

## Delfin SP5

Vi har til vores study point projekt nr. 5 valgt Delfin opgaven. Vi valgte denne opgave på grund af den lignende struktur fra forhenværende opgave om Marios Pizzaria. Vi mente at det var en god lejlighed til at dygtiggøre os mere indenfor OOP og især MySQL. Delfin opgaven viste sig at være en væsentlig mere omstændig opgave i forhold til SP4 afhængig af hvor langt man vil gå med opgaven. Derfor har der været diverse features som vi ikke har lykkedes at komme helt i mål med, men vi er overordnet tilfredse med vores afsluttede projekt.

Eksempler på disse features er:

- Et mere automatisk fungerende betalingssystem: Vi havde en idé om, at når en ny member blev tilføjet til svømmeklubben (og dermed også til databasen) ville denne member også blive tilføjet til vores payment table. Her har vi en boolean *hasPaid* som der altid vil blive sat til false, indtil memberen har betalt beløbet.

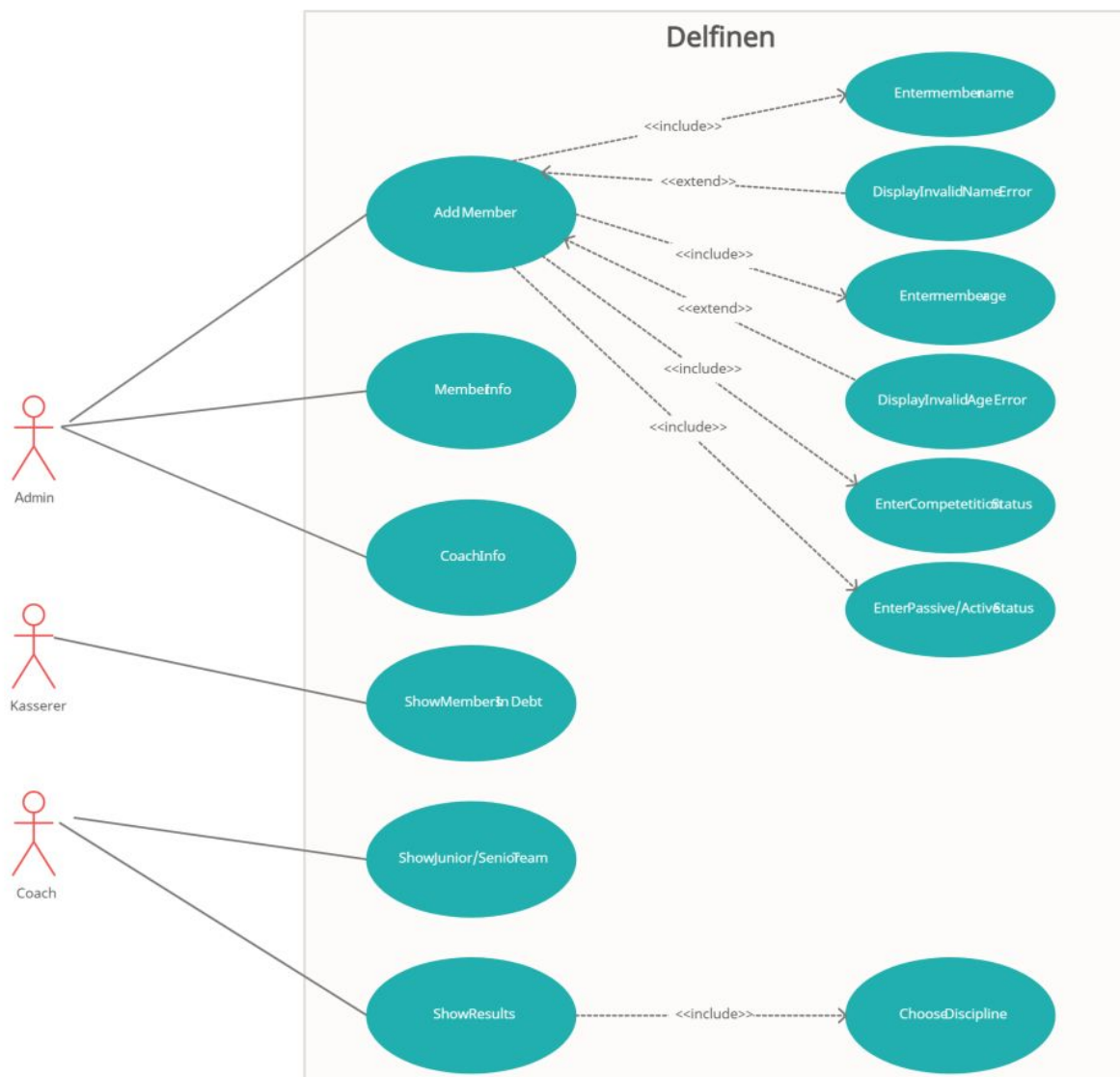
Udover ovenstående, mener vi selv, at vi har opfyldt de resterende krav, men der er selvfølgelig ting der kan forbedres. Heriblandt vores UI og navigering af programmet samt kodens struktur.

## Kravspecifikation:

Delfin SP5 går ud på at fremstille et administrativt IT-system til svømmeklubben Delfinen. Svømmeklubben fremstiller nogle krav der skal udføres, disse krav er følgende:

1. Svømmeklubbens formand ønsker en metode til at tilføje nye medlemmer til klubben samt at holde styr på diverse medlemsoplysninger, herunder kontingent og stamdata.
2. Svømmeklubbens trænere ønsker et system der kan gøre det nemmere for de individuelle trænere at udtage enkelte konkurrencesvømmere til svømmestævner. De vil gerne se en liste over alle konkurrence svømmere samt deres resultater, gerne vist på de hold de selv træner. Det skal være nemt at se de bedste resultater ud fra hver disciplin på begge hold.
3. Svømmeklubbens kasserer, der står for hvert medlems betaling, vil gerne kunne se en liste over de medlemmer som står i restance, så der kan ses hvem der mangler at betale.

Use case:



## **Main Priority Use Case – Add member**

### **Primary actors:**

Admin

Cashier

Coach

### **Preconditions:**

None

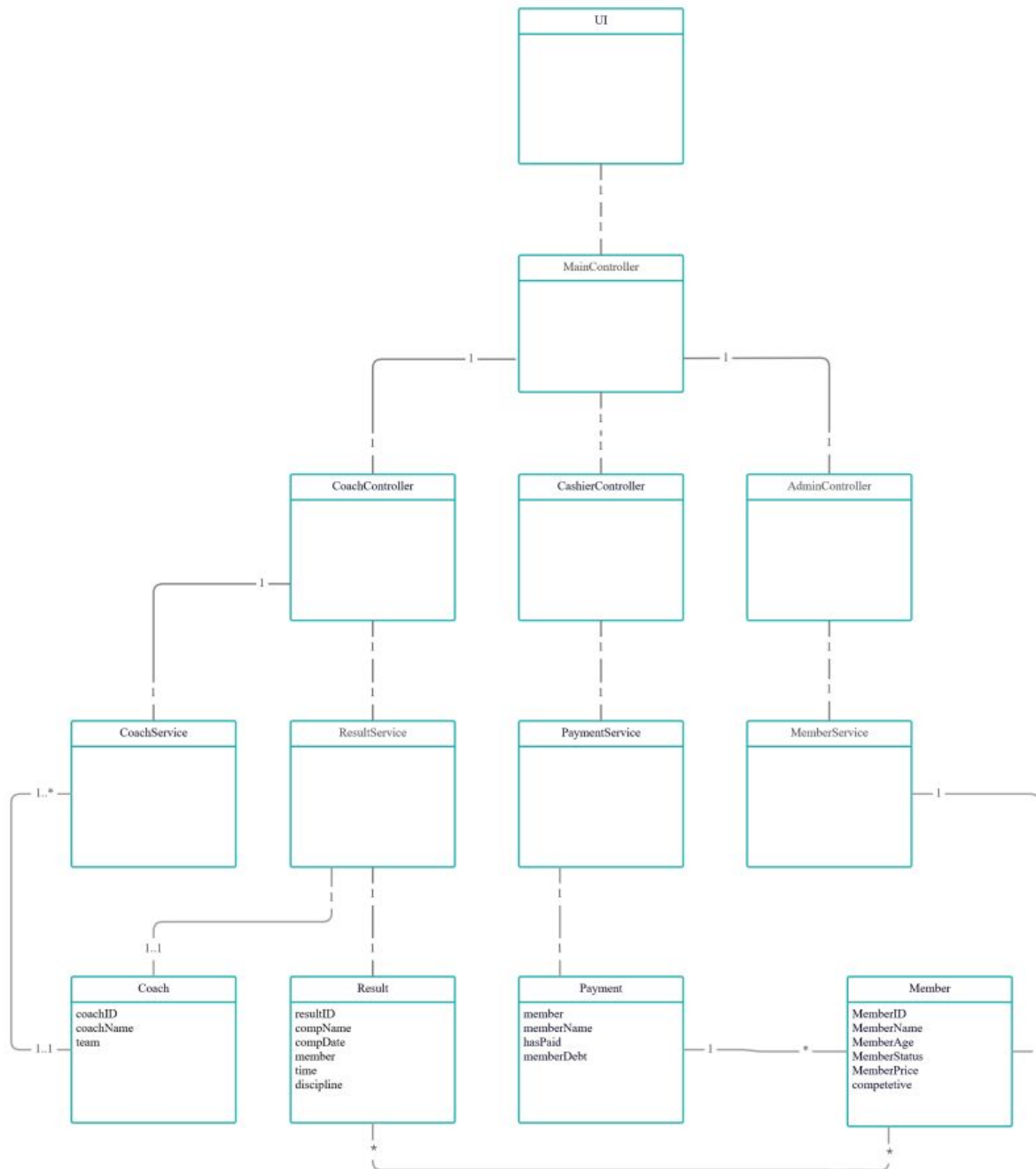
### **Basic flow of event:**

1. Enter name of new member
2. Enter age of new member
3. Enter status of new member (active/passive)
4. Enter status of competitiveness (competitive/exercise)
5. Member is added

### **Alternative flows:**

- If name is empty
  - o Program displays error message
  - o User can try again
- If age is 6 or under
  - o Program displays error message
  - o User can try again

## Domænemodel:



## ER-Diagram:

