

DAT17A

# DELFINEN

7.12.2017

MARTINE GARDE MAGH  
03-05-1993

ANE MARIE PETERSSON  
21-04-1996



VERONIQUE CACHO BASSON JENSEN  
17-09-1992



MAHAM ALI  
05-01-1998

## Indledning.

Svømmeklubben Delfinen er den opgave, vi har fået tildelt det første semester, opgaven er delt op i tre dele, Systemdesign, system konstruktion og IT Organisations delen. Ingen af os har haft et stort kendskab til nogen af disse, før vi startede på KEA-Datamatikeruddannelsen. Vi har derfor valgt at arbejde sammen som en gruppe om opgaven i stedet for at dele den op og lave for meget individuelt, vi vil gerne fokusere på at samarbejde og kommunikere med hinanden, så alle gruppens medlemmer har noget at skulle have sagt og forstår det produkt vi laver. Det er derfor ikke defineret i opgaven hvem der helt præcist har lavet hvilke dele. Vi har tildelt lektier til hinanden, som vi så har kigget på sammen og foretaget rettelser på i fællesskab, hvis det skulle være nødvendigt, men vores primære vision for opgaven er at udarbejde den i fællesskab.

- Vi har kaldt vores system for Delfinen2017
- Projektet er udarbejdet af Maham, Ane, Martine og Veronique.
- Projektet afleveres d. 7/12-2017

# Indholdsfortegnelse

Glossary .....	4
Kravliste .....	6
Use Case diagram: Svømmeklubben Delfinen .....	7
Use Case 1: tilføjMedlem .....	8
Use Case 5: seRestance .....	10
Use Case 7: tilføjResultater .....	12
Brief Use Cases .....	14
Domænenmodel.....	15
Sekvens diagram.....	16
SWOT - Analysis (ITO).....	17
Interessentanalyse (ITO) .....	17
MATRIX .....	18
Interessenterne .....	19
Konklusion .....	20
Kode.....	20

## Glossary

Medlemsoplysninger	Et medlems navn, alder, medlemstype, hold, svømmedisciplin, svømmeresultater. Så alle medlemmets informationer.
Redigere	Tilføje og fjerne informationer.
Prisklasser	Kontingent i forhold til medlems typer.
Medlemstype	<ol style="list-style-type: none"><li>1. Under 18 år = 1000 kr. årligt.</li><li>2. Over 18 år = 1600 kr. årligt.</li><li>3. Over 60 år får 25% rabat = 1200 kr. årligt</li><li>4. Passive medlemmer = 500 kr. årligt.</li></ol>
Betalingsinformationer	Kortnummer, kvitterings adresse, navn, adresse, telefonnummer, mail adresse.
Top 5	5 svømmere inde for hver svømmedisciplin, med de bedste resultater.
Responstid	Responstid i forbindelse med computerteknologi, er den forløbne tid mellem en forespørgsel på et system, og svaret på denne forespørgsel.
Oppetid	Den tid et computersystem er i drift, f.eks. skal systemet have en oppetid på 99.9%.
Håndtere	Behandle noget.
Svømmeart	De forskellige former for svømmediscipliner konkurrence svømmere skal udøve, indeholder butterfly, crawl, brystsvømning og hundesvømning,
Placering	Det sted man som konkurrencesvømmer rangerer under konkurrencen.
Beløb	Den mængde penge et individ skal betale for et medlemskab, varierer for forskellige medlemskaber.
Distance	Den længde som man som konkurrencesvømmer skal svømme, varierer fra disciplin til disciplin.
Medlemsinfo	Indeholder: medlems nummer, fornavn, efternavn, fødselsdato, adresse og medlemstype.
LoginMenu	<ul style="list-style-type: none"><li>• Formand Login</li><li>• Kassér Login</li><li>• Træner Login</li></ul>

Restanceliste	En liste med alle de medlemmer, der ikke har betalt deres kontingent.
Send girokort	Send et girokort med de kontingenter de ikke har betalt.
Redigerer medlem	Ændre i deres medlemsinfo.

## Kravliste

### Medlemsoplysninger:

1. (F) Systemet skal vise medlemsoplysninger
2. (F) Systemet skal kunne registrere medlemmer (opretMedlem)
3. (F) Systemet skal kunne redigere eksisterende medlemmer (redigerMedlem)
4. (F) Systemet skal kunne registrere medlemmer som aktive eller passiv (sætAktive / sætPassiv)
5. (F) Systemet skal kunne slette informationer om enkelte svømmere (sletMedlem)

### Kontingenter:

6. (F) Kasséren skal kunne styre betaling af kontingent
7. (F) Kasséren skal kunne se en oversigt af medlemmer
8. (F) Systemet skal vise de forskellige prisklasser delt op efter alder og medlemskaber (under 18, over 18, over 60, passivt)
9. (F) Systemet skal kunne redigere i betalingsinformationer (registrereBetaling)
10. (F) Systemet skal vise aktive eller passive medlemmer
11. (F) Systemet skal kunne vise en liste over restance medlemmer (seRestance)

### Svømmeresultater:

12. (F) Systemet skal kunne gemme informationer om medlemmets svømmers resultater (registreStævneResultat)
13. (F) Systemet skal vise medlemmernes svømme resultater fra konkurrencer, stævner og svømmediscipliner
14. (F) Systemet skal vise en oversigt over top 5 svømmere fra hver svømmedisciplin (seTopFem)
15. (F) Systemet skal kunne vise konkurrence medlemmernes svømmeresultater, svømmediscipliner, og hvilket hold de går på (ungdomshold eller seniorhold)

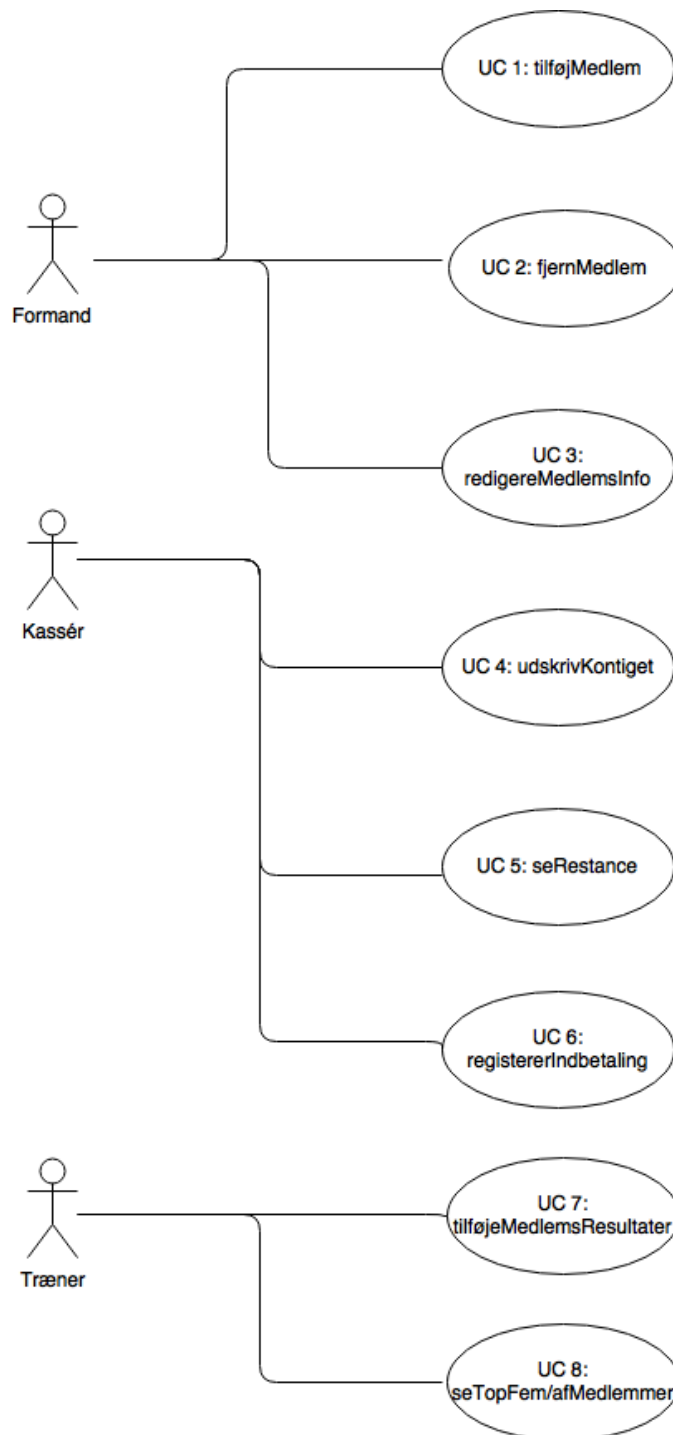
### Andet:

16. (I) Systemet skal laves i Java
17. (I) Systemet skal være oppe at køre d. 7. december 2017
18. (F) Systemet skal have en responstid på max 2 sekunder pr. side
19. (I) Der skal være dokumentation som beskriver hvordan systemet er bygget
20. (F) Systemet skal have en opetid på mindst 99,9%

### FURPS+

<u>Functionality:</u>	<u>Usability:</u>	<u>Reliability:</u>	<u>Performance:</u>	<u>Supportability:</u>
Krav 1-15 Krav 18 Krav 20	Krav 19	Krav 18	Krav 11 Krav 18 Krav 20	

## Use Case diagram: Svømmeklubben Delfinen



Vi har lavet 1 stor use cases fra 3 forskellige aktører vinkler.  
Det er blevet sat op efter rækkefølge.  
Formanden håndterer medlemmerne.  
Kassereren håndterer kontingent og restance.  
Træneren håndterer konkurrence resultater.

## Use Case 1: tilføjMedlem

### Scope:

Medlemssystem

### Level:

User goal

### Primary actor:

Formand

### Stakeholders and interests

- Formanden: Have oversigt over alle medlemmer
- Træneren: Skal kunne se hvilke medlemmer er på hvilke hold
- Kassereren: At se de nye medlemmer og håndtere deres betalinger
- Medlem: Vil kunne melde sig på hold og eventuelt deltag i stævner
- Skat: At få MOMS og firmaSKAT

### Preconditions

Betalingsmetoder

Medlemmet skal vælge om de ønsker motionist eller konkurrence medlemskab

### Success Guarantee

Kunden kan få et medlemskab

Som medlem kan kunden få lov til at vælge hvad han ønsker at deltage i

Medlemmet betaler kontingentet

### Main success scenario

1. Kunden beslutter hvilket medlemskab han ønsker
2. Kunden informerer formanden om hvilket medlemskab han vil have
3. Formanden taster kundens rigtige oplysninger og medlemstype ind
4. Kunden giver betalingsoplysninger
5. Kunden betaler kontingent
6. Kunden bliver oprettet som medlem
7. Kunden kan officielt gøre brug af delfinens faciliteter

### Extensions

1. Kunden vælger forkert medlemskab
2. Kunden informerer formanden om det forkerte medlemskab
3. Formanden taster de forkerte oplysninger ind
4. Kunden giver forkerte betalingsoplysninger
5. Kunden kan ikke betale kontingentet
6. Kunden bliver ikke oprettet i systemet
8. Der er ikke flere pladser på kundens ønskede medlemskab
9. Medlemskabet stemmer ikke overens med kundens forventninger



### Special requirements

Systemet skal kunne gemme oplysninger

Systemet skal kunne tilføje og fjerne medlemmer

### Technology and data variations list

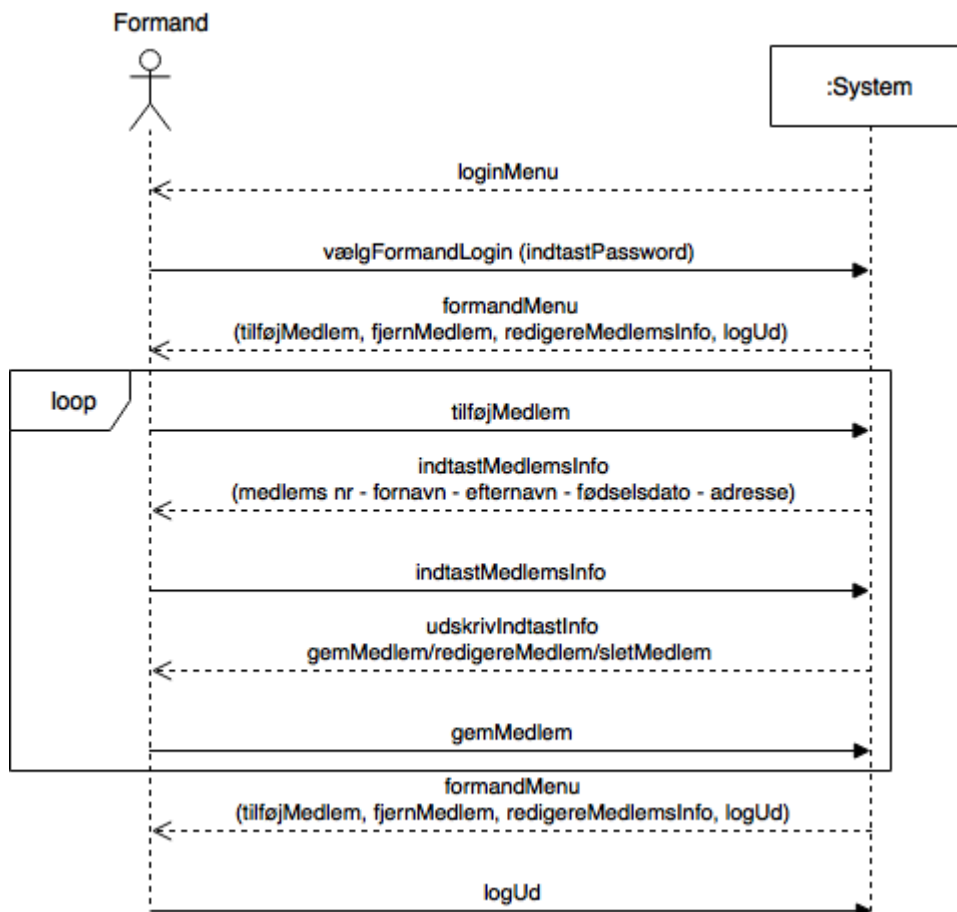
Det skal have en opetid på 99.9%

Systemet skal laves i Java

### Frequency of occurrence

Der forventes at komme ca 5 nye medlemmer om måneden

### System sekvens diagram:



## Use Case 5: seRestance

### Scope:

Medlemssystem

### Level

User goal

### Primary actor

Kasserer

### Stakeholders and interests

- Kasserer: Skal kunne se hvilke medlemmer er i Restance og derfra kunne lukke deres medlemskab for manglende betaling
- Medlem: Skal gøres opmærksom på at de ikke har betalt og informeres hvis der eventuelt er et problem med deres betalingskort
- Formand: Skal være tryk i at alle medlemmer betaler deres kontingent

### Preconditions

Medlemmer der ikke betaler kontingent og kommer i restance kan miste deres medlemskab

### Success Guarantee

Kassereren kan se hvilke kunder der har og hvilke der ikke har betalt

Kassereren kan kontakte medlemmer der er i restance

### Main success scenario

1. Kassereren logger ind i sit system
2. Kassereren vælger se restanceliste
3. Kassereren kan se en liste over de medlemmer der er i restance
4. Kassereren sender en rykker ud med en advarsel til dem der mangler at betale.
5. Hvis et medlem betaler sin restance fjerner hun dem fra listen.

### Extensions

1. Kassereren kan ikke logge ind
2. Kassereren kan ikke vælge "se restancelisten"
3. Kassereren kan ikke se listen over de medlemmer der er i restance
4. Kassereren kommer til at sende en rykker, til den forkerte person
5. Kassereren glemmer at fjerne medlem fra restancelisten
6. Kassereren glemmer at gemme opdateringer
7. Systemet er gået ned eller er ikke opdateret

### Special requirements

Systemet skal kunne gemme oplysninger

Systemet skal kunne tilføje og fjerne medlemmer

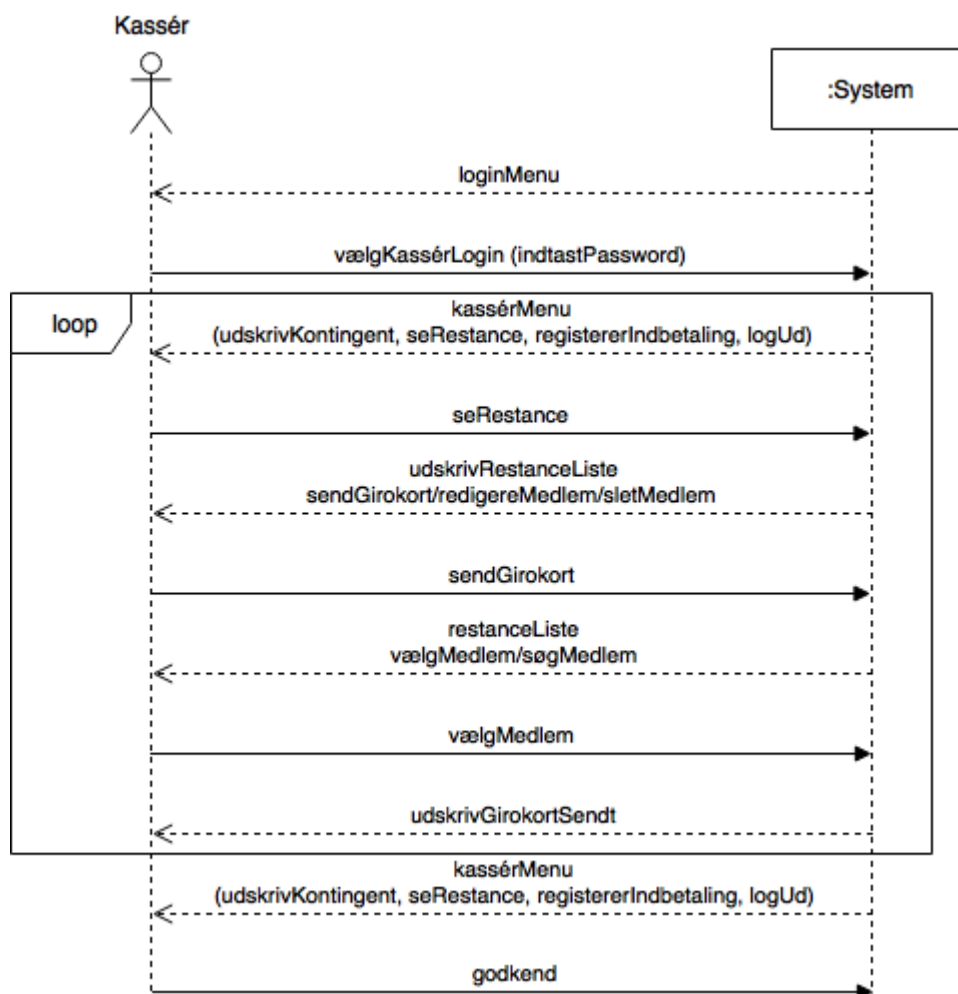
### Technology and data variations list

- Det skal have en oppetid på 99.9%
- Systemet skal laves i Java

### Frequency of occurrence

Kassereren skal kunne se medlemmer hele tiden, men i hvert fald 1-2 gange om måneden skal hun kunne se hvem der er i restance.

### System sekvens diagram:



## Use Case 7: tilføjResultater

### Scope:

Medlemssystem

### Level:

User goal

### Primary actor:

Træner

### Stakeholders and interests

- Træner: skal kunne skrive de forskellige konkurrencemedlemmers resultater ind, og derved kunne se hvem er top 5
- Medlem: Skal kunne se hvor højt de rangerede på listen af deltagere
- Formanden: Ønsker at hans medlemmer rangerer så højt som muligt for at klubben skal være konkurrencedygtig.

### Preconditions

Systemet skal være oppe og køre

Træneren skal have resultaterne nedskrevet på forhånd

### Success Guarantee

Træneren skriver de forskellige medlemmers resultater ind så både formanden og svømmerne kan se hvor de ligger henne

### Main success scenario

1. Træneren logger ind
2. Træneren finder "Indskriv resultater"
3. Træneren skriver de rigtige resultater ind
4. Træneren gemmer resultaterne
5. Systemet fremviser top 5 inden for de forskellige kategorier
6. Træneren offentliggøre resultaterne så klubben og formanden kan tage dem til overvejelse

### Extensions

1. Træneren kan ikke logge ind
2. Træneren kan ikke trykke på "Indskriv resultater"
3. Træneren skriver de forkerte resultater ind
4. Resultaterne bliver ikke gemt korrekt
5. Systemet går ned under sessionen
6. Træneren glemmer at offentliggøre resultaterne

### Special requirements

Systemet skal gemme oplysningerne.

Systemet skal gøre det muligt at se top 5 af alle svømmere baseret på resultaterne.

### Technology and data variations list

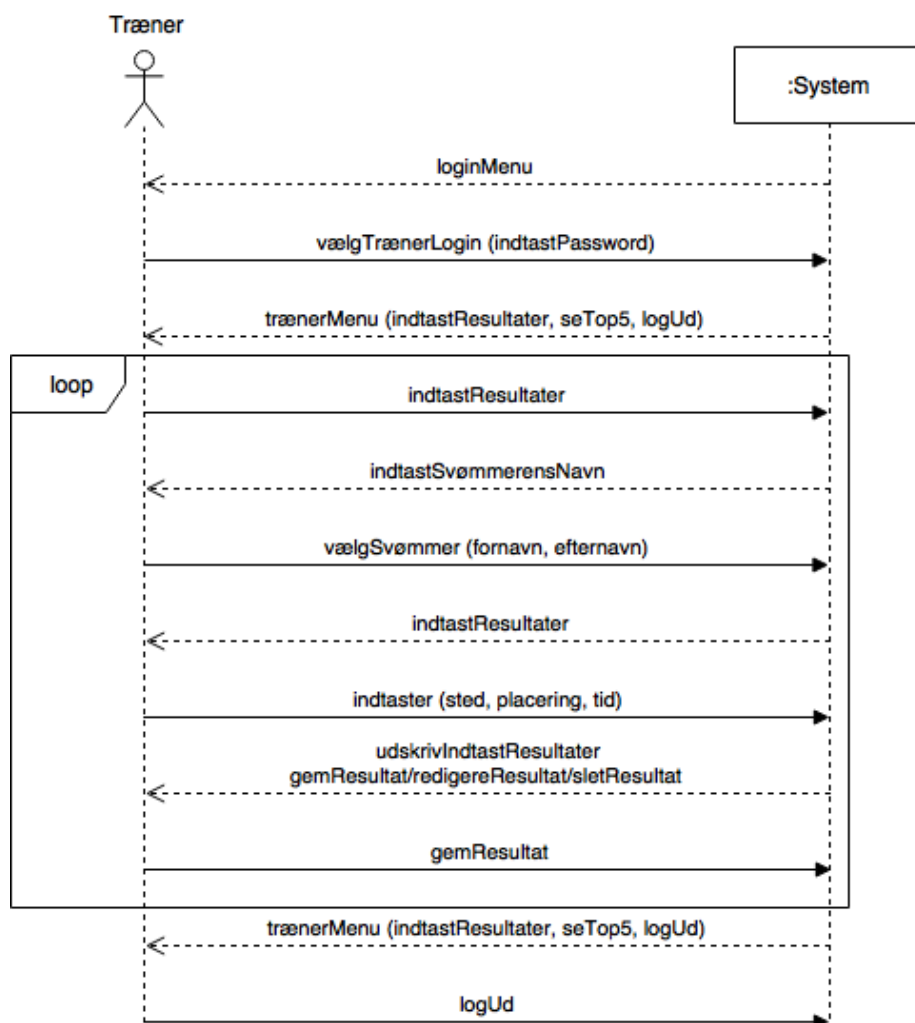
Det skal have en oppetid på 99.9%.

Systemet skal laves i Java.

### Frequency of occurrence

Der forventes at være svømmestævner 1 gang om måneden

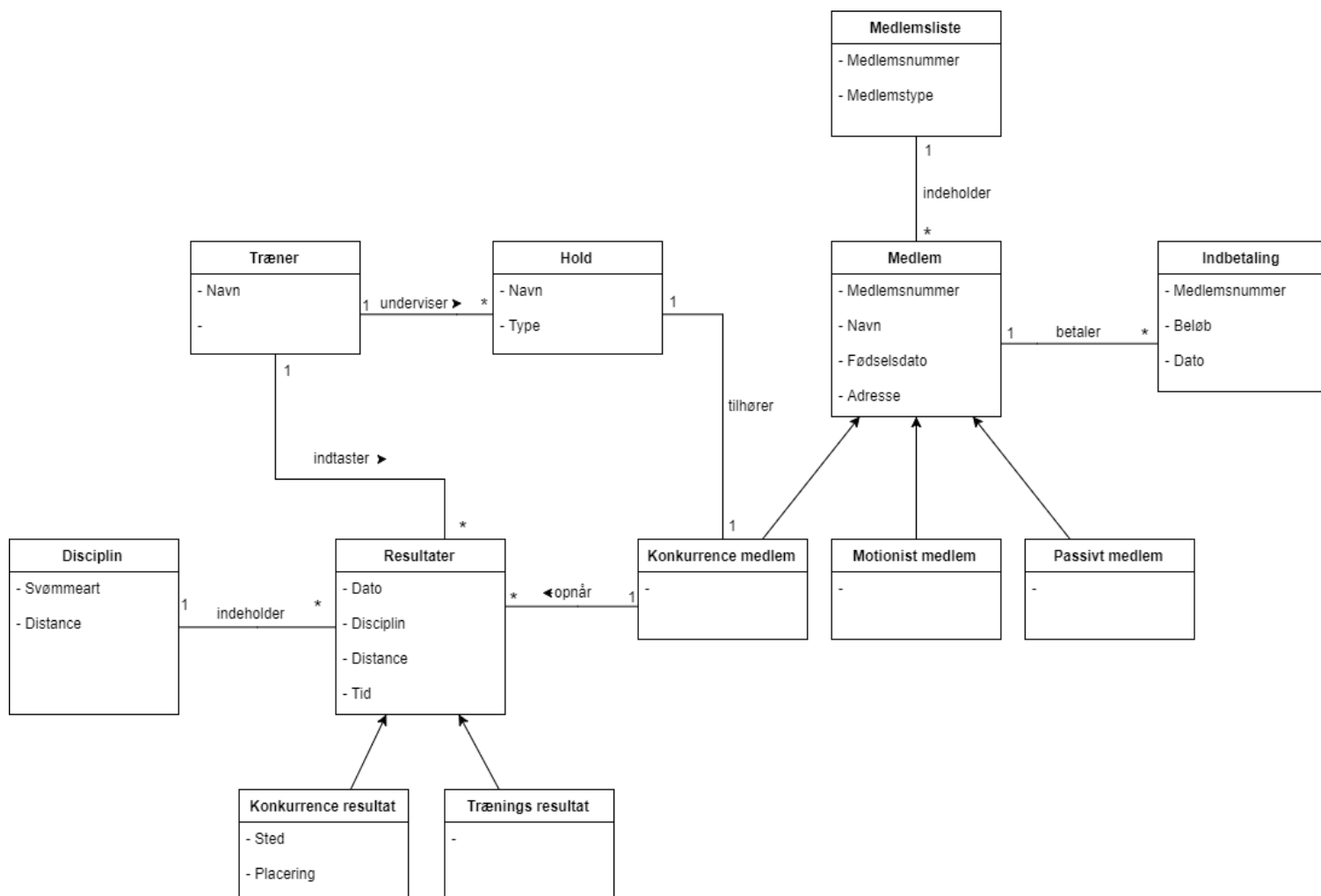
### System sekvens diagram:



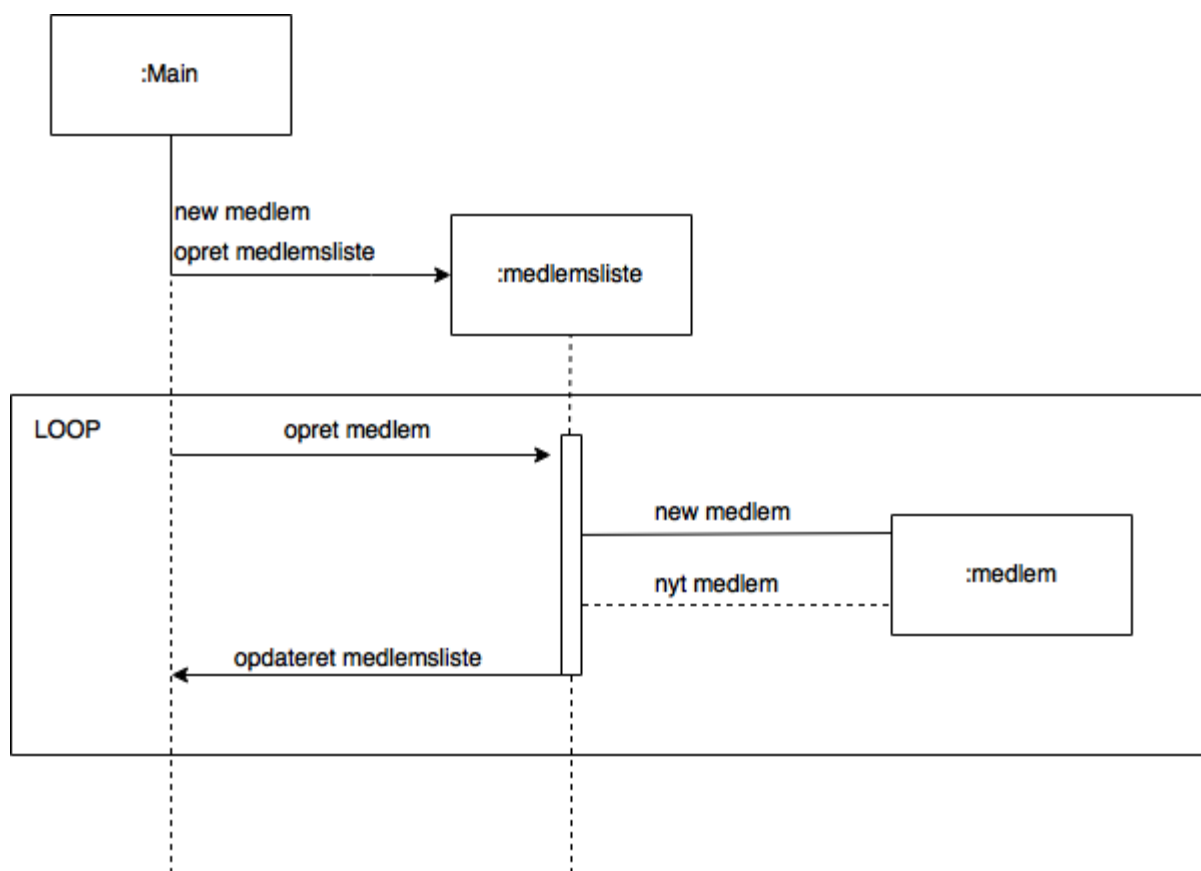
## Brief Use Cases

Actor	Goal	Brief
UC 3 - Formanden	Registrere medlemsinfo	Denne Use Case beskriver hvordan aktøren (formanden) bruger systemet til at redigere medlemsoplysningerne. Formanden går ind i systemet og ændrer de gamle oplysninger (adresse, telefonnummer, medlemstype og holdtype) for et medlem, til dets nye og korrekte informationer. Hvorefter, systemet opdaterer medlemsprofilen og informere aktøren om, at oplysningerne er opdateret/gemt med succes.
UC 6 - Kassereren	Registrere Indbetaling	Use casen her beskriver hvordan aktøren (kassereren) bruger systemet til at registrere medlemmernes indbetalte beløb. Kassereren sørger for at medlemmerne har betalt alt (hvis nu nogen er i restance), ellers påminder aktøren de enkelte medlemmer om betalingen ved indsendelse af en rykker/advarsel.
UC 8 - Træneren	Se top 5 medlemmer	I dette Use Case diagram beskrives der hvordan aktøren (træneren) bruger svømmeklubbens administrativt system til, at få fremvist konkurrenceresultaterne for klubbens medlemmer. Ved hjælp af systemet kan aktøren nemt komme ind og se top 5, altså de 5 bedste svømmere inde for hvert hold og dets svømme discipliner.
UC 2 Formanden	Fjern medlem	Denne Use case er relevant når formanden der er den primære aktør i dette scenario skal fjerne et medlem der enten er blevet smidt ud eller har opsagt sit medlemskab. Han skal kunne logge ind på systemet og finde det specifikke medlem, hvorefter han skal kunne fjerne dem permanent fra systemet så de hverken fremgår eller skal betale for medlemskab længere. Derefter skal systemet oplyse ham om at medlemmet er fjernet og oplysningerne gemt.
UC 4 Kassereren	Udskriv kontingent	I dette use case er kassereren den primære aktør, kassereren skal her kunne logge ind på sit system og se alle medlemmerne, hun skal så derefter kunne se hvor meget hvert enkelt medlem skal betale og hvornår betalingen falder, derefter skal hun kunne udskrive dette og sende det videre til hvert enkelt medlem i god tid, så de har mulighed for at betale.

## Domænemodel



## Sekvens diagram





## SWOT - Analysis

Interne situation	
Strengths	Weaknesses
God kommunikation. Godt kontingent og restance overblik. God svømme resultat overblik.	Konkurrencedygtige modstandere. Usikkerhed i kode delen. For mange medlemmer at håndtere. Svagt system.
Eksterne situation	
Opportunities	Threats
Udvidelse af svømmehallen (fx. flere forskellige medlemstyper). Oprettelse af en internetside med information. Medlems login Salg af systemet til andre klubber.	Andre og større svømmeklubber med bedre system. Klubben vælger andre programmører til at udarbejde systemet.

## Interessentanalyse (fra svømmeklubben- og system udviklernes side)

### Hvem er ejerne af Svømmeklubben Delfinen? Hvem har en indflydelse i svømmeklubben?

Ledelsen, Formanden, Kassereren, Træneren, Medlemmer

### Hvilken rolle har formanden i klubben? (Udover at sidde i ledelsen)

Formanden indmelder og udmelder medlemmer, og redigere deres oplysninger.

### Hvem betaler til klubben?

Det gør passive og aktive medlemmerne.

De forskellige kontingenttyper:

1. Under 18 år = 1000 kr. årligt.
2. Over 18 år = 1600 kr. årligt.
3. Over 60 år får 25% rabat = 1200 kr årligt
4. Passive medlemmer = 500 kr. årligt.

### Hvad skal systemudviklerne være opmærksomme på mht. priserne?

- Om medlemmerne er over eller under 18 år.
- Om medlemmerne er over 60 år.

### Hvem kan blive en trussel for klubbens tilstedeværelse?

Konkurrenter til klubben, altså andre svømmeklubber i nærområdet eller konkurrencesvømmere der konkurrerer imod Delfinerne.

### Er der mulighed for, at skabe en karriere i svømmeklubben?

Man kan gå til konkurrencesvømning, med en tilknyttet træner dertil. Der er både for dem under 18 (Junior) og dem over 18 (Senior).

## MATRIX

### Matrix: Svømmeklub Delfinen

	Ikke indflydelse i klubben	Indflydelse i klubben
<b>Interesse i klubben</b>	Andre svømmeklubber Andre konkurrencesvømmere Nye svømmeklubber	Aktive medlemmer Passive medlemmer Konkurrencesvømmere Klubbens ansatte (Formanden, kassér, træner, osv...) Sponsorer
<b>Ikke interesse i klubben</b>		Mindre klub som er i vækst Rengøring af klubben Pedellen

### Matrix: Systemudviklernes

	Ikke indflydelse i klubben/systemet	Indflydelse i klubben/systemet
<b>Interesse i klubben/systemet</b>	Andre systemudviklere	Os som systemudviklere Ledelsen (et hurtigt system)
<b>Ikke interesse i klubben/systemet</b>	Det gamle styresystem	Skift af gamle styresystem

## Interessenterne

### Hvad ønsker formanden at se fremadrettet?

- En øget vækst af medlemmer.
- Flere sponsorer.
- Gode resultater fra konkurrencesvømmerne.
- Flere penge i kassen.
- At være nummer 1. (mod de andre svømmeklubber).

### Hvad ønsker formanden af det nye styresystem?

- En nemmere måde registrerer og redigere medlemmer.
- Systemet er optimalt.
- Systemudviklerne overholder deadline.
- Et system der holder i lang tid
- Et system der kan videreudvikles.

### Vil der kunne opstå ulemper som formanden ikke ønsker at se fremadrettet?

- En faldende vækst af medlemmer.
- Dårlige resultater fra klubbens konkurrencesvømmerne.
- Nye svømmeklubber åbner i nærheden.
- Systemet er dårligt.
- Faldende omsætning.

### Hvad ønsker udviklerne af ledelsen?

- De har en klar idé om hvad systemet skal kunne.
- Et realistisk ønske mht. systemet
- En fast deadline.

## Konklusion

Efter at have udarbejdet opgaven kan vi konkludere at vores vision for opgaven stort set stemte overens med det endelige resultat med et par forskelle. Vi sad meget sammen og lavede opgaven, men vi opdagede at folk i gruppen havde forskellige styrker og svagheder i det faglige stof. Nogle var bedre til at arbejde med design delen af opgaven mens andre bedre forstod programmeringsdelen, det var dog generelt programmeringen vi alle fandt sværest. Vi startede primært med at lave design delen, hvor vi så langsomt gik i gang med ITO og software konstruktion efter omkring en uge. Vi brugte desværre for lang tid på designdelen af opgaven, at vi undervurderede hvor svært vi ville finde programmeringen, hvilket desværre resulterede i, at vi ikke nåede hele softwarekonstruktionen, som vi ellers ønskede.

## Kode

<https://github.com/Veronique0165/Delfinen.git>