

# Delfinen

---

## Stræber centralen:

- Christopher Marc Lønborg/ Chri402s@stud.kea.dk/ Ceazer101
- Güler Nur Dönmez/ Gule0153@stud.kea.dk/ Gulernur
- Kenneth Lønfeldt Frederiksen/ Kenn741q@stud.kea.dk/ Kennlf
- Maja Matboue/ Maja962h@stud.kea.dk/ Maja962h
- **GitHub repository link:**
- <https://github.com/maja962h/DelfinenRedo>

9. december 2021

KEA Dat21D

## Indledning

Dette projekt vil være et program til Svømmeklubben, Delfinen, der skal hjælpe dem med at få et overblik over deres medlemmer, kontingent, træningsresultater mm. Programmet skulle gerne hjælpe klubbens ledelse i deres daglige arbejde med administration af klubben.

Programmets intention er at optimere arbejdsgangen ved at lade forman oprette og slette medlemmer i systemet, lade kassereren holde styr på manglende betalinger fra medlemmer og den totale kontingentindtægt, og sidst, men ikke mindst, lade trænerne registrere svømmernes træningstider og konkurrence resultater.

I opgaven forventes der at følgende use cases bliver implementeret:

- Oprette medlemmer i systemet.
- Se medlemsliste.
- Check om medlemmer har betalt.
- Overblik over total kontingentbetaling.
- Se top 5 svømmere inden for hver disciplin.
- Registrere svømmernes resultater, løbende.
- Registrere tid, dato og sted for stævner.

## Afgrænsning

Håndtere ubetalt kontingent, Forventet total kontingent, top 5, regi tider, regi stævne, placering

### Implementering af Use-Case:

Den første Use-Case vi vil implementere, er at 'oprette' og 'gemme' medlemmer. Dette gøres fordi det er essentielt for programmet, at man har mulighed for at tilføje klubbens medlemmer til databasen, så data'en sidenhen kan håndteres.

Herefter vil vi implementere 'Se medlemsliste'. Der vil være forskellige udgaver af denne Use-Case, for at kunne se forskellige oplysninger omkring klubbens medlemmer. Dette kunne for eksempel være: hvilken type kontingent medlemmerne betaler, om de er aktive medlemmer, eller om de er motionister eller konkurrencesvømmere. Vi har valgt at prioriterer denne Use-Case ret højt da den kan bruges til mange forskellige ting, og af flere brugere i systemet, baseret på hvad det er nødvendigt for den specifikke bruger at se.

De næste Use-Cases vi vil implementere, vil være 'Håndtering af ubetalt kontingent' og 'Forventet total kontingent'. Her er der tale om nogle funktioner som påvirker mange medlemmer og klubben overordnet set, så det er derfor en stor fordel at kunne få det overblik.

Som noget af det sidste vil vi implementere Use-Cases: 'se top 5 rangliste', 'registrere træningstid' og 'registrere tid og placering for stævne'. Grunden til at disse er prioriteret som noget af det sidste er at det er funktioner som vil påvirke færrest mulige medlemmer og derfor også vurderes at have den mindste indflydelse i brugernes anvendelse af programmet.

## Interessentanalyse

Nedenunder ses projektets interessentanalyse.

| Interessentanalyse |   |   |  |   |   |   |
|--------------------|---|---|--|---|---|---|
| Interessenter      | Deres mål   | Tidligere reaktion  | Hvad der kan forventes   | Indvirkning Pos/neg   | Mulig fremtidig reaktion  | Ideer                                       |
| Formand            | Godt produkt der gør arbejdet nemmere                           | Frygt for at programmet ikke vil løse hans udfordringer tilstrækkeligt / komplicerer det yderligere | Bliver frustreret hvis tingene ikke går som forventet.                     | - Negativ hvis tingene går galt. -Meget positiv hvis det går godt.  | Kunne nægte at anvende programmet hvis ikke han har tiltro til det/manglende evner.       | Holdes opdateret. Løbende test/involvering. |
| Kasserer           | Godt produkt der gør arbejdet nemmere. Fokus på pris / kvalitet | Frygt for spildte penge/større omkostninger end forventet.  | Bliver frustreret hvis projektet tager længere tid/bruge flere ressourcer. | - Negativ budget/deadline overskrides. - Positiv hvis alt går som planlagt og de får en produkt der kan bruges. | Nægte at gøre projektet færdigt hvis budget/omkostning overskrides i for stort et omfang. | Holdes opdateret. Løbende test/involvering. |

|                   |  |   |  |   |  |  |
|-------------------|--|---|--|---|--|--|
| Træner            | Et program der er nemt at anvende og overskueligt.   | Frygt for at der bliver rettet i tider/placerin g for deres respektive ansvarsområ de.                    | Bliver frustreret hvis programmet er uoverskueli gt.   | Negativ hvis programmet ikke er nemt at benytte.<br><br>Positiv hvis det er brugervenligt , let overskueligt så det ikke tager mere tid fra hans tid som træner.        | Nægte at anvende programmet hvis det ikke er overskueligt /mangler informationer   | Løbende test med relevante funktioner.                         |
| Stræber centralen | Et rimelig godt produkt der opfylder kundens vigtigste krav.<br><br>Et program der ikke crasher.<br><br>Blive færdige hurtigst muligt. | Frygt for ikke at nå kundens vigtigste krav.<br><br>Frygt for at have misforstået kundens vigtigste krav. | Frustreret hvis tingene ikke går som forventet.<br><br>Frustreret hvis der forekomme r ændringer i krav. | Negativ hvis deadline overskrides.<br><br>Positiv hvis produktet bliver færdig til deadline.<br><br>Positiv hvis programmet er godt og opfylder kundens vigtigste krav. | Kunne nægte at færdiggøre programmet hvis der forekommer væsentlige ændringer til krav uden ekstra tid og større budget. | Løbende dialog med kunden.<br><br>Løbende brugertest/ feedback |

## Risikoanalyse

Nedenunder ses projektets risikoanalyse.

Vi har benyttet følgende skala for vores vurderinger:

- Meget lav (<10%) værdien 1
- Lav (10%-25%) værdien 2
- Moderat (25%-50%) værdien 3
- Høj (50%-75%) værdien 4
- Meget høj (>75%) værdien 5

| Risikoanalyse                     |               |            |         |
|-----------------------------------|---------------|------------|---------|
| Risikomoment                      | Sandsynlighed | Konsekvens | Produkt |
| Nøglepersoner forlader projektet. | 2             | 3          | 6       |
| Fatale fejlestimer                | 3             | 7          | 21      |
| Mangelfuld testning               | 3             | 7          | 21      |
| Sygdom                            | 4             | 3          | 12      |
| Store ændringer til systemkrav    | 4             | 7          | 28      |
| Manglende uddannelse/introduktion | 4             | 3          | 12      |

## Risikomatrix

Nedenunder ses projektets risikomatrix.

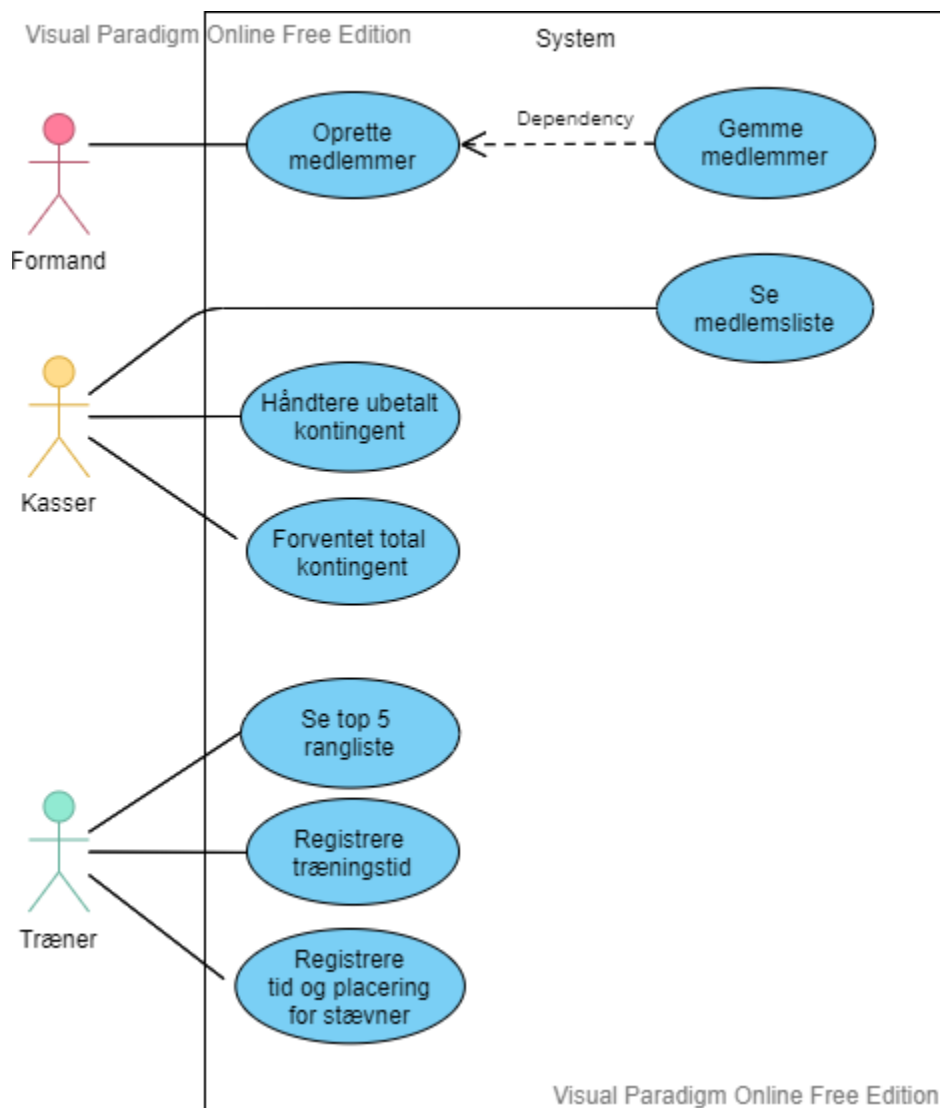
Vi har benyttet følgende skala for vores vurderinger:

- Ubetydelig                værdien 1
- Tålelig                    værdien 3
- Alvorlig                  værdien 7
- Katastrofal              værdien 10

| Risikotabel                       |               |            |         |   |   |  |                                   |
|-----------------------------------|---------------|------------|---------|---|---|--|-----------------------------------|
| Risikomoment                      | Sandsynlighed | Konsekvens | Produkt | Præventive tiltag   | Ansvarlig                                 | Løsning forslag  | Ansvarlig                         |
| Nøglepersoner forlader projektet  | 2             | 3          | 6       | Dokumenter arbejdsproces.<br><br>Dokumenter krav og specifikationer fra testbruger. | Projektleder og projektdeltagere          | Gennemlæs udarbejdet dokumentation.<br><br>Interview med kunde.<br><br>Evt. ny testbruger. | Projektleder/<br>Projektdeltagere |
| Store ændringer til systemkrav    | 4             | 7          | 28      | Interview med kunden før projektstart.  | Projektleder, projektdeltagere og kunden. | Møde med kunden om fordeling af ressourcer og deadline.                                    | Projektleder.                     |
| Manglende uddannelse/introduktion | 4             | 3          | 12      | Fremvisning af programmet på papir.<br><br>Løbende inddragelse af testperson/bruger | Projektleder, projektdeltagere og kunden. | Udarbejdelse af en simple guide til programmet.  | Projektleder/<br>Projektdeltagere |

## Use case model

Nedenunder ses projektets use case model. Denne model blev lavet i starten af projektets forløb, hvilket gav os bedre overblik over hvad programmet indeholder.



## Use case beskrivelser

### **Oprette medlemmer:**

*Giver formanden mulighed for at oprette nye medlemmer i systemet.*

- Et nyt medlem henvender sig til formanden om indmeldelse.
- Formanden starter oprettelsen af et nyt medlem.
- Formanden indtaster det nye medlems oplysninger.
- Systemet gemmer det nye medlems oplysninger.
- Medlemmet er registreret og formanden beder medlemmet om at henvende sig til kassereren for betaling.
- Medlemmet forlader formandens kontor.

### **Se medlemsliste:**

*Formand, kasserer og træner kan se listen over alle medlemmer i systemet.*

- Brugeren tænder for systemet.
- Brugeren navigerer i systemet og vælger punktet 'se medlemsliste'.
- Herunder, har de mulighed for at vælge om listen skal sorteres på en bestemt måde eller om de vil se den fulde liste af medlemmer.
- Programmet udskriver den liste brugeren har valgt på skærmen.
- Brugeren kommer tilbage til hovedmenuen og kan enten afslutte programmet eller de kan benytte programmet andre funktioner.

### **Håndtere manglende betaling af kontingent:**

*Kasserer kan få et overblik over medlemmer der ikke har betalt, samt ændre deres betalingsstatus.*

- Kassereren tænder for programmet.
- Kassereren navigerer i systemet og vælger punktet 'tjek kontingent status'.
- Programmet udskriver, på skærmen, en liste over medlemmer der ikke har betalt deres kontingent i indeværende sæson.
- Herefter, har kassereren mulighed for at rette enkelte medlemmers betalingsstatus.
- Når kassereren er færdig med dette, returneres kassereren til hovedmenuen og kan herfra afslutte programmet eller benytte programmet andre funktioner.

### **Forventet total-kontingent:**

*Kasserer kan se den forventede samlede kontingentbetaling.*



- Kassereren tænder for programmet.
- Kassereren navigerer i systemet og vælger punktet 'total-kontingent'.
- Programmet udregner den totale kontingent ud fra antallet af medlemmer i databasen.
- Programmet udskriver, på skærmen, den forventede samlede kontingentbetaling.
- Når kassereren er færdig med dette, returneres kassereren til hovedmenuen og kan herfra afslutte programmet eller benytte programmet andre funktioner.

### **Se top 5 rangliste:**

*Træner kan se top 5 svømmere indenfor hver disciplin.*

- Træneren tænder for programmet.
- Træneren navigerer i systemet og vælger punktet 'svømme rangliste'.
- Herunder, har træneren mulighed for at vælge hvilken disciplin træneren vil se .
- Programmet udskriver, på skærmen, en liste over top 5 svømmere ud for den valgte disciplin.
- Når træneren er færdig med dette, returneres kassereren til hovedmenuen og kan herfra afslutte programmet eller benytte programmet andre funktioner.

### **Registrere træningstid:**

*Træner kan registrere svømmernes bedste træningstider.*

- Træneren tænder for programmet.
- Træneren navigerer i systemet og vælger punktet 'registrer træningstid'.
- Nu, kan træneren indtaste svømmetid og dato for træningstiden.
- Systemet gemmer oplysningerne.
- Når træneren er færdig med dette, returneres kassereren til hovedmenuen og kan herfra afslutte programmet eller benytte programmet andre funktioner.

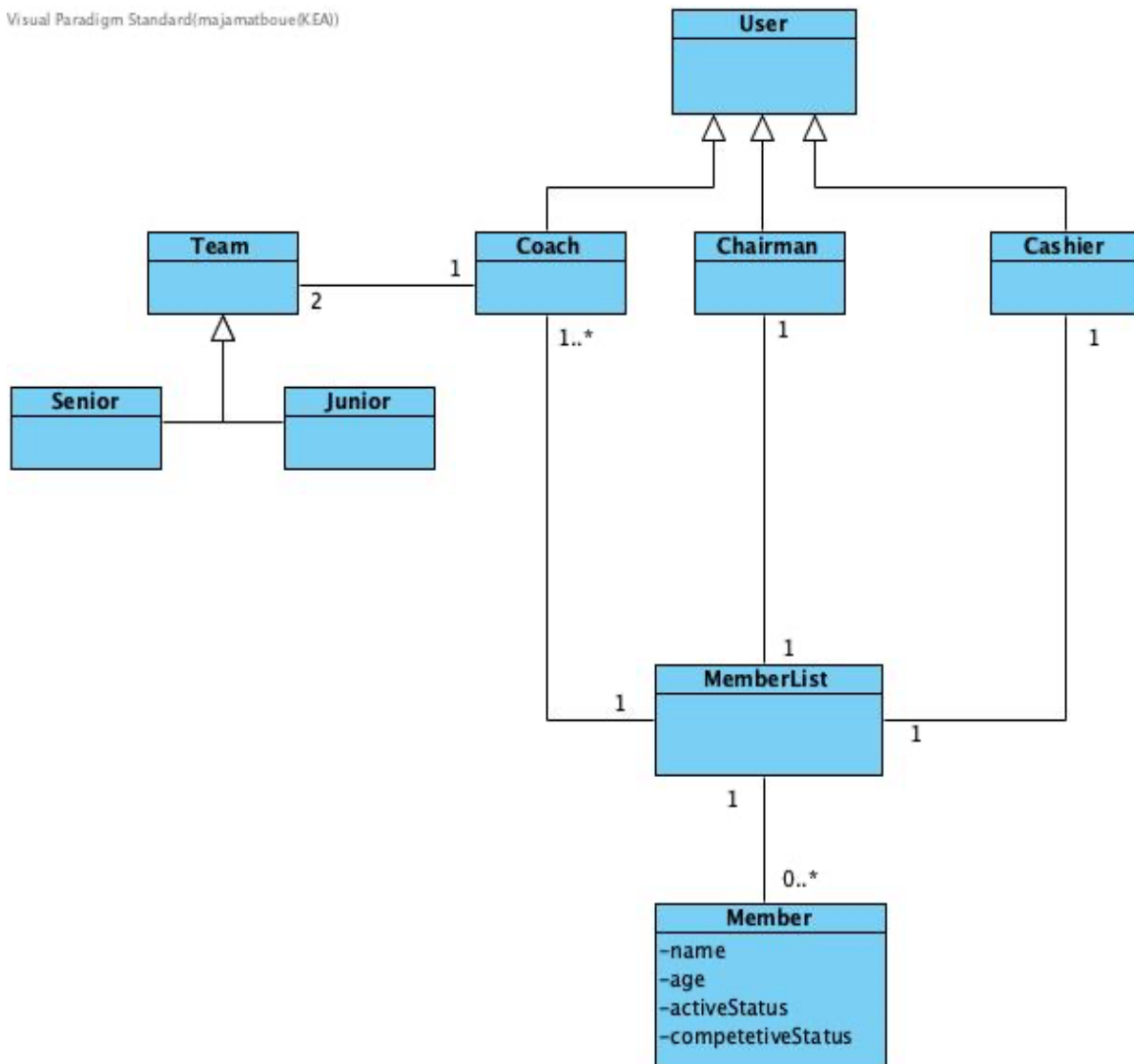
### **Registrere tid og placering for stævner:**

*Træner kan registrere svømmerens bedste tid og placering for stævner.*

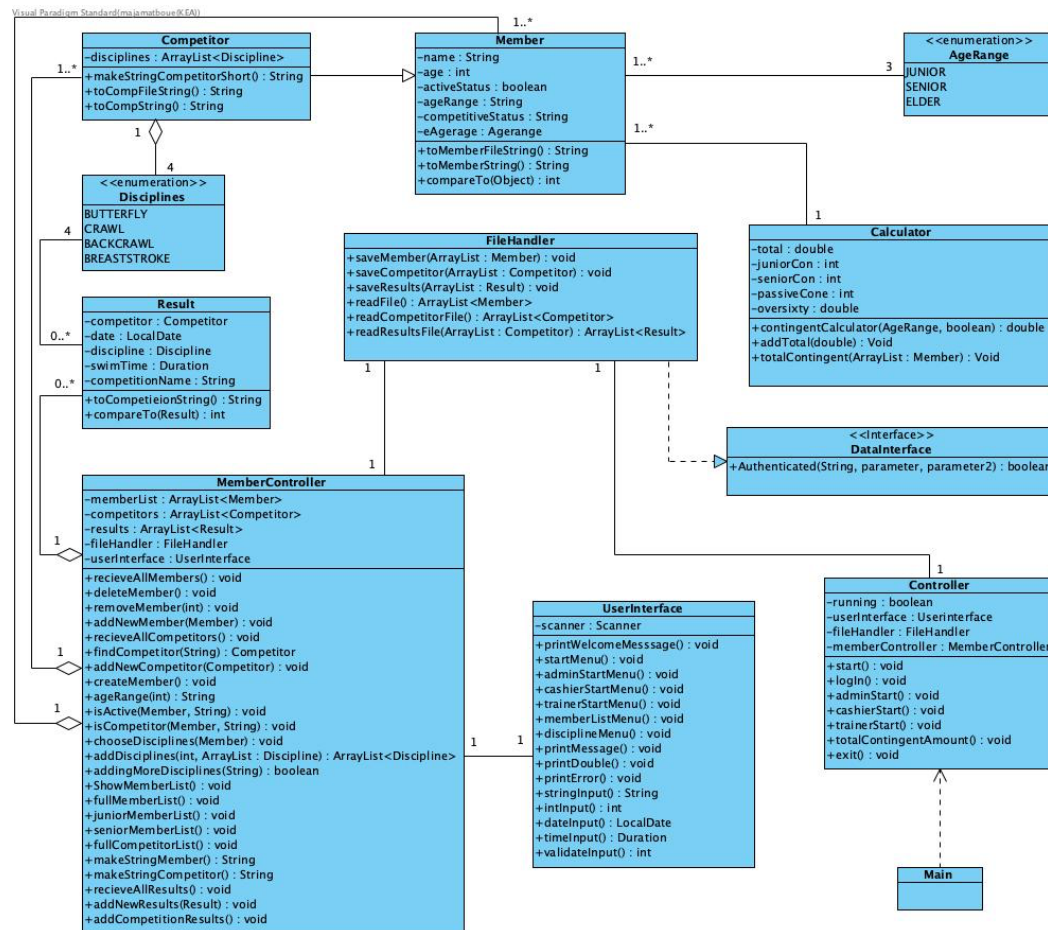
- Træneren tænder for programmet.
- Træneren navigerer i systemet og vælger punktet 'registrer bedste konkurrence placering'.
- Nu, kan træneren indtaste tid, placering og dato evt. konkurrencens navn.
- Systemet gemmer oplysningerne.
- Når træneren er færdig med dette, returneres kassereren til hovedmenuen og kan herfra afslutte programmet eller benytte programmet andre funktioner.

# Domænemodel

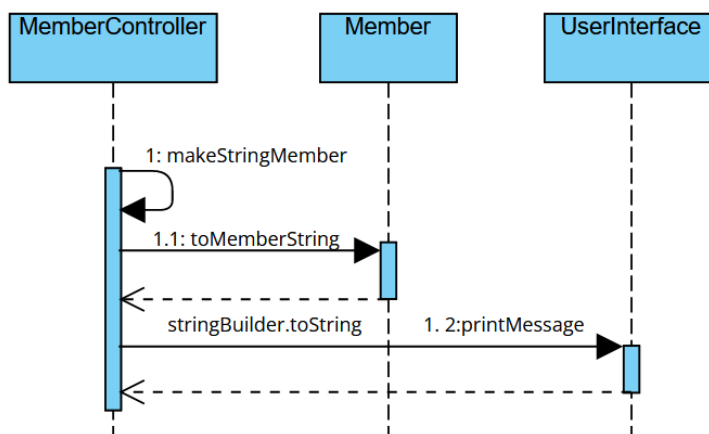
Visual Paradigm Standard(majamatboue(KEA))



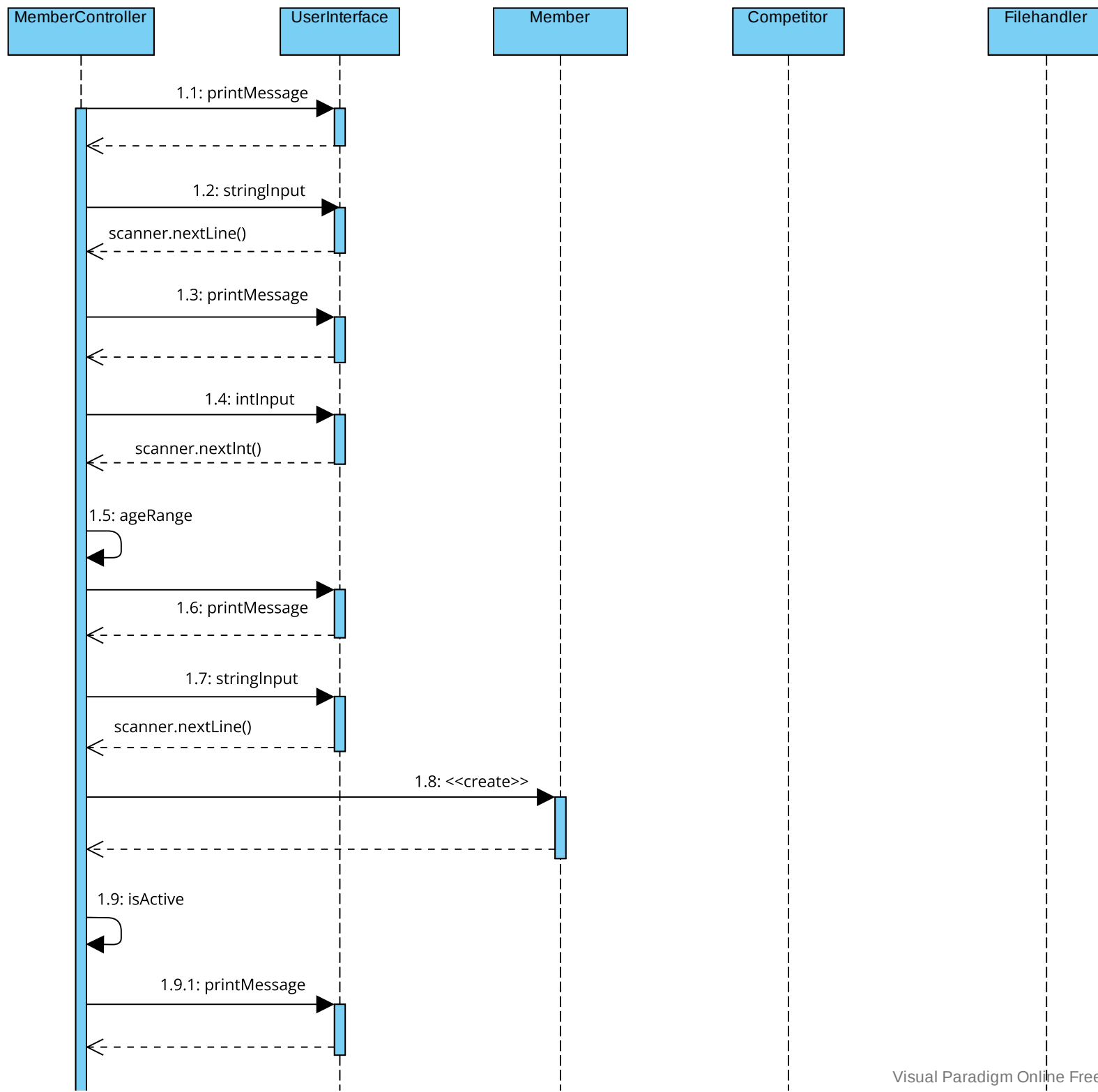
# Klassediagram

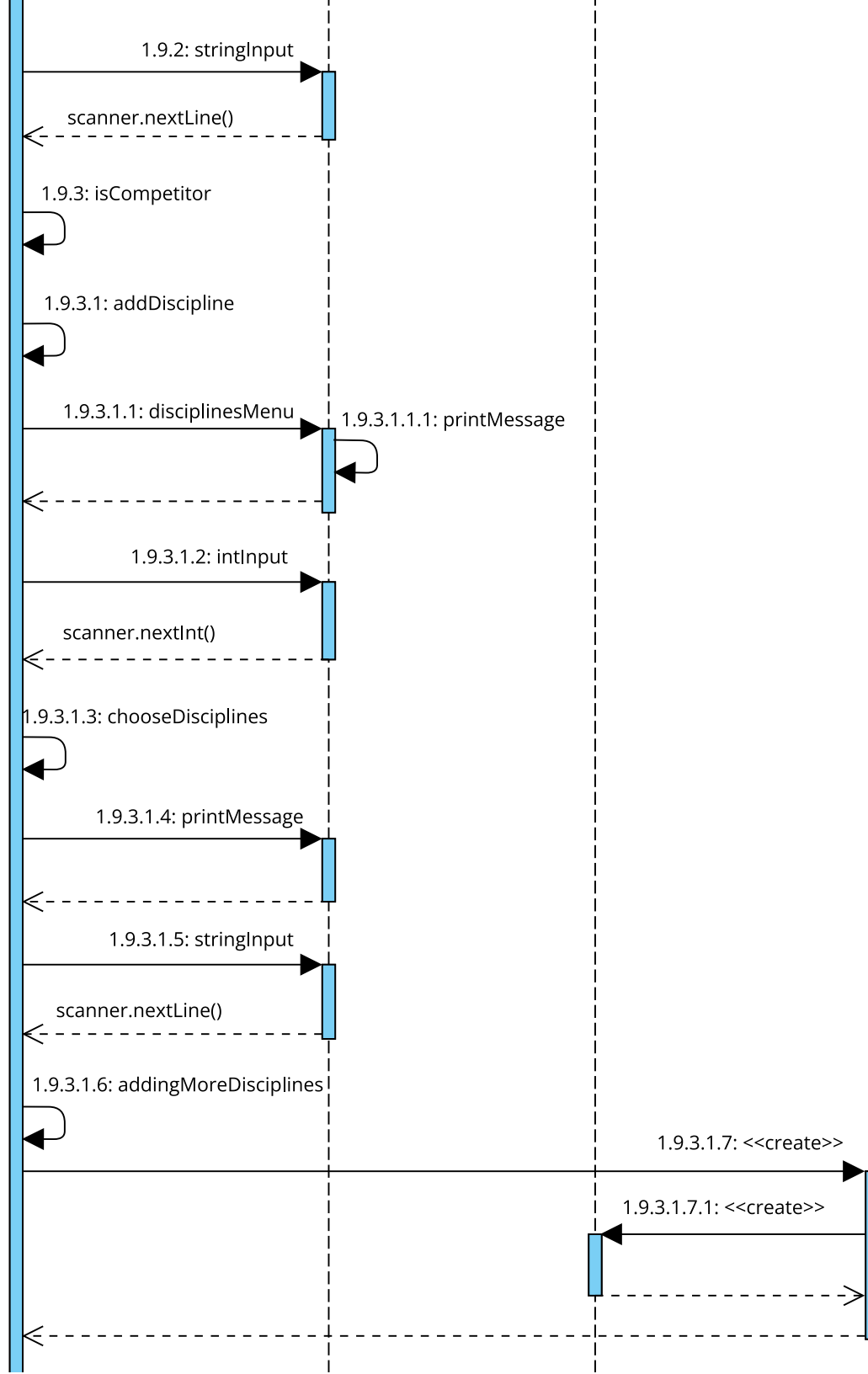


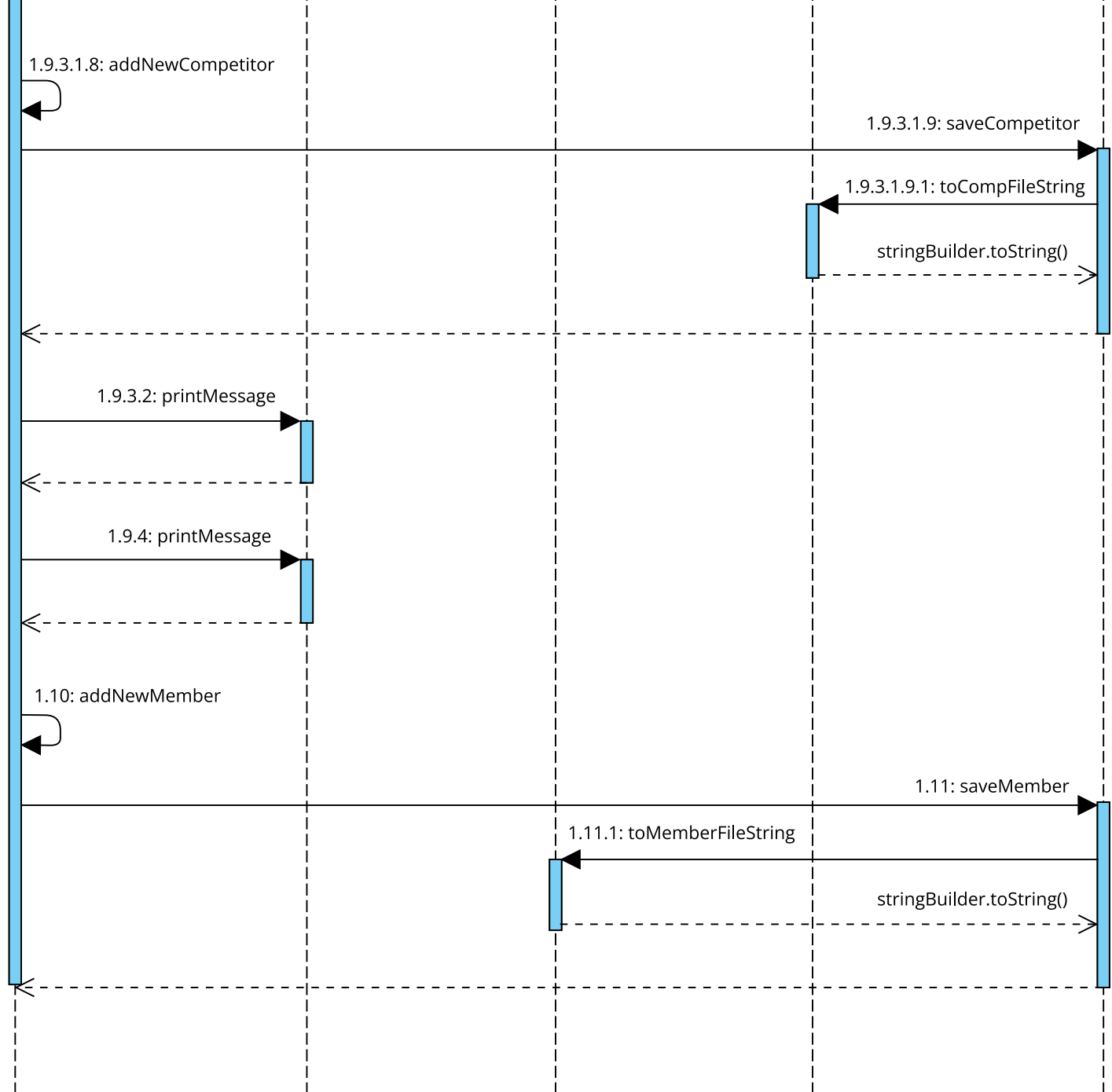
## Sekvensdiagram ( 1 ) showMemberList()



# Sekvensdiagram ( 2 ) createMember()







## Status på implementering:

### Use case:

*Ved aflevering af dette projekt, er følgende use cases implementeret:*

- Oprette medlemmer
- Se medlemsliste
- Overblik over total kontingentbetaling
- Registrerer tid, dato og sted for stævner.
- Gemme medlemmer til fil.

*Ved aflevering af dette projekt, er følgende use cases ikke implementeret helt:*

- Se top 5 svømmere inden for hver disciplin.
- Redigere (løbende registrere) svømmeresultater.
- Det er ikke muligt at håndtere/administrere ubetalt kontingent.

### Bugs:

- Hvis man indtaster den forkerte datatype i visse metoder, vil programmet fejle.
- Man kan ikke slette en konkurrencesvømmer, hverken fra arraylisten eller i filen.
- Man kan ikke slette et medlem, hverken fra arraylisten eller i filen.