslektioner. Programmet er på undervisningsdelen fungererende, dog er det specifikke pensum, som sejlklubben Sundet selv står for at udvælge, ikke implementeret. Men det menes at skifte fra det overfladiske pensum som programmet er skabt med, til det faktiske pensum som sejlklubben Sundet anvender, ville være relativt simpelt, og derfor kan programmet også bruges i andre sejlklubber med undervisning, hvilke anvender et andet pensum.

Begivenheder blev også fremsat under listen af produktkrav. **Begivenhedsdelene** er blevet implementeret og det er muligt for brugere selv at oprette, samt tilmelde sig begivenheder. Desuden kan en bruger, når de logger ind på systemet se deres kommende begivenheder. Dog er der som sådan ikke blevet lavet et egentlig kalender, men i stedet en agenda med begivenheder. Agendaen blev valgt i stedet for kalenderen, da et agenda format ville egne sig

bedre til at fremvise bestemte dage med begivenheder, fremfor en kalender hvor alle dage, med og uden begivenheder, ville blive vist.

Ses der på problemformulering i forhold til den færdige løsning, kan det argumenteres for at der er blevet udviklet et it-system, som digitalisere en stor del af det dokumenthåndtering som foregår i klubben, såsom undervisningsfremskridt for hver elev, booking af både, udfyldelse af logbøger, oprettelser af begivenheder og visning af begivenheder. Dog er målet om at understøtte frivilligt arbejde ikke blevet opfyldt til fulde, da der er blevet lagt mere fokus på elev, undervisere og almindelige medlemmer i programmet. Programmet er dog godt til at håndtere al den dokumentation som påkræves af sejlklubben Sundet vedrørende sejlture og undervisning. Desuden hjælper programmet også med at sikre at alle nødvendige informationer bliver skrevet ind, og ikke bliver glemt.

Konklusion

17

I det initierende problem blev der fundet frem til, at der skulle laves en softwareløsning, som kan hjælpe frivillige i fritidsklubber til administration og som er let at benytte. Der blev afgrænset i afsnittet Fritidsklubber til at arbejde med sejlklubber, fordi det blev vurderet at det var lettere at udarbejde et program, ud fra en bestemt type fritidsklub. Der blev yderligere afgrænset, i interessentanalysen, til at arbejde med Sejlklubben Sundet, for at gøre processen med at udarbejde et program endnu lettere. I organisationsafsnittet blev Sejlklubben Sundet undersøgt bl.a. for at få afklaret deres administrative opgaver. I teknologianalysen blev mulige teknologier til en softwareløsning undersøgt og under "State of the Art" blev andre typer af bådsystemer undersøgt. Ud fra analysen, blev der skabt en problemformulering, som fokuserer på frivilliges arbejde i fritidsklubber og hvordan de kan hjælpes i deres arbejde. Efter udarbejdelsen af problemformuleringen, blev kravene til produktet undersøgt, som skulle bruges til det videre forløb, med at udarbejde et program.

Perspektivering 18

I dette afsnit vil der blive beskrevet forskellige funktioner, som der er blevet vurderet nyttige hvis projektet fortsat skulle udvikles på. Der vil desuden blive diskuteret generelle overvejelser af programmet, som kunne have været udformet anderledes.

Der har på gruppen været diskuteret, at en SMS-service ville være smart at bruge. Denne service kunne bruges til at sende notifikationer på dagen eller dagen før et medlems booking af en båd finder sted. Det kunne også tænkes, at hvis man vil booke en båd i et tidsrum, hvor båden allerede er booket, kan man melde sig på som næste booker. Det betyder at hvis medlemmet, der har den reelle booking pludselig aflyser, vil den næste booker dermed få en sms om, at nu er båden altså fri til at blive booket i det tidsrum. Her kunne der evt. bruges en svarmulighed, så medlemmet, ved at svare på sms'en, kan melde om de vil overtage bookingen eller ej, dvs. uden at skulle åben systemet op. Et lignende system kan konstrueres ved e-mail notifikationer.

Som programmet er nu fungerer ingenting online, og det vil sige, man skal ned i klubben for at kunne booke en båd, eller tilmelde sig begivenhederne i klubben, samt for at tilgå alle de andre funktioner i programmet. En måde at ændre dette på kunne være at lægge databaseforbindelsen online, så der kan laves ændringer i databasen uden at logge på nede i sejlkлубben. Dette ville gøre programmet meget mere fleksibelt, samt hjælpe medlemmerne og gøre brugen af klubbens faciliteter lettere at tilgå.

Som nævnt i analysen vil programmet med nogle ændringer, kunne bruges i andre sportsklubber. Et eksempel kunne være en sportshal, som vil gøre det muligt at booke hallen til diverse arrangementer, melde sig til begivenheder der måtte foregå og lignende. Dog vil dette som nævnt tidligere kræve ændringer i programmets funktionaliteter, da løsningen fremstillet til projektet her, er specifikt lavet en sejlkлуб med sejlerskole, med udgangspunkt i Sejlkлубben Sundet.

Endeligt er det blevet konkluderet at et system som dette fremstillet til projektet her, ville tjene sig bedre som værende en hjemmeside. På denne måde kan tilgå funktionerne via ens webbrowser, og tilgå databasen igennem denne hjemmeside. Hermed undgår man, at brugerne skal installere programmet, som de ellers ville være nødt til, hvis det nuværende program brugte en online database forbindelse, som nævnt tidligere. Men da det var angivet af projektoplægget at man skulle skrive projektet her i C#, var dette ikke en mulighed, selvom gruppen klart så dette som værende den bedste løsning.¹

FiXme Note!

¹FiXme Note: Stadig muligt med C#, se diskussion under WPF-afsnit

Del C

Referencer

Figurer

3.1	Metode for Kontekstmodellen	7
10.1	Demonstration af WPFs Controls	33
12.1	Person	37
12.2	SailClubMember	37
12.3	StudentMember	38
12.4	Team	39
12.5	SailTrip	39
12.6	Eventklasse	41
12.7	Logbook	41
14.1	DateTimePicker	46
14.2	FrontPage-Usercontrol	47
14.3	Boat screenshot	48
14.4	Login interface	49
14.5	CreateBoatbookingWindow	50
14.6	CreateCrewWindow	52
14.7	CreateLogBookWindow	53
14.8	UIStudyTeacher	56
14.9	UIStudyStudent	57
14.10	UINewLecture	57
14.11	UINewTeam	57
E.1	UML-diagram over klasserne i projektets program	91

Tabeller

14.1	Tabel over alle brugerniveauer og deres tilladte funktioner.	45
------	--	----

Del D

Appendiks

Informationer påkrævet af sejlkлубben Sundet



A.1 Data på sejlladser og medlemmer

Dette bilag har til formål at beskrive de information Sundet påkræver af de sejlere der sejler fra deres klub. Herudover er der også informationer omkring hvilke informationer sejlerskolen Sundet skal bruge på deres elever for at kunne lade dem bestå.

Når en skolesejllads skal foretages skal følgende informationer skrives ned skoleprotokollen:

- Navne på læren og eleverne
- Ugedag
- Dato
- Fremmøde, i form af X for mødt, A for afbud
- Navne på gæster
- Afgang og ankomsttider
- Vind forhold
- Evt. Kommentarer, normalt henvisninger til skadesrapporterne, hvis sådanne er opstået

Alle disse informationer skrives for hver skolegang, og arkiveres i klubhuset, dvs. det er ikke elektronisk.

Når der sejles uden for skolen, dvs. fornøjelsesture, eller kapsejladser skal lignende information udfyldes, dog med visse ændringer:

- Bådnavn/id
- Navn, telefonnummer for føren af båden
- Navne på alle i besætningen, evt. telefonnummer
- Startid og dato
- Forventet ankomstdato og tid
- Formål med turen, med indikation af område for sejladsen
- Reelle ankomstdato og tid
- Vind og vejrforhold
- Kommentarer

Hvis der sker skader på en båd under en sejllads skrives disse ned i skadesrapporten. Føren skal også læse op i denne bog før en sejllads for at sikre at man er klar over evt. skader på båden, som man skal være klar over. I denne bog skrives flg. oplysninger:

- Dato
- Tekst der beskriver uheldet, eller skaden
- Her kan også skrives evt. svar fra bådchefen(Personen der sørger for båden)

Til at holde styr på medlemmernes oplysninger, bruger Sundet et Microsoft Access baseret system. Her er der personlige oplysninger på eleverne og medlemmerne, samt kan der printes girokort ud til håndtering af kontingenterne. Det bruges også til at holde styr på bådene i havnen som er ejet af nogle af klubbens medlemmer, samt deres lokationer. Dette system kan kun tilkobles lokalt på computeren lokaliseret i klubhuset.

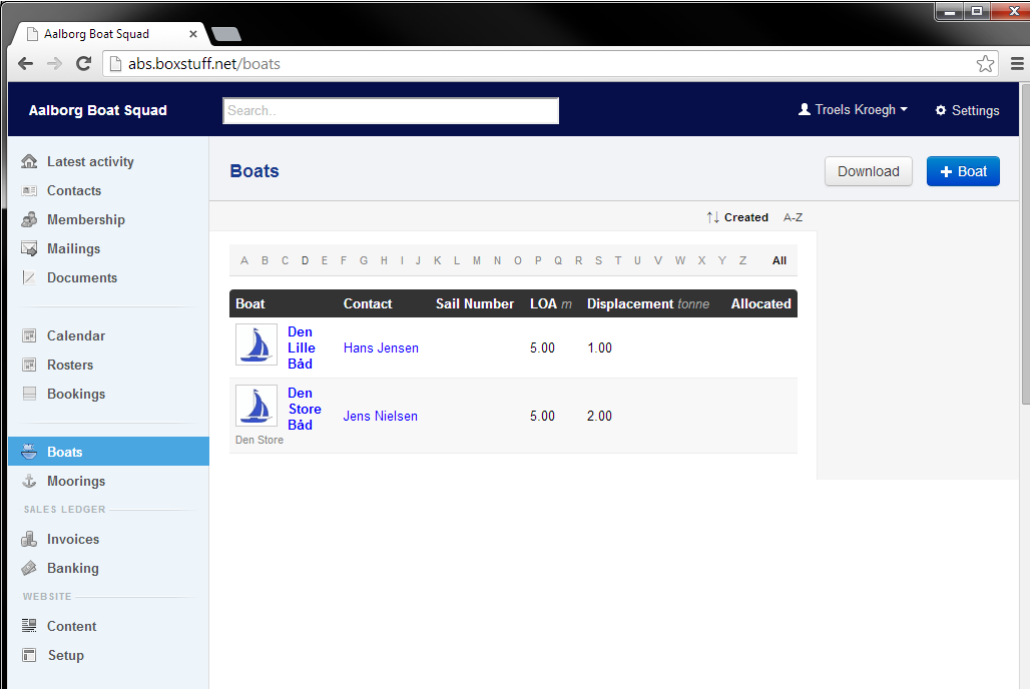
A.2 Informationer og andre events

Når Sundet skal give informationer ud til medlemmerne, sættes disse op på en opslagstavle. Disse informationer kan f.eks. være tilmeldinger til diverse begivenheder såsom 24-timers sejladser eller onsdagsmatcher. Dvs. for at kunne tilmelde sig begivenheder i klubben skal man skrive sig på det bestemte tilmeldingsskema man finder i klubhuset. Det er ikke sikkert at medlemmer har tid eller når at opdage tilmeldingsfristerne på diverse begivenheder, hvilket kan resultere i et lavere fremmøde, og mindre aktivitet i klubben.



Screenshots af SailingClubManager B

Dette bilag har til formål at underbygge afsnittet kapitel 6 på side 17. Dette er et system, til administration af både, som vi har fået tilladelse til at prøve af skaberen. Siden hedder SailingClubManager, og er placeret på url'en <http://abs.boxstuff.net>, "abs" da vores klub derunder hedder "Aalborg Boat Squad". Det er også muligt at anvende denne service til at lave sin egen hjemmeside til offentligheden, et eksempel på dette er: *Thames Sailing Club*.

Siden /boats indholder de både, som er tilføjet af brugeren.

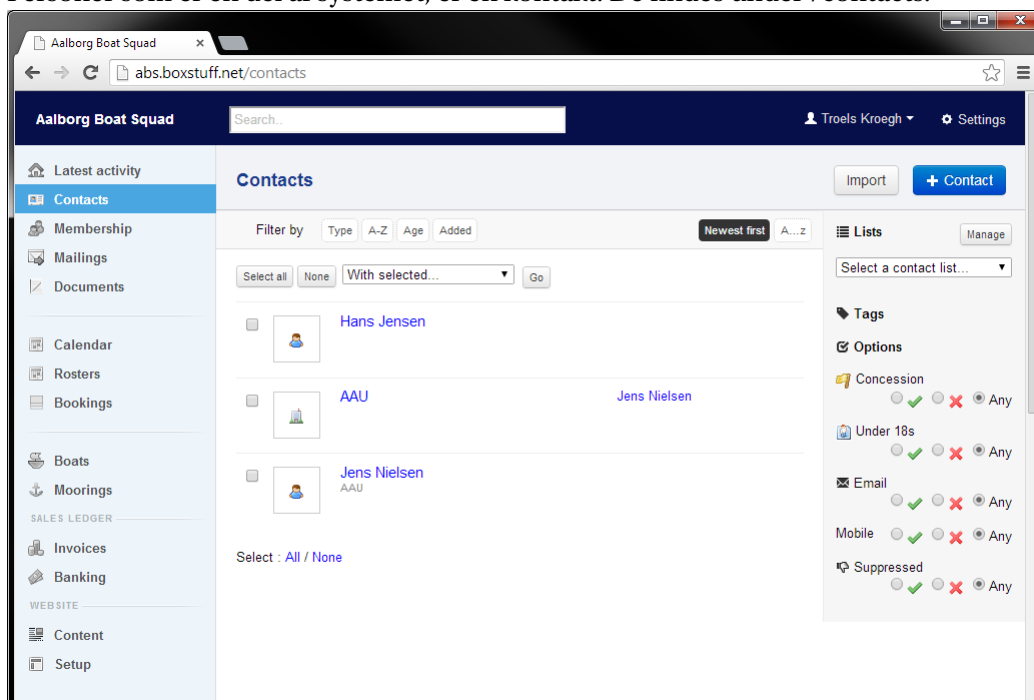


The screenshot shows a web browser window with the address bar displaying abs.boxstuff.net/boats. The page title is "Aalborg Boat Squad". The left sidebar contains a menu with the following items: Latest activity, Contacts, Membership, Mailings, Documents, Calendar, Rosters, Bookings, Boats (highlighted), Moorings, SALES LEDGER, Invoices, Banking, WEB SITE, Content, and Setup. The main content area is titled "Boats" and includes a "Download" button and a "+ Boat" button. Below the title, there is a table with columns: Boat, Contact, Sail Number, LOA m, Displacement tonne, and Allocated. The table contains two rows of data:

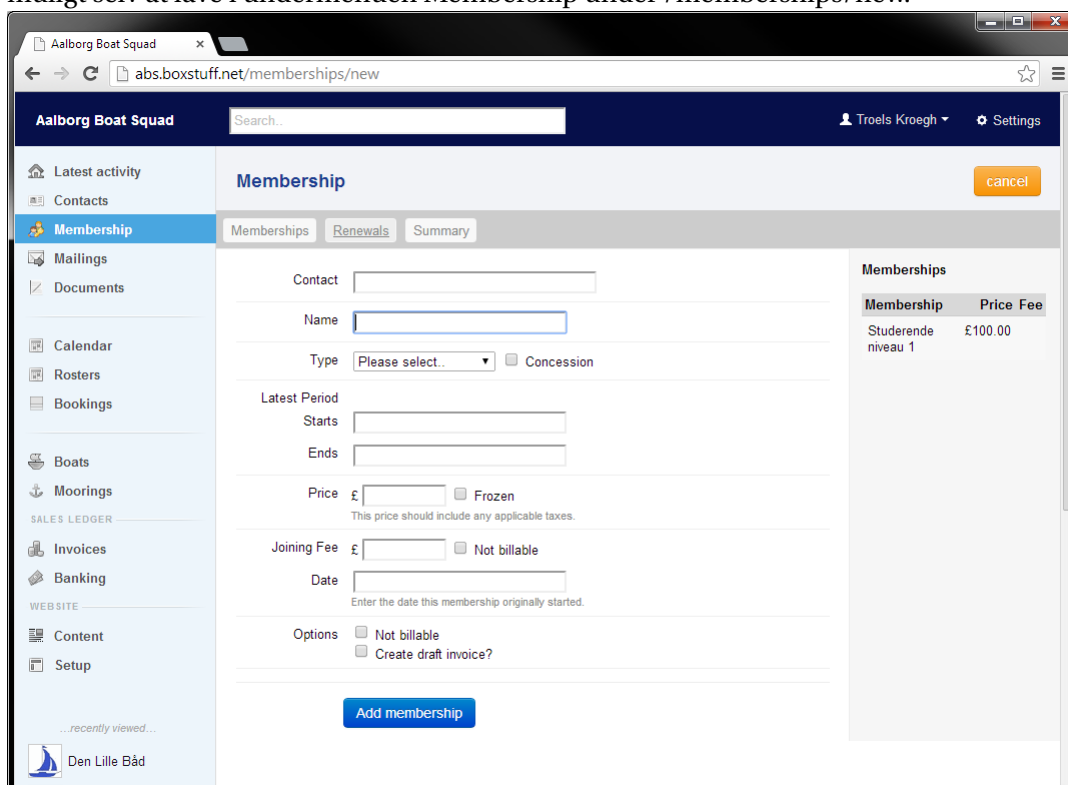
Boat	Contact	Sail Number	LOA m	Displacement tonne	Allocated
 Den Lille Båd	Hans Jensen		5.00	1.00	
 Den Store Båd	Jens Nielsen		5.00	2.00	

Below the table, the text "Den Store" is visible. The top right of the page shows the user "Troels Kroegh" and a "Settings" button.

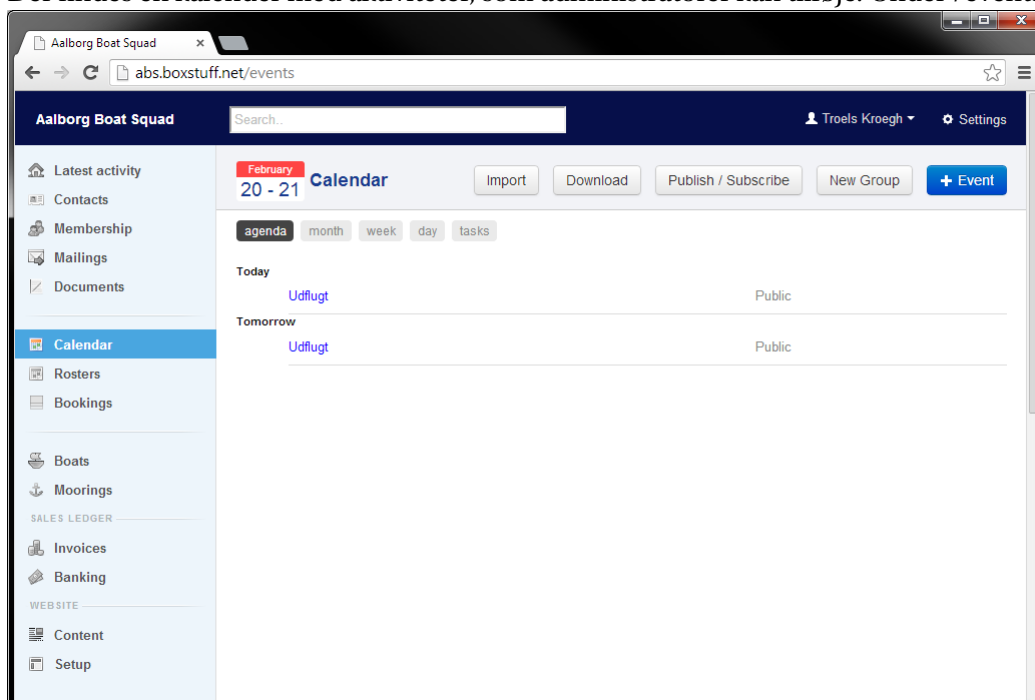
Personer som er en del af systemet, er en kontakt. De findes under /contacts.



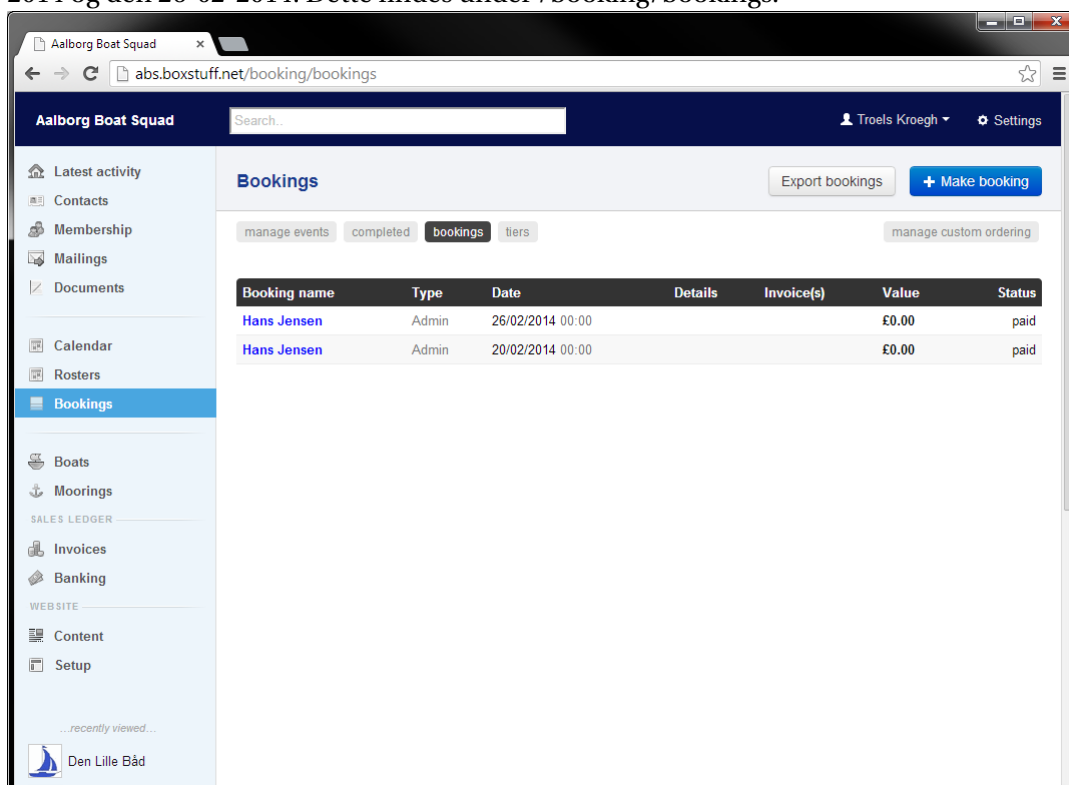
Det er muligt at tilføje nye medlemmer, herunder hvilke type medlem de er. Disse typer er også muligt selv at lave i undermenuen Membership under /memberships/new.



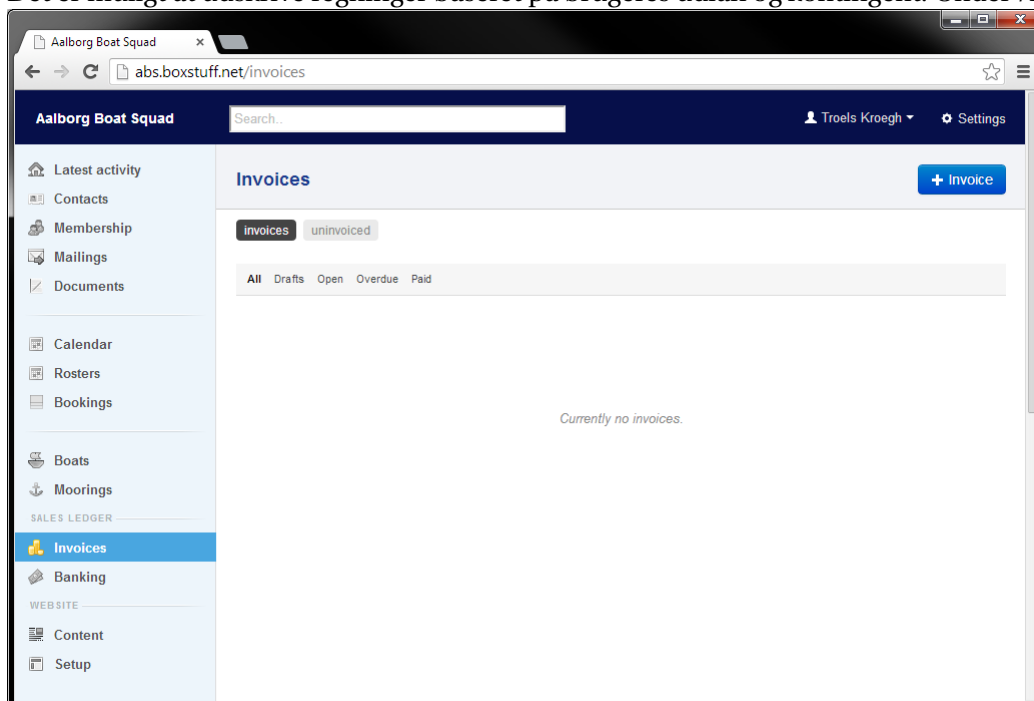
Der findes en kalender med aktiviteter, som administratorer kan tilføje. Under /events.



Det er muligt for brugere at booke bådene, her har Hans Jensen booket en båd både den 20-02-2014 og den 26-02-2014. Dette findes under /booking/bookings.



Det er muligt at udskrive regninger baseret på brugeres udlån og kontingent. Under /invoices.



Interview

Som en del af informationsindsamlingen til projektet, er der blevet lavet et kort interview med Jacob Nørbjerg, tidligere skolechef i sejlkлубben Sundet. Ideen i interviewet var, at få viden omkring hvordan en sejlkлуб administreres direkte fra en person, som havde haft med administrationen at gøre på nærmeste hold. Dog kan der være forældede informationer i interviewet, da Jacob Nørbjerg ikke længere er skolechef og ikke har været det i et par år. Udover selve interviewet med Jacob Nørbjerg, blev der udleveret dokumenter, som beskriver nogle bestemte administrative strukturer i sejlkлубben nærmere, disse dokumenter refereres der også til i den følgende tekst.

På baggrund af interviewet med Jacob Nørbjerg findes der nu informationer om hvordan en større sejlkлуб, Sundet, med udlån af både og sejl-skole administreres. Interviewet var delt op i to sektioner, den første sektion med fokus på, hvordan administrationen fandt sted da Jacob var klubleder, og den anden sektion med fokus på problemerne ved administrationen.

Administrationen i klubben

Administrationen af sejlkлубbens medlemmer foregik på et hjemmelavet IT-system, som holdte styr på informationer omkring medlemmerne, som navn, adresse, telefonnummer, medlem-nummer, om medlemmet var bådejer eller ej. Systemet kunne desuden håndtere indbetaling fra medlemmerne, for fx lån af værktøj til vedligeholdelse af bådene, og organisationen af bådene når de lå på land, altså pladsuddeling. IT-systemet var Access baseret, en udgivelse fra Microsoft, brugt til at skabe brugervenlige databaser uden programmeringsviden. Ud fra både interviewet og de udleverede dokumenter, blev den del af administrationen, som ikke foregår elektronisk forklaret. Ud fra dette kunne det ses, at sejlkлубben har faste procedurer i henhold til bådudlån og skolesejls, som alt sammen foregår på papir. I en logbog noteres udlån af bådene, med informationer vedrørende hvem der sejler, hvorhen, forventet tilbagekomst, og noter om båden, som alt sammen skal udfyldes af personen, som låner båden. Desuden skal en havarilog udfyldes, også på papir, vedrørende eventuelle skader på fartøjet som følge af turen. En sejlsprotokol skal også udfyldes i tilfælde af, at der har været undervisning, med informationer vedrørende fører, dato for sejlsen, hvilket elever der var med og eventuelt hvilke gæster som var med. Og med hensyn til reservationer fandtes der lister med fartøjerne, hvor det efter først til mølle princippet gjaldt om at skrive sig på først. Alle de informationer kunne kun udfyldes på papir i sejlkлубbens klubhus, og det var derfor nødvendigt for medlemmer og elever at tage til klubhuset, for at undersøge om hvilke både der var ledige, og om de var sat på

et hold, som skulle ud og sejle osv.

Problemer ved administrationen

Problemerne ved administrationen var netop, at informationerne vedrørende hvilke både der var ledige, og hvornår der var undervisning, befandt sig alt sammen i klubhuset, og ikke var tilgængeligt fra nettet. Desuden var det et større arbejde, når der skulle udskrives regninger, da listen med udlån skulle gås igennem, for at se hvilke personer der havde lånt både, og hvor mange gange de havde lånt bådene, det samme var gældende for krydstjekning af antallet af undervisningstimer for eleverne. Alle former for tilmeldinger og reservationer foregik også på papir, hvor først til mølle princippet gjaldt uden mulighed for at tjekke i forvejen, om hvad der var frit/havde frie pladser.

Ud fra interviewet kan det konkluderes, for sejlklubben Sundet, at de mangler et bredere system, som ville kunne indeholde alle de informationer de i øjeblikket skriver ned i diverse protokoller og log-filer. Problemet med den måde de gør tingene på, er at de tvinger folk til at tage ned til klubhuset for at finde de informationer de søger, og samtidigt bliver arbejdet med at krydstjekke informationerne fra diverse papirer et større arbejde, i forbindelse med udskrivning af regninger og elevernes undervisningstimer.¹

FiXme Fatal!

¹FiXme Fatal: Dette skal nok tages op i hoveddelen.

Interview transskribering



Jacob: I skal huske at det her baseret på muligvis forældet viden.

Alle: Jaja.

Jacob: Hvad mener i med at administrere i sejlklubben? Det er jo et bredt spørgsmål.

Marc: Det er primært hvordan i holder styr på de ting som ligger under sig.

Jacob: Vi har et hjemmestrikket medlemssystem. Det var jo den gang ikk'. Der ligger alt det sædvanlige ikk': Navn, adresse, telefonnummer. Så ligger der fuldt medlemsnummer og så ligger der ens uddannelse, altså sejlskole, hvor langt man er kommet i forløbet på sejlskolen og om man har førebrevet, duelighedsbevis. Så ligger der om man er bådejer, det koster nemlig flere penge. Der ligger vidst også noget om landpladsadministration, alt hvad der hedder vand, plads i vandet, der organiserer havnen. Plads på land det arrangerer klubben. De klubber har forskellige landområder vi disponerer over. Det er så det generelle, det vi har på medlemmerne. Så kan man, f.eks. hvis man som medlem har været... Hvis man skal betale for et eller andet, hvis man har lånt en båd og skal betale for det, så er der også et skærmbillede til det. Jeg kan ikke lige huske om man klikker direkte fra medlem eller man skal indføre medlemsnummeret der, men der er et sted hvor man kan gå ind og sige: "Her er en der skal betale noget. Han skal betale for så og så mange dages leje af en båd og en motor osv. osv. og så printer vi en girokort ud." Det samme typisk skærmbillede bruger vi til: "En som skal betale for en landplads og hans båd fylder så og så mange kvadratmeter dut [efterligner klikkelyd]". Så kan man skrive her ind, som jeg gjorde i weekenden, lånt en af klubbens slibemaskiner med støvsuger og det koster så og så meget pr. dag, så meget af det ligger i et pr. accessbaseret tudsegammelt hjemmestrikket system som ligger på computeren nede i klubben.

Søren: Ligger det kun lokal eller hvad?

Jacob: Ja, det ligger kun der lokalt og der er forskellige record udtræksningsmuligheder os' ikk': Gi' mig en liste over alle elever på andet år, gi' mig en liste over alle bådejere eller gi' mig en liste over hvilken rækkefølgen både skal sættes op i og sådan noget, så det er der også. Så har vi selvfølgelig senere, men det kan i se på nettet, så har vi jo indført forskellige... Altså meget af det information som vi kan trække det bliver så på forskellig vis smidt op på nettet. Det jeg sagde lige før med liste over hvornår både skal op den ligger på nettet, og den kan alle og enhver gå ind og se, så i kan gå ind og se hvad min båd hedder og hvornår den skal i vandet. Det ligger bare som en PDF. Nå, hvad har i på jeres medlemmer? [oplæsning af spørgsmål, spørgsmål var

givet på forhånd] Der er sagt, altså vi har typisk om de har eller ikke har førebrevet, om hvor langt de er kommet i skolen, hvis vi altså husker at vedligeholde det, det er ikke altid vi gør det. Det er vel sådan cirka det.

Hvor mange både har vi til rådighed? [oplæsning af spørgsmål]

Ja det svinger meget. Den gang jeg var skolechef der havde vi 2 gaffelrigger, 3 drabanter og en spækhugger. Nu er konfigurationen ved at ændre sig lidt så nu hedder det stadig to gaffelriggere og så hedder det vidst nok to eller tre spækhuggere, er de ved at lave det om til og så nogle nye som hedder J80, sådan nogle satans små flyvepap. Jeg tror de unge mennesker kommer til at få nogle forskrækkelser når de skal ud og sejle dem første gang hvis de ikke har sejlet før, men det er jo deres problem. Det er jo en politisk diskussion. Og så har vi en enkelt yngling. Det var en som blev til overs engang, den er bare blevet foræret til klubben. Den er til at lære til at lege i og så har faktisk en fire... men de er ikke rigtig taget med der, vi har en fire til seks Mini12'er, hedder de. Det er til handicapafdelingen. De ligner gammeldags Americas Cup både som de så ud den gang de lignede sejlbåde, skaleret ned i en to-tre meters længde, så kan man sidde nede i dem og styre med hænderne eller fødderne eller hvad man har at bruge, så den bruger handicapafdelingen til når handicappede er ude og sejle.

Et EDB-system? [oplæsning af spørgsmål]

Ja, i den forstand at vi har overblik over hvilke både vi har. Så er der kommet det der online reservationssystem, men den situation jeg kendte, som jeg tror stadig ligger bagved, fordi der kan man kun reservere en båd, det er jo at der inde i skolestuen, dvs. det er der hvor der ligger redningsveste og grej til alle bådene og sådan, der ligger der en stor bog hvor man går ind og skriver "3. maj, sådan og sådan, Jacob, besætning, telefonnummer på Jacob, hvornår sejler vi, hvornår tror vi at vi kommer hjem, hvor tror vi nok vi tager hen" og så et eller andet sted, en kolonne som hedder: Det var her vi kom hjem, og hvis der er nogen kommentar, så det er sådan en stor bog af den tykke der, den fylder sådan her [viser hvor står bogen er]. Så er der på hver båd en havariprotokol, dvs. at hvis der er et eller andet med båden, så tager man den rigtige bog frem og skriver: "Jeg var ude og sejle med båden og jeg smadrede ind i Oslobåden" eller et eller andet, der var noget som knækkede, sejlede revnede eller hvad det nu var og hvis man ikke kan reparere det selv så skriver man det der. Hvis man føler eller ved at det er noget som bare skal repareres i en helveds fart, så er man stærkt opfordret til at ringe til skolechefen eller den der har ansvar for båden. Hver båd har en bådchef der er ansvarlig på den båd, så hvis man kan se at det her kræver mere end jeg lige kan klare, det kræver noget værktøj, reservedele, det kræver måske syn af en bådbygger, så skal man så ringe til den pågældende person som så vil sætte det i værk. Hvis man kan reparere det selv så prøver man også på at gøre det, men bare lige skriv et notat om at der var noget eller hvis det er noget som skal laves, men det kan godt sejle uden det laves så skriv det lige.

Så er der skolen: Hver båd hver aften har sådan et fortrykt ark: Fører, elever, sejlsadsnummer, dato, afkrydsning, hvem og hvad. Ud fra hver, mulighed for bemærkning: Skete der et eller andet, hvad gjorde vi, er der nogen som ikke dukker op, er der gæster med så bliver de også bare skrevet på der nedenunder den faste liste, og hvis det er en anden fører skriver man også det på, så krydser man af den aften det handlede om og det brugt man tildels til at holde øje med hvem der kommer, hvem der ikke kommer, hvem som husker at melde afbud, og glemmer det, det er fyfy at glemme at melde afbud, og tælle op antal gange man har sejlet, det er jo også

vigtigt, altså i forhold til proportion på skolen, det var sejladsprotokollen til skolen.

Så har vi selvfølgelig reservationssystemet, som flytter lidt der, men der er stadig sådan et andet slags reservationssystem som handler om når skolen arrangerer noget. Når skolen siger: "Nu på første weekend i juni der er der 24-timers sejlads, det er en skide god oplevelse. Er der nogen elever der gerne vil være med?" Så hænger der gerne sådan et stykke papir nede i gangen hvor der står 24-timer sejlads, her kan førerne skrive sig på og her kan eleverne skrive sig på og så må man håbe at der er nogen som skriver sig på. Så det er sådan en: "Her er et tilbud om en ekstratur af en eller anden slags, skriv jer lige på her." Det foregår også på papir. Så det kræver altid at man kommer der ned og gør noget, skrive sig på, sørge for overblik. Så det der *pullik 9:11* der hedder "Hvad er jeres største problem", det største problem jeg oplevede var simpelthen det her med at holde styr på... Som administrativ ansvarlig, så er der bare en fandens masse papir som man skal holde styr på. Man skal ind og tælle op på den der mødeliste, man skal... Den gang skulle man hvis nogen ringede ned og sagde kunne jeg reservere en båd på det og det tidspunkt, så skulle man ud og finde... Vi havde gerne sådan en kalender hængende nede på opslagstavlen med bådene og datoerne, så kunne man så blive krydset af, enten gjorde folk det selv eller så bad de os om at gøre det. Havde anden kalender bl.a. sedel som hang der med reservation til onsdags-kap-sejlads, sådan at folk "Jeg vil gerne ud og sejle onsdags-kap-sejladsen i næste uge" så skriver man sig lige på, "jeg har reserveret båden" og den der første når frem med en kuglepen har vundet og enten skriver man "jeg har besætning" eller "jeg har plads til to elever" eller hvad det nu er og så må folk finde ud af det på den måde. Det kræver igen at man kommer ned og kigger på opslagstavlen og forstår det der. Der var en regel omkring onsdagsmatch med at man kun kan reservere en ude ud i forvejen. Det der med at man bare lige, nogen af de gamle, der er altid nogen som sejler onsdagsmatch rundt i vores både de tager blokreservation hen over hele sommeren, nænæ. Man kan reservere en ude i forvejen. Det er jo en politik, som man har.

Det er meget papirarbejde. Når man skal finde ud af hvor meget der skal betales, så skal man ind i den store [bog] og så skal man ind og se hvem har egentlig lejet en båd og så gå igennem, og så skal man jo prøve på... Typisk så gør man det måske tre-fire gange i løbet af en sommer, maks. Dvs. at man skal finde alle de gange hvor den person har haft fat i båden, og hvad er det: En aften, en dag, en weekend, hvad fanden er det for noget?

Marc: Krydsrefereres om der er elever med?

Jacob: Ja det kan man... Det er vel den nye politik, så skal man også finde ud af om der er elever med. Det tror jeg nok vi havde en... [politik]. Så indførte vi en da det kom så indførte vi med at man lige kunne lave et kryds på besætningen og sige at man har en elev med eller skrive et sted at man har en elev med, så skal man også lige finde ud af det. Så gik vi ikke mere ned i det end at vi stolede på, at hvis man skrev at man havde en elev med, så stolede vi på det. [eksempel:] Nå det var så Anette, hun har jo sejlet der, nå er han blevet stillet ind: fint. Og så gå igennem sådan 2-3 sider, der bliver hurtigt fuldt op på sådan en side så, fandens besvær, derfor gad vi heller ikke gøre det til sidst. Og så blev der skrevet girokort ud som blev puttet i en kuvert og sendt til den pågældende. Nogen af os kunne godt finde ud af at fiske girokortet over i en PDF og så bare for det sendt på mail, men som regel blev det noget med en kuvert. "Ikke nogen elektronisk opkræven her."

Hvis vi går sådan lidt længere ud... Så omkring det hele med skolen, men det er nok lidt

voldsomt, der synes jeg at det havde været et kæmpe puslespil det med at få styr på hvilke elever skal være hvor og hvilke dage og hvordan får vi placeret dem på både osv. Men problemet var ofte lige så stort... Skyldtes lige så meget at folk bare ikke fik meldt tilbage når vi skrev: Nu må i godt lige få sendt jeres ønsker. Når vi skriver til folk: "Nu skal vi lige have af vide: Om du har tænkt dig at sejle til sommer, om du sidder over et år eller hvad du gør, hvilke dage du kan sejle." Og hvis de ikke kommer så sidder man bare der... Man kan ikke lave planen. Det synes jeg at vi brugte meget tid på, vi havde ikke rigtig noget støtte til at lave den der [plan], det blev sådan et Excel-regneark til sidst, med båd og elever og så blev det bare hængt op på opslagstavlen: Sådan her ser det ud. Det var sådan lidt... Det virkede. Som elev og som medlem synes jeg at det største var det besværlige med man kan bare ikke få overblik over en skid hjemmefra. Altså hvilke både er ledige? Det kan så se med det nye system som de har lavet, men jeg kunne virkelig godt tænke mig at sejle 24-timers sejlads. Der skal man altså have fat i *et eller andet 13:58* nede i Sundet og bede om: Er den der indkaldelse kommet op og hænge. Jeg vil gerne ud og sejle onsdagsmatch, der kan man så gå igen. Jeg vil gerne faktisk finde ud af om der er en ledig plads på en onsdagsmatchbåd. Altså, en ting er om den er reserveret, en anden ting er om der lige er en er en... "Kan jeg lige springe på som besætning her." Så den der manglende afgang til information når man sidder der hjemme og gerne vil et eller andet og så det der med at det hele er papirbaseret. Skulle gå igennem en stor protokol for at finde ud af hvad folk skal betale for et eller andet, det må da kunne gøres smartere. Min vision var den gang noget integration mellem de forskellige ting, mellem brug af både til skole, til fritidsbrug, til kapsejlads, til skolearrangeret fritidsbrug. Altså der er forskel på at jeg beslutter tage ud søndag og sejle med nogle venner og så at skolen siger at den søndag er der et arrangement som skolen gerne vil ha' at kommer ud, så der laver skolen nærmest en *for-et-eller-andet 15:17*, som så først bliver oplyst måske tre dage før at man finder ud af at der kommer noget, så bliver båden frigivet. Så der er det igen med at man som medlem kan gå ind og se: Kan man komme til der. For det er forskellig måder at reservere og bruge en båd på som godt kan være lidt indviklet at finde rundt i.

Som medlem kan man leje og låne et fartøj? [oplæsning af spørgsmål]

Ja det kan de sagtens. Det ved i godt. Kravene er førebeviset. Omkostninger: Vi sender et girokort. Administreres: Ja det gør de jo stortset ved at vi har de der papir, protokoller hvor man skriver sig i.

Søren: Det der førerbevis det får man bare igennem skolen på de to år som det tager eller er det kun duelighedsbeviset?

Jacob: Altså det tager to år på skolen, det er på den måde vi organiseret det på. Det er et valg man har truffet i klubberne der nede. Jamen det tager to år og faktisk har de tre klubber en lidt forskellig policy omkring det, men i Sundet har det været sådan, *der siges et eller andet 16:35*, det var sådan at det første år sejlede man med yoghurtbærger, undskyld plastikfiberbåde og det andet år sejlede man gaffelriggere. Og det ud fra sådan rent proportion, altså glasfiber, sådan en drabant der, den skal men for det første være rigtigt ond ved den før den vælter og for det andet så er den forholdsvis nem at sejle og man får meget klar og hurtig besked hvis man gør noget forkert, båden giver meget hurtig tilbagemelding. Gaffelriggeren er svær, den er tung, den er længere om at reagere, den gir' ikke så hurtig og klar besked om hvad der foregår, dvs. at man skal være mere erfaren for at forstå hvad den fortæller en. Så den prøver vi på ikke at skræmme førsteårseleverne i. Alene mængden af torvværk det ku' godt få folk til at blive bekymrede når de sådan skal finde ud af det. Set enkelt er det en fantastisk båd at undervise i, den har et

stort åbent...[plads] Man kan gå rundt nede i den. Man sidder ikke sådan klemmt sammen på smalle bænke, man kan gå omkring, fedt. Så det tager to år. Så har vi, nu glemte jeg at sige før med Spækhuggeren, jeg sagde vi havde en Spækhugger, vi har skolen 2 år. Så har vi det der hedder... Det er så blevet en kapsejladsskole, sammen med de andre klubber i havnen, så folk der er blevet uddannet fra sejlkлубberne, kan så tilmelde sig kapsejladsskolen, dvs., det er mest spækhugger vi gør det i, dvs. så kommer de... Så for det noget teoriundervisning og noget praksisundervisning i kapsejlads på forskellige niveauer kulminerende med at man så kan deltage i almindelig kapsejlads. Det er ikke noget man skal, lige så snart man har førerbeviset, så kan du gå ud og sejle kapsejlads, det er slet ikke det, men der er mange der godt vil have den der uddannelse, forstå hvordan man *et eller andet med start 18:33*, hvordan lægger man taktik, hvordan får man samarbejdet til at fungere, til kapsejlads skal det gå temmelig stærkt, så det har vi så sammen med de andre, det er så os klubben, så spækhuggeren, den ene og muligvis også den næste, er mere eller mindre fastreserveret til kapsejladsskolen mandag, tirsdag og onsdag tror jeg det er. Så det er jo også en anden fast ting, altså hvor vi har... Faktisk bliver de jo... Den protokol de udfylder er den samme som skoleprotokollen, den er fortrykt med navne på, afkrydsning, du er på dette kapsejlads hold, på den båd om mandagen. Så må du krydse af når du kommer. Og det her med at vi hvem der er på båden det er jo i forhold til skolen, i forhold til skolens regler for om man er dygtig nok når man har gjort nok, og det er helt generelt at vi vil vide hvem som er på båden, altid, når den er ude og sejle, det er rent sikkerhedsmæssigt. Så det har det dobbelte formål.

Søren: Ja det var...

Jacob: Har jeg fået svaret på alle jeres spørgsmål? Jeg kørte bare der ud af.

Alle: Ja det tror jeg.

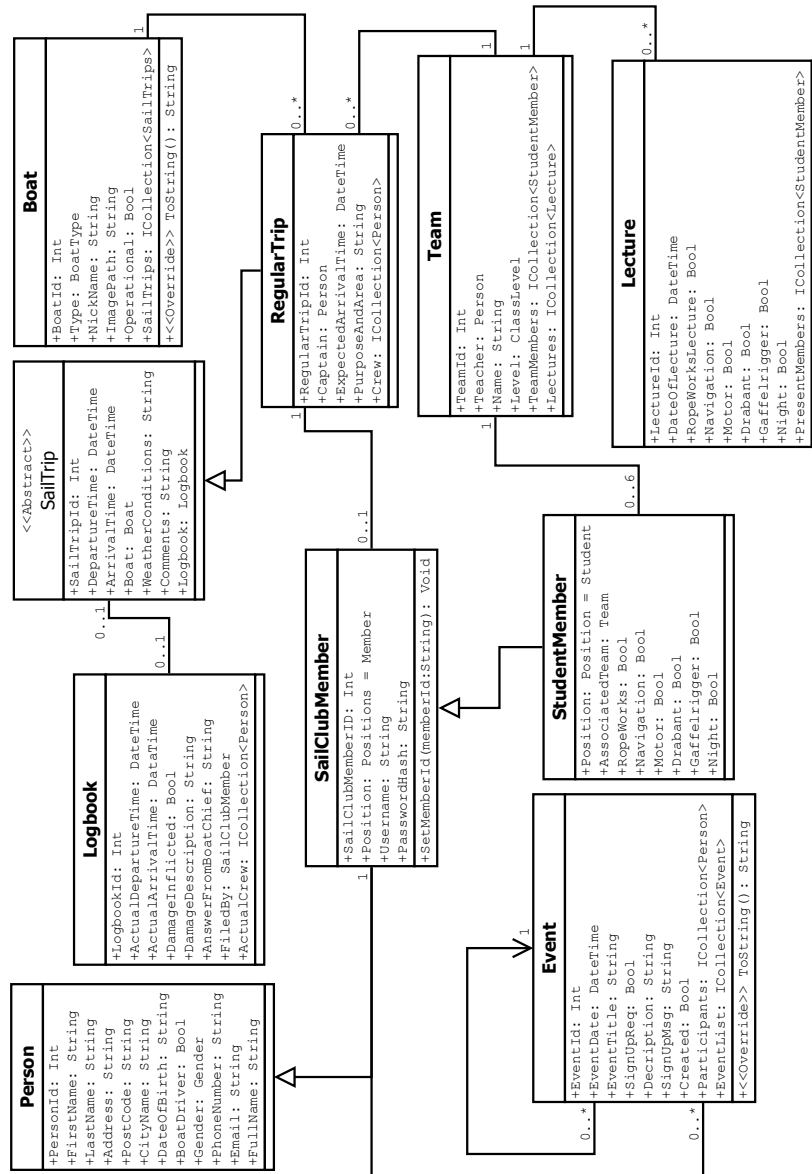
Interview Spørgsmål



I dette afsnit ses de spørgsmål interviewet med Jacob Nørbjerg bygger på.

- Hvordan administrere i sejlklubben lige nu (medlemmer, fartøjer og lign.)?
- Hvilken information har I på jeres medlemmer (kvalifikation, niveau, erfaring eller lign.)?
- Hvor mange fartøjer har klubben selv til rådighed (sejlbåde, joller, ynglinge m.m.)?
- Benytter I et EDB system til at administrere klubbens egne fartøjer?
 - Hvis ja, hvilke egenskaber har systemet og hvordan kunne det forbedres?
 - Hvis nej, hvordan administrere I så fartøjer, og er I tilfredse med denne metode?
- Hvad er jeres største problem mht. administration af fartøjer?
- Hvor meget tid anvendes der på administration af fartøjer (regninger, planlægning, vedligeholdelse osv.)?
- Har I selv en vision om hvordan et system kan gøre det lettere for jer at styre ting i klubben (medlemmer, kontingenter, arrangementer, fartøjer osv.)?
- Kan medlemmer leje/låne jeres fartøjer (joller, sejlbåde m.m.) når de ikke benyttes af sejskolen?
 - Hvis nej, hvorfor tilbydes denne funktion ikke, er det noget der kunne tilbydes hvis der var et system til at hjælpe med administration?
 - Hvis ja har vi et par ekstra spørgsmål herom:
 - * Hvad er kravene for at låne fartøjet?
 - * Hvordan håndteres omkostninger ved udlån af fartøjet?
 - * Hvordan administreres udlånte fartøjer og kunne hertil benyttes et system til hjælp? I så fald i hvordan kunne et system bedst muligt være til hjælp?

UML-diagram F



Figur F.1. UML-diagram over klasserne i projektets program

Use Cases

Case 1:

En bruger af systemet ønsker at komme ud og sejle. Brugeren logger ind på systemet med sit personlige login. Han navigerer ind på bookingsiden. Han vælger sin foretrukne båd og vælger datoen for hans booking. Han gennemfører bookingen og logger derefter ud af systemet.

Case 2:

En bruger som har været ude og sejle logger ind på systemet. Han ser på sin forside at han mangler at udfylde logbogen for den sejltur han lige har været på. Han vælger logbogen og trykker udfyld. Vinduet hvor han skal udfylde de resterende informationer popper op, og han udfylder felterne. Han trykker derefter på gem log, og logger ud af systemet.

Case 3:

En administrator har fået at vide at en båd er blevet skadet under en sejlads. Han logger ind på systemet og navigere ind på den pågældende båds side. Han åbner logbogen for bådens seneste tur og ser at båden er skadet. Han skriver en kommentar til logbogen, hvori han skriver at skaden vil blive udbedret indenfor kort tid. Administratoren logger derefter ud igen.

Case 4:

Sejlkлубben ønsker at afholde en grilaften med tilhørende sejlads. En administrator skaber en begivenhed, via interfacet, og skriver en passende titel, beskrivelse og tidspunkt. Han bruger båd interfacet til at book et passende antal både og tilføjer sig selv som kaptajn. Folk tilmelder sig begivenheden, via interfacet, og møder op til arrangementet. Når sejlturen er ovre udfylder administratoren logbøger, her vil han udfylde med det korrekt data for kaptajn og besætning.¹

FiXme Fatal!

¹FiXme Fatal: Måske skal der være et vindue til at ændre bookings

Bruger tests af program



Du skal nu teste programmet vi har udviklet i forbindelse med vores P2-projekt på AAU. Programmet er lavet som et management system til en sejlkлуб, så de kan holde styr på deres både, deres medlemmer, deres undervisningsforløb, og desuden deres logbøger de holder for alle klubbens både. I den forbindelse vil du blive bedt om at udføre forskellige opgaver i vores program. Et eksempel kunne være at vi beder dig logge ind med et givent brugernavn og kodeord, og herefter finde frem til medlemmet Troels Kroegh, i medlemsoversigten.

Hvis der er noget du kommer i tvivl om, f.eks. hvad du skal i den givne opgave, så kan du spørge, og vi vil vurdere om det vil ødelægge testen at få svaret på dit spørgsmål.

H.1 Opgave 1

Log ind med følgende brugeroplysninger:

- Brugernavn: kasper
- Kodeord: eriksen

Når du har gjort dette bedes du finde tilmelde dig begivenheden "Pizza Fest" som finder sted den 20/6-14 klokken 19.00.

H.2 Opgave 2

Forbliv logget ind til denne opgave. Du bedes nu booke en båd med følgende oplysninger:

- Bådnavn: Anastasia
- Besætning:
 - Lars Olsen
 - Karen Wolff
 - Bodil Kjær
 - Anne Frank
- Kaptajn: Anders And
- Formål: "Vi skal ud og fiske på havet."
- Afgang: "20/6-14 klokken 09:00 "
- Ankomst "20/6-14 klokken 17.00 "

Log herefter ud af programmet, så du er klar til næste opgave, som du finder på næste side.

H.3 Opgave 3

Log ind med følgende brugeroplysninger:

- Brugernavn: oskar
- Kodeord: lauridsen

Dette medlem er administrator da medlemmet er en af sejlernes undervisere. Du har lige været ude og undervise eleverne og skal nu registrere at dit hold har fuldført et nyt punkt på deres liste over mål i forbindelse med uddannelsen.

Du bedes

FiXme Note!

- Finde holdet: Svenskerne
- Vælg lektion: 05/09/2014 kl. 15:00 ¹
- Afkryds at alle medlemmer på holdet har fuldført deres natsejls
- Gem herefter lektionsinformationerne

H.4 Opgave 4

Forbliv logget ind til denne opgave.

Gør følgende:

- Vælg hold: Svenskerne
- Opret lektion for holdet der starter kl. 19.00 og slutter kl. 21.00 den 1. august 2014

H.5 Opgave 5

Forbliv logget ind til denne opgave.

Her skal du:

- Opret et nyt hold
- Navngiv holdet: TestHold
- Holdet skal bestå af følgende medlemmer:
 - Isabella Christensen
 - Malthe Frandsen
 - Malene Jensen
 - Josefine Henriksen
- Holdet skal være et andenårs hold
- Gem nu holdet i programmet

H.6 Opgave 6

Forbliv logget ind til denne opgave.

FiXme Note! Holdet 'MasterRace'² har færdiggjort alle læringsområderne, så de kan nu få et bådførerbevis.

For at opnå dette, gør følgende

- Vælg hold: MasterRace
- Forfrem holdet

H.7 Opgave 7

Forbliv logget ind til denne opgave.

Du bedes:

- Vælg hold: TestHold
- Slet holdet

H.8 Opgave 8

Inden du logger ud fra administratorbrugeren bedes du oprette en ny begivenhed.

Begivenheden skal foregå den 7/6-14, og skal hedde, Bålfest ved molen. Beskrivelsen kan være hvad som helst, alle kan tilmelde sig. Log herefter ud igen.

H.9 Opgave 9

Log ind med følgende brugeroplysninger:

- Brugernavn: michelle
- Kodeord: kristensen

Dette medlem en elev. Du er færdig med de to års sejlskole og vil være sikker på at du kan få dit bådførerbevis.

Tjek derfor at alle læringsområderne er tjekket af.

Log herefter ud af programmet.

H.10 Opgave 10

Log ind med følgende brugeroplysninger:

- Brugernavn: marcus
- Kodeord: husmand

Du er nu kommet hjem fra en sejltur og skal udfylde din logbog for turen.

Turen foregik den 18/6-14, og havde formålet: "Sejler til Sverige og hjem igen."

¹FiXme Note: Kunne ikke åbne programmet da jeg lavede det her så er ikke sikker på formatet

²FiXme Note: Alle eleverne på holdet skal have sat alle læringsområderne til true

Find logbogen frem og udfyld felterne, efter egen fantasi. (Du får ikke oplyst informationerne her, du skal altså selv finde på noget at skrive. Det behøver ikke give mening.)

Når du har gjort dette vil vi bede dig finde frem til din nu udfyldte logbog.

H.11 Testen er nu slut

Tak fordi du ville være med i vores test, vi har nogle opfølgende spørgsmål som vil blive stillet til dig af observatøren fra vores gruppe.

3

³FiXme Note: Her skal være bedre beskrivelse af både opret event samt undervisning, jeg kender ikke nok til funktionerne i har lavet.

Spørgeskema for brugertests af programmet.

På en skala fra 1-5 hvor overskueligt synes du så programmet var at navigere rundt i?

[1 er meget uoverskueligt, og 5 er meget overskueligt.]

- Vurdering:
- Hvorfor denne holdning?
- Hvilke elementer tænker du specifikt på?

På en skala fra 1-5, hvor let var det at overskue et skærmvindues funktionaliteter? - Se tabellen nederst på siden.

[1 er meget svært, og 5 er meget let.]

Var der noget ved programmet du ikke forstod?

Var der noget der fungerede godt?

Hvad fungerede mindre godt?

Hvis du kunne ændre noget på programmets udseende eller navigering, hvad ville det så være?

Har du nogle forslag eller tanker om programmet?

	Vurdering	Kommentar
Forside		
Undervisning		
Medlemmer		
Både og Booking		
Begivenheder		
Logbogsvinduet		
CreateCrewWindow		
CreateBoatBookingWindow		