

# Rapport: Delfinen Svømmeklub

<b>Indledning</b>	<b>1</b>
<b>Gannt chart</b>	<b>2</b>
<b>Interessent analyse</b>	<b>3</b>
<b>SWOT Analyse</b>	<b>4</b>
<b>Risiko analyse</b>	<b>5</b>
<b>Use Case Diagrammet</b>	<b>6</b>
<b>Use Cases</b>	<b>7</b>
Use Case: Create new member	7
Use Case: Subscription payment	7
Use Case: Arrears overview	8
Use Case: Resultatregistrering	8
Use Case: Result chart	9
<b>System Sequence Diagrams</b>	<b>10</b>
Subscription payment SSD	10
Create New Member SSD	11
Check Arrears SSD	12
Result Registration SSD	13
Results Chart SSD	13
<b>Domain model</b>	<b>14</b>
<b>Klassediagram</b>	<b>15</b>

## Indledning

Svømmeklubben delfinen ønsker et system, der kan håndtere klubbens medlemsinformationer, kontingenter og resultater.

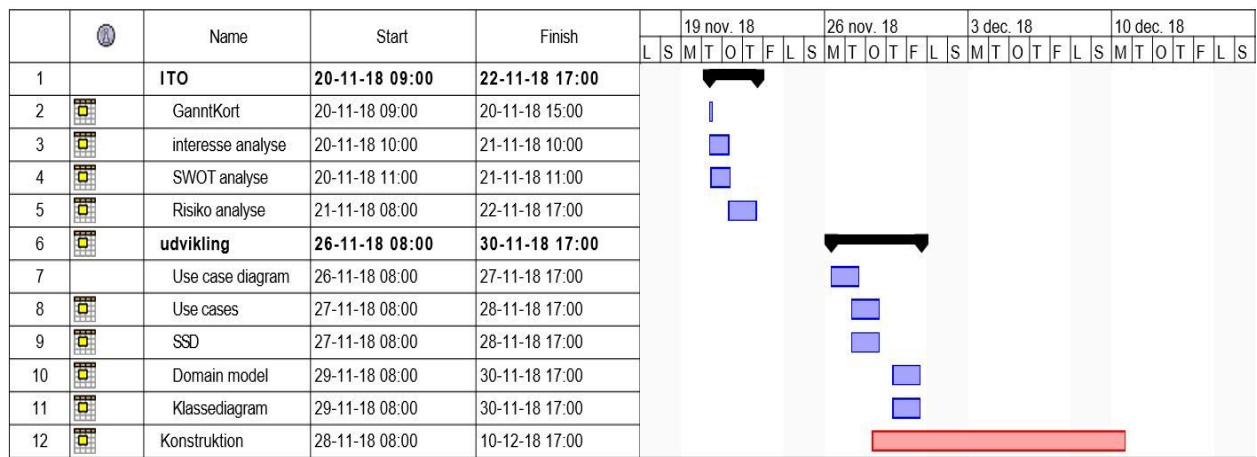
Systemet skal kunne:

- Oprette nye medlemmer
- Give et overblik over restancer
- Tage imod betaling fra medlemmer
- Tage imod resultater fra en træner
- Oplyse data fra konkurrencesvømmere, så trænerne kan udtage hold til konkurrencer

Følgende rapport beskriver systemet, som er blevet udviklet for at opfylde de krav, der er stillet af Svømmeklubben Delfinen. På grund af tidsbegrænsningen er der kun udviklet en prototype af det færdige program, og opgaven beskriver derfor systemet, som det ville være bygget, hvis vi havde arbejdet videre på det. Rapportens opbygning følger samme system, som er sat op i gannt chartet nedenfor. Hvert afsnit indeholder et eller flere diagrammer, der, sammen med en tilhørende tekst, søger at belyse projektets udvikling og egenskaber.

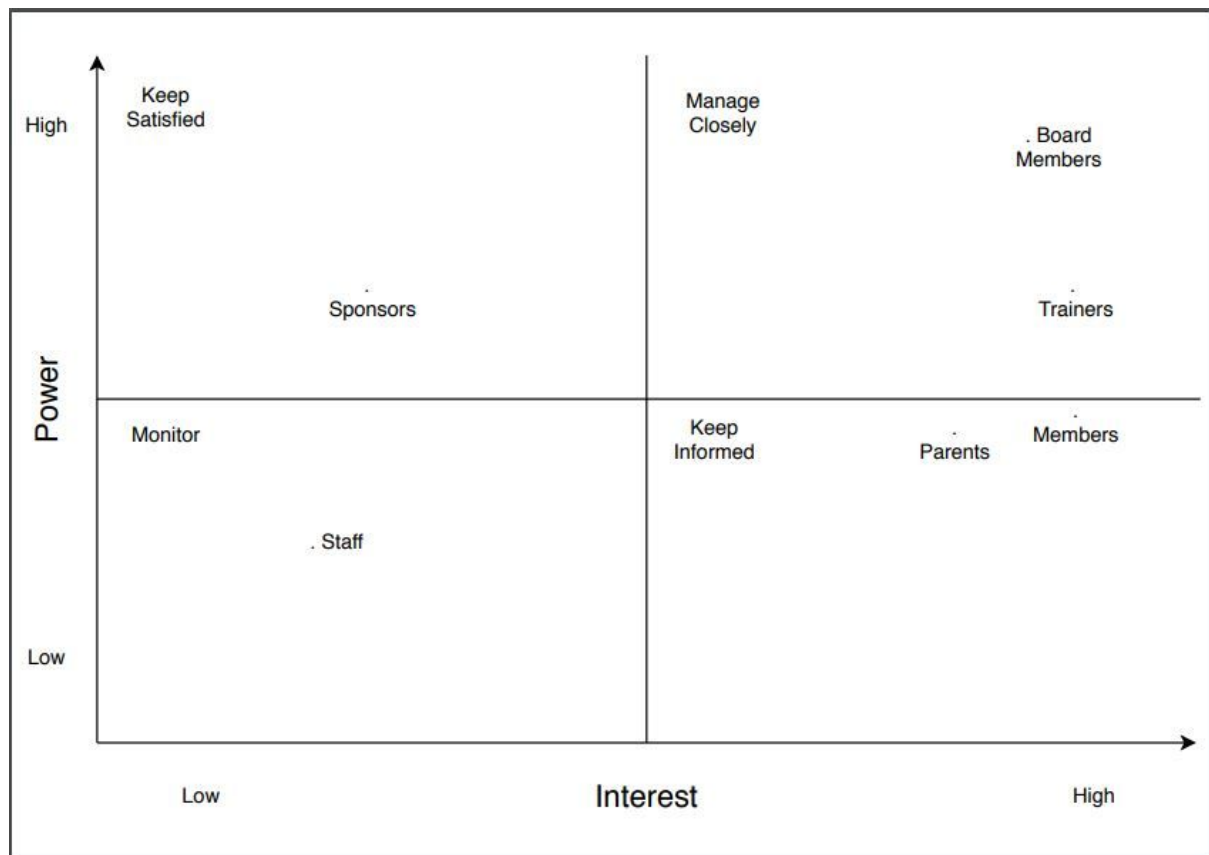
## Gannt chart

Her under ses et Gannt chart, som beskriver projektets planlægning og forløb.



I chartet er der tre hovedpunkter; ITO, udvikling og konstruktion, som er de overordnede områder, hvorigennem projektet er blevet opbygget. Fordelingen af hovedområder er blevet fulgt forholdsvis systematisk for at kunne holde styr på den løbende udvikling. Gannt chartet beskriver kun det planlagte arbejde, der er blevet gjort i henhold til tidsbestemmelsen. Vi har derudover foretaget løbende ændringer i tidligere afsluttede punkter, så f.eks 'use cases' har passeret til selve programmet. Det planlagte arbejde er sat til at slutte d. 10/12, så der en buffer før afleveringsdeadline.

## Interessent analyse



Interessent analysen viser hvilke aktører, der har en relation til klubben. Aktørerne er inddelt i fire grupper, efter hvor stor interesse de har i klubben, og hvor stor indflydelse de har. Inddelingen grupper giver et overblik over, hvordan man skal forholde sig til aktørerne hvis de er kritisk instillede overfor driften, eller hvis man gerne vil holde dem positive. I vores analyse ses det f.eks, hvordan sponsorerne interessante for klubben. De er en stor intægtskilde, som er nødvendig for klubbens drift, og hvis man ikke holder dem tilfredse, så kan det have økonomiske konsekvenser for klubben.

# SWOT Analyse

SWOT analysen beskriver Svømmeklubben Delfinens interne og eksterne forhold.



SWOT analysen er blevet udarbejdet ud fra fiktive fakta, da der ikke er blevet givet nogle konkrete bilag til hvordan svømmeklubben er som virksomhed, eller hvordan dens omverden ser ud. Vi har derfor selv fundet på de styrker og svagheder som vi tror en svømmeklub har i virkeligheden. Internt er der f.eks. er en generel efterspørgsel på svømmehaller, da der ikke findes en i alle byer. I de eksterne forhold har vi igen antaget hvordan det ser ud i virkeligheden. Svømmeklubben vil have en mulighed i at få flere voksne ind i svømmeklubber, da det hovedsageligt er børn/ unge og ældre, der bruger svømmehaller. Modsat mulighederne ser vi det som en stor trussel, at fitnesscentre bliver større og større, da de også er begyndt at bygge egne svømmebaner i centrene, eller på anden vis tiltrækker samme kundekreds.

# Risiko analyse

Her ses projektgruppens risikoanalyse.

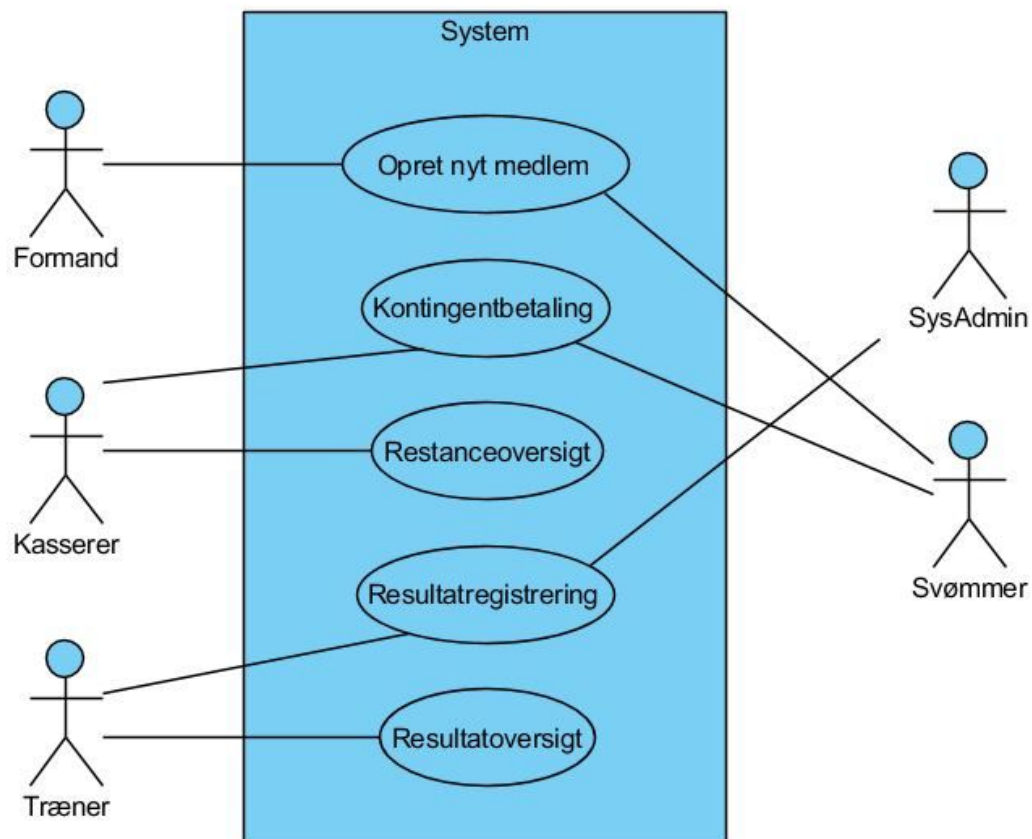
fx	A	B	C	D	E	F
1						
2		Risikomoment	Sandsynlighed	Konsekvens	Produkt	Løsningsforslag
3		Nøgleperson forlader projektet	1	3	3	Holder løbende kontakten
4		Nøgleperson bliver syg	4	2	8	Sørger for at alle kan overtage opgaver
5		Kan ikke levere til tiden (med prototype)	1	10	10	Sigter efter at være på forkant med buffer
6		Produkt lever ikke op til krav	7	1	7	Accepterer at det kun er en prototype som udgangspunkt, og får løbende vejledning
7						
8						
9						
10	Risiko 2.0	Risikomoment	Sandsynlighed	Konsekvens	Produkt	Løsningsforslag
11		Nøgleperson forlader projektet	1	1	1	Holder løbende kontakten
12		Nøgleperson bliver syg	4	1	4	Sørger for at alle kan overtage opgaver
13		Kan ikke levere til tiden (med prototype)	1	10	10	Sigter efter at være på forkant med buffer
14		Produkt lever ikke op til krav	10	1	10	Accepterer at det kun er en prototype som udgangspunkt, og får løbende vejledning
15		Ikke udarbejdet rapport	1	10	10	Sørger for at være færdige med buffer tid
16		Mister færdigt arbejde	1	9	9	Sørger for at have taget ekstra backups
17						
18						

Risikoanalysen er blevet skrevet ud fra gruppens perspektiv og ikke ud fra selve systemet. Vi udarbejdede den første risikoanalyse i starten af projektet, før vi var gået i gang med selve programmet. Den opdaterede version blev skrevet et par dage før vi var afleveringsparate.

- Risiko 1.0. Blev skrevet i betragtning af, hvordan selve gruppen kunne slå fejl og hvordan vi kunne dæmme op for nogle af de eventuelle problemer. Vi aftalte ud fra analysen at mødes hver hverdag og planlægge, hvor meget vi ville nå den pågældende dag, og hvad vi ville arbejde på hjemme inden næste dag. På den måde forsøgte vi at komme risiciene i forkøbet ved at holde et stramt skema med mulighed for hjemmearbejde, hvor det var nødvendigt.
- Risiko 2.0. Skrev vi med et henblik på selve opgaverne. F.eks var gruppen næsten færdige med rapporten, og vi vidste derfor, at sandsynligheden for at den ikke blev færdig, var relativ lav. Vi vidste også på daværende tidspunkt, at produktet ikke ville leve op til alle kravene, og at det ville forblive en prototype til afleveringen. Vi sørgede for, vi havde alle filer på 3 computere og i skyen, så ingen data gik tabt.

# Use Case Diagrammet

Her under ses det Use Case Diagram vi har udarbejdet på baggrund af vores Use Cases.



Use Case Diagrammet sammenhængen mellem use cases og aktører i system. Da vi ikke har færdiggjort hele systemet og kun har udviklet en prototype, er det ikke alle use cases, der er blevet færdigudviklet. Det drejer sig om Resultatregistrering og Resultatoversigt. Med udgangspunkt i use casen: Opret nyt medlem, kan man se overordnet at der kommet til at være en funktion i programmet der opretter et nyt medlem. For at bruge funktionen Opret nyt medlem, skal Formanden(Primary actor), bruge en Svømmer(Supporting actor), da det i dette tilfælde er den primary actor der skal gøre noget med de informationen han for fra den supporting actor.

# Use Cases

Vores use cases blev skrevet en af de første arbejdsdage og blev senere opdateret. Use cases'ne beskriver her, hvordan funktionerne skulle se ud, men de beskriver ikke nødvendigvis funktionerne i prototypen.

---

## Use Case: Create new member

**Primary actor:** Chairman

**Precondition:**

A person would like to become a member of the swimclub.

**Main Success scenario:**

The Chairman ask the system to create a new member. System ask the Chairman if the new member will be competitive. The Chairman chooses yes or no. He then enter first-, last name & age. The system automatically set the member to active. The system the create a new member and saves the info in the "Member" document.

---

## Use Case: Subscription payment

**Primary actor:** Member

**Precondition:**

At the beginning of each year, the member is asked to pay a value corresponding to their payment level. Payment level is set according to age and whether the membership is active or not.

**Main success scenario:**

A member would like to pay their Arrears. The Cashier check the swimmers arrears with their ID in the system. The system print the arrears. The swimmer pays the Cashier. The Cashier types the amount into the system. The system updates the members Arrears.

**Extensions:**

- If the member cannot pay in full, the system updates the balance of the account accordingly.
  - If the member tries to pay more than the balance value, the system cancels the payment and gives a message to signal this.
- 

## Use Case: Arrears overview

**Primary actor:** The Cashier

**Main success scenario:**

The Cashier opens the system. The cashier chooses an option to check arrears. The system makes a list of all unpaid memberships based on ID's. The system prints out a message showing arrears overview.

**Extensions:**

If there are no unpaid memberships, the system doesn't print anything.

---

## Use Case: Resultatregistrering

**Primary actor:** Trainer

**Supporting actors:** System administrator

**Prerequisite:**

Member participates in competition or training and sets a new best time in a discipline.

**Main Success Scenario:**

Trainer records the new best time, discipline, and competition name if applicable. Trainer opens system and chooses an option to register new best time. Trainer chooses either training or competition. Trainer enters the member ID and registers the new best time in seconds, with a date, and with competition ID if applicable. System updates with new information.



**Extension:**

- The new best time isn't better. The system doesn't update and gives a corresponding message.
  - Trainer accidentally registers a wrong time/date/competition/discipline. System administrator manually changes back to the correct information if available
- 

## **Use Case: Result chart**

**Primary Actor:** Trainer

**Main success scenario:**

The trainer would like to set the team for an upcoming competition. He opens the system and chooses the Result-chart. The system asks if the trainer would like to see junior or senior results. The system asks what type of swimming style he would like to see. The trainer chooses the desired style, and the system shows the 5 best swimmers in the given style.

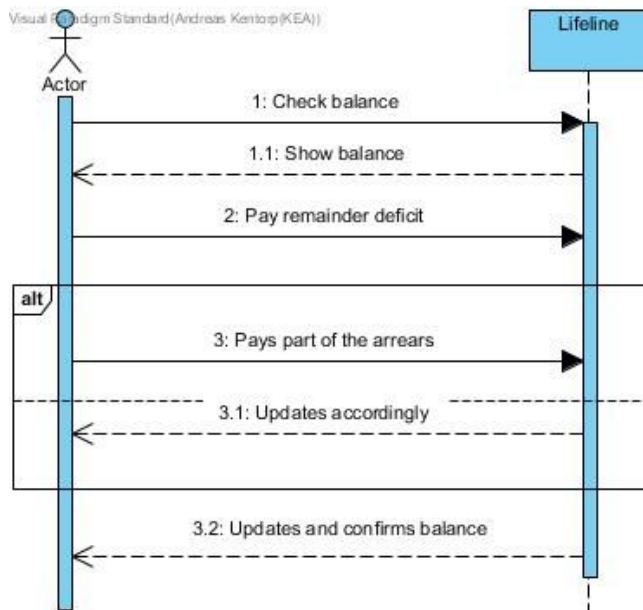
**Extensions:**

if there are less than 5 or no results in the chosen bracket, the system prints less or no results.

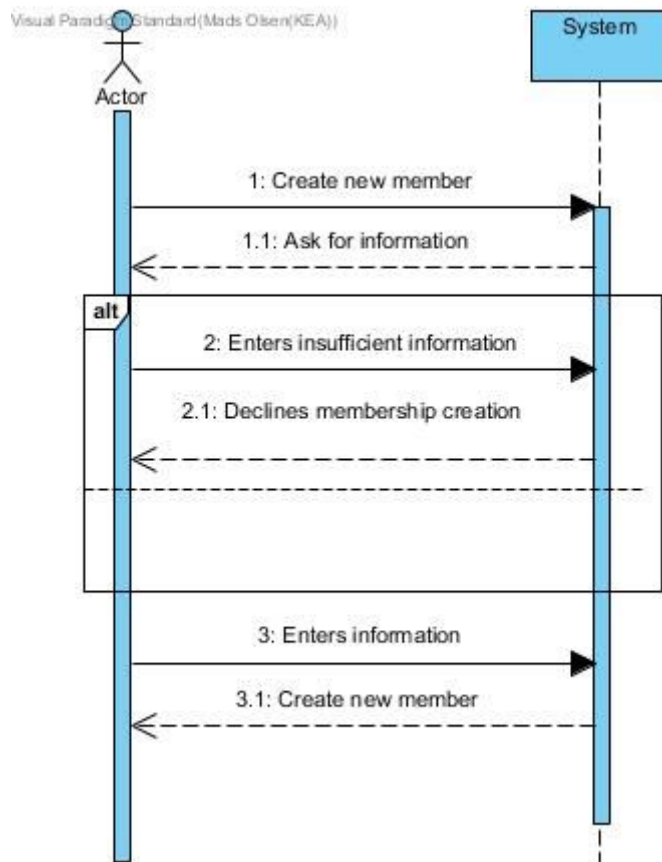
## System Sequence Diagrams

SSD'erne beskriver visuelt, hvordan use cases tager udformning i systemet. Actor'en giver input, som bliver besvaret med et output fra systemet. Alternative flows er vist i kasser og beskriver hvad der sker, hvis actor'en ikke følger *main success scenario*.

### Subscription payment SSD

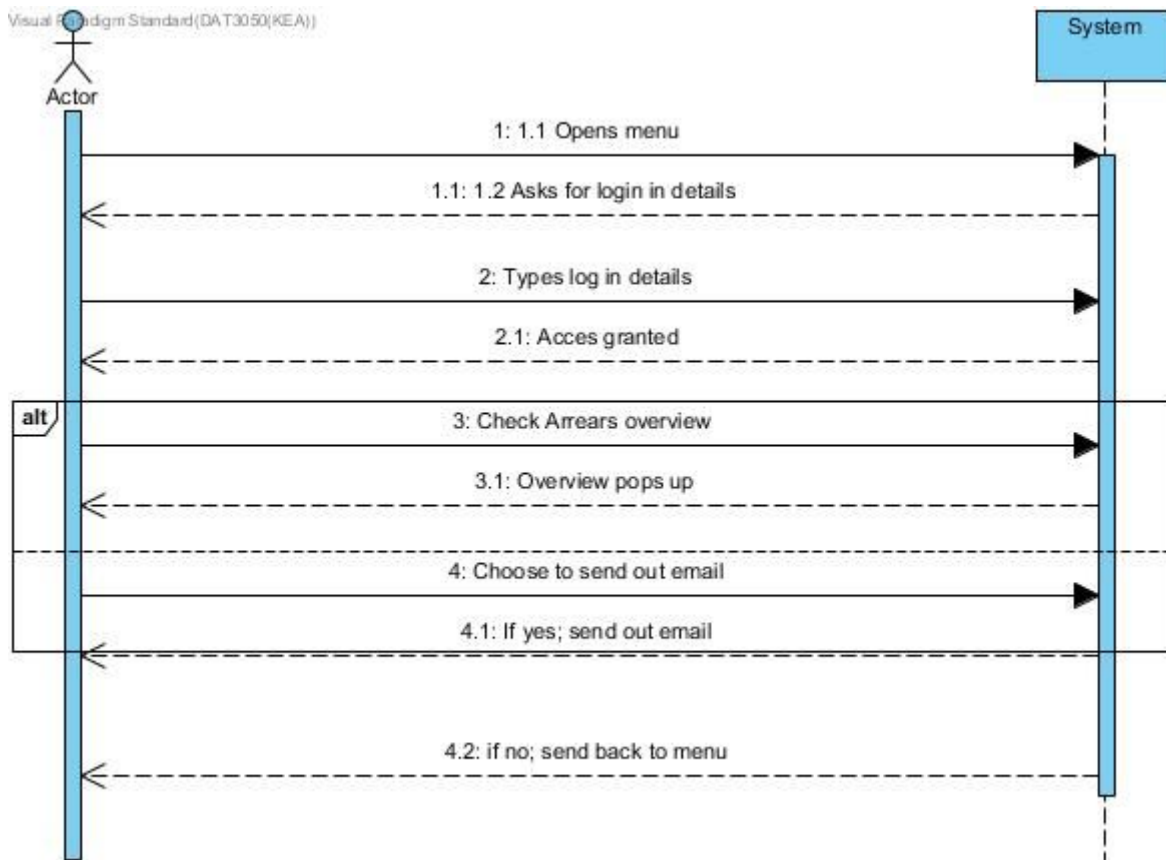


## Create New Member SSD

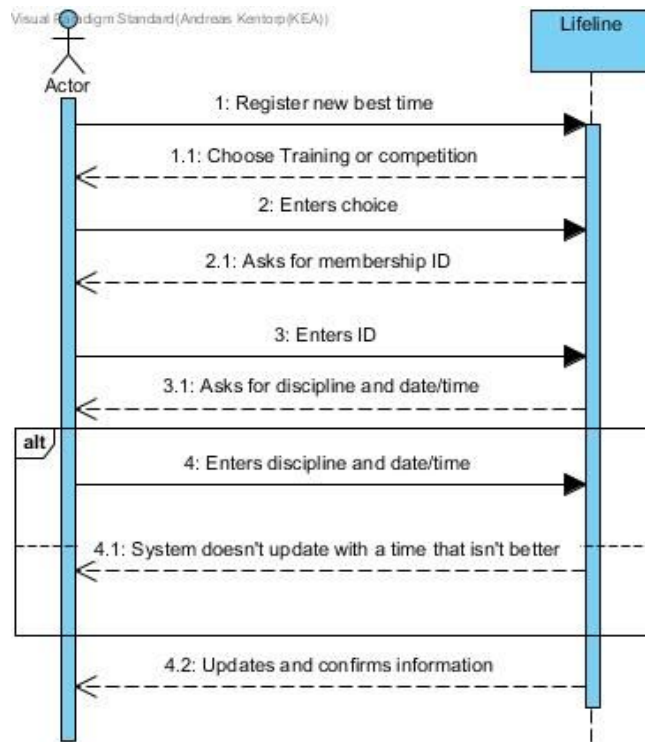


## Check Arrears SSD

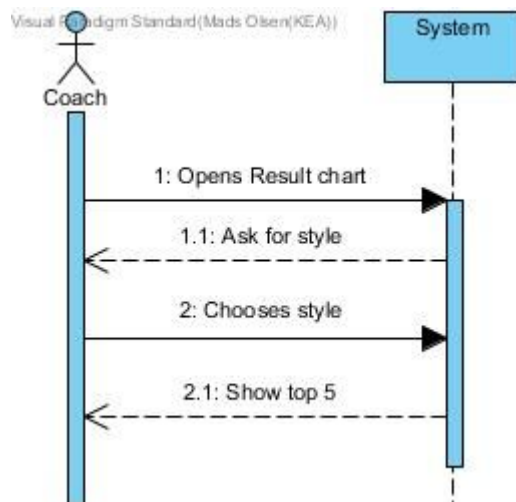
Visual Paradigm Standard (DAT3050(KEA))



## Result Registration SSD

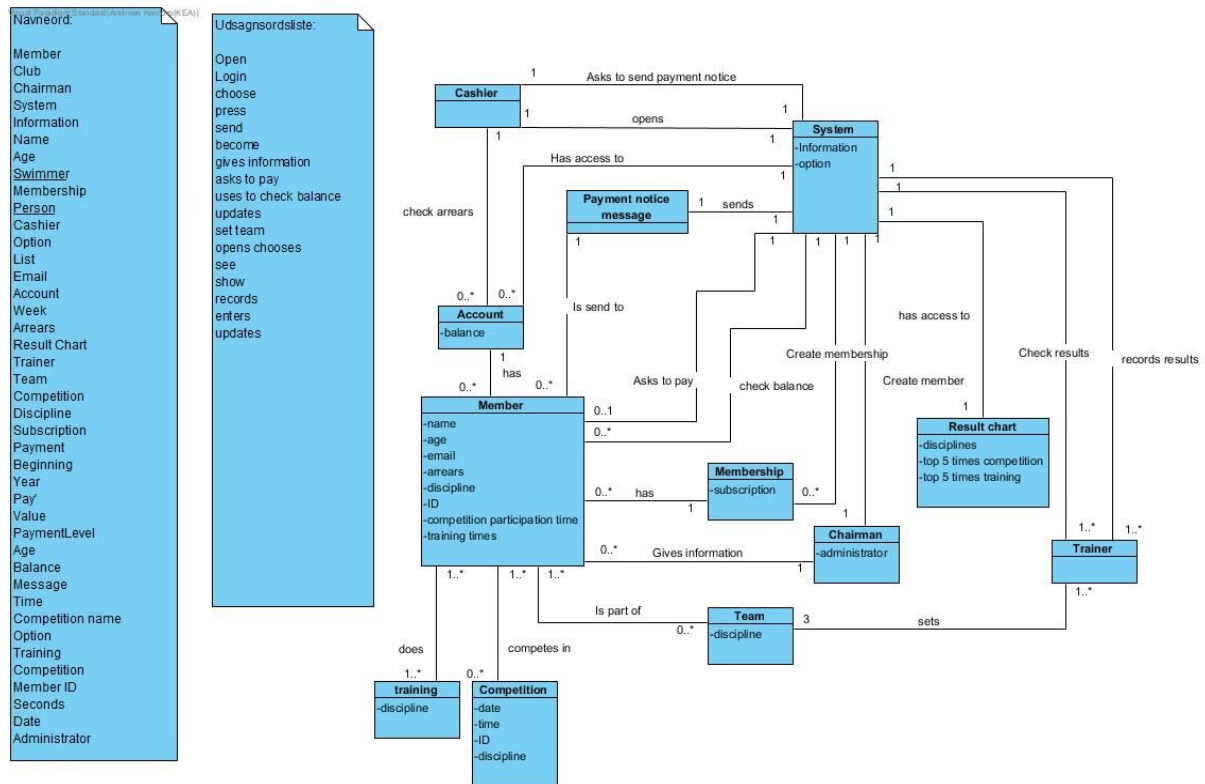


## Results Chart SSD



## Domain model

Her under ses en domænemodel, der viser hvad der sket i den virkelige verden. Det er også på baggrund af den, at vi kan se hvor vi skal rette vores klasser hen.



På baggrund af alle navneord i use cases'ne, udarbejdes der en navneord liste. Navneords listen bliver derefter reduceret, og der laves domæner, som er med til at vise et abstrakt element af systemet. Derefter udarbejdes en udsagnsords liste på samme måde som med navneords listen. Udsagnsordene viser hvordan de forskellige domæner agere med hinanden. Multiplicity viser hvor mange der kan forekomme af hvert domæne. F.eks kan en chairman, ved at bruge de informationer han får fra et member, bruge funktionen "Create new Member" i systemet, som så opretter et member og gemmer de pågældende data.

# Klassediagram

klassediagrammet viser hvilke attributter, primitive typer og metoder vi brugt i de forskellige klasser. Med klassediagrammet beskrives det også, hvilke klasser interagerer med hinanden.

