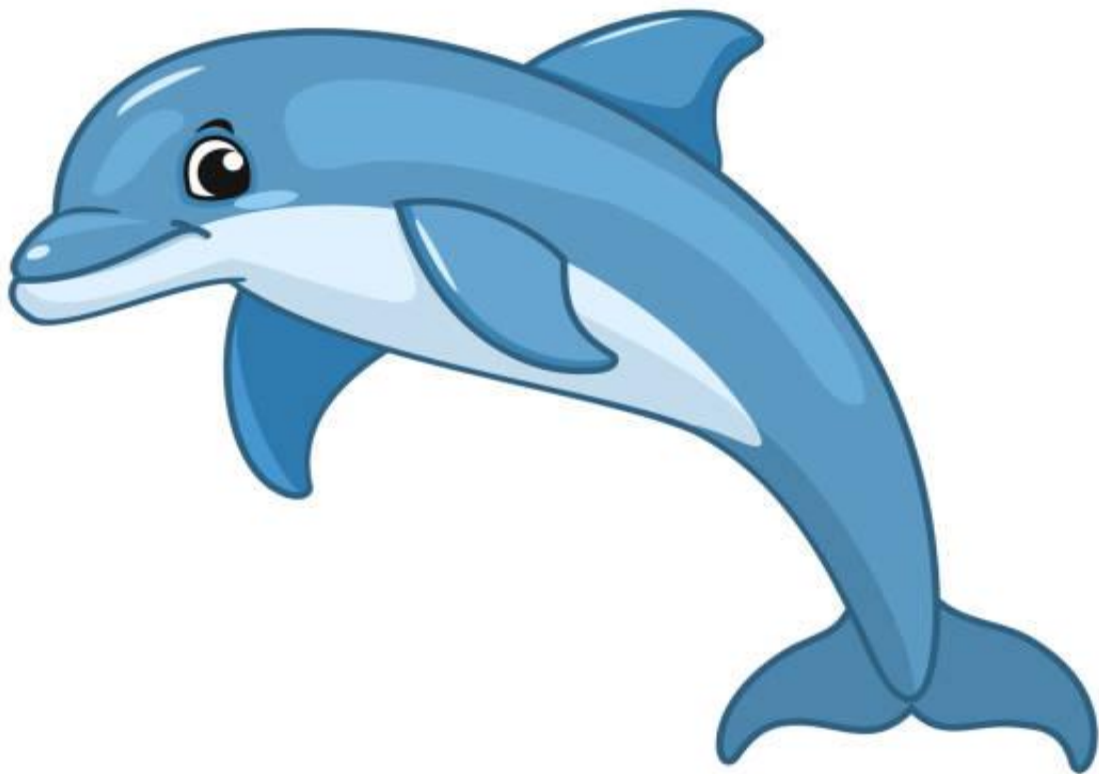


Svømmeklubben

Delfinen



Svømmeklubben	
Delfinen	1
Vision	3
Glossary	4
Use cases	5
Fully Dressed use case I - Indregistrering af nyt medlem	5
Fully Dressed use case II - Tjek restance for medlem	8
Fully Dressed use case III - Registrer konkurrenceresultat i butterfly for seniorsvømmere	10
Domain Model	12
Class Diagram	13
System Sequence Diagrammer	14
SSD I - Indregistrering af nyt medlem	14
SSD II - Tjek restance for medlem	15
SSD III - Registrer konkurrenceresultat i butterfly for seniorsvømmere	16
Overvejelser ift. GRASP	17
Information Expert	17
Creator	17
Low coupling	17
High cohesion	17
Polymorphism	18
Afgrænsninger og antagelser	19
Bilag	20

Vision

Vi vil udforme et system i form af et medlemsoverblik, som skal sikre, at Svømmeklubben Delfinens medarbejdere kan gennemføre administrative opgaver hurtigt og effektivt.

Vi ønsker at tage højde for de respektive medarbejdere, som skal benytte systemet - heriblandt formanden, svømmetrænere og kasserer. Med dette for øje, forventer vi at begrænse medarbejders muligheder, ved hjælp af et simpelt Login-system.

Medlemsoverblikket skal kunne varetage medlemsadministration og -segmentering, journal for konkurrencesvømmers resultater og økonomien ifm. kontingentbetaling, herunder gæld og forventet indkomst.

Glossary

- ❖ Konkurrencesvømmere og deres svømmediscipliner
 - Vi tracker kun konkurrencesvømmerne på en distance i hver disciplin; 100 meter.

Grunden til dette er, at der ellers ville være alt for mange discipliner at tracke.
- ❖ Kontingent/Contingent
 - Fastsat beløb som er et medlems årlige kontingent, beløbet ændres ikke løbende.
- ❖ Restance/Arrears
 - Beløb som instantieres jf. kontingentet. Beløbet ændres løbende af Kassereren/Cashier, som kan registrere indbetalinger fra medlemmer.
- ❖ GDPR-behandling
 - General Data Protection Regulation, også kaldet persondataforordningen, er en EU-lov der gælder i alle lande der er medlem af EU. Loven indbefatter blandt andet at besøgende på hjemmesider skal informeres om hvad deres data bliver brugt til og hvor meget de vil dele.

Use cases

Fully Dressed use case I - Indregistrering af nyt medlem

Scope: Medlemsoverblik

Level: User goal

Primary actor: Formanden(backend - læs: Chairman)

Stakeholders and interests:

- Formanden: Ønsker en simpel måde at registrere og administrere klubmedlemmer i Svømmeklubben Delfinens medlemsoverblik.
- Potentielle klubmedlemmer: Ønsker at deres data bliver lagret bag et login-system hvilket resulterer i sikkerhed og bidrager til at mindske frygt for menneskelige fejl.
- Kasserer: Ønsker automatisk at modtage kontingentpris og faktureringsoplysninger om nye medlemmer, baseret på alder og aktivitet.
- Svømmetræner: Ønsker altid at kunne danne sig et overblik over indmeldte medlemmer, med fokus på de aktive, ifm. registrering og overblik over svømmernes resultater.
- Klubejer: Ønsker at have mulighed for nemt at overdrage svømmeklubben og dens medlemmer til eventuelle købere. Digitaliseret medlemsadministration giver svømmeklubben mere værdi.

Preconditions: Formanden er tilstede og er i selskab med et nyt medlem.

Success Guarantee (or Postconditions): Medlemmet gemmes i medlemsoverblikket ud fra alder. Medlemmets faktureringsoplysninger og kontingentpris registreres i systemet automatisk og medlemmet registreres under svømmetræneren, såfremt dette er konkurrerende.

Main Success Scenario (or Basic Flow):

1. Formanden tilgår medlemsoverblikket
2. Formanden indtaster sine login-oplysninger korrekt
3. Formanden vælger muligheden for at indregistrere et nyt medlem
4. Systemet efterspørger relevante oplysninger (navn, adresse, fødselsdato, mail, osv...) og formanden modtager disse fra det nye medlem

5. Det nye medlem registreres i systemet og formanden modtager en bekræftelse på oprettelsen samt et medlemsnummer, som videregives til det nye medlem.

Extension (or Alternative Flows):

1. Formanden glemmer login-oplysninger
 - a. Formanden kontakter udviklerne af systemet
 - i. Systemudviklerne tilgår koden og supplerer denne information til formanden
 - ii. Systemudviklerne tilgår koden og opretter en ny adgangskode og/eller et nyt brugernavn.
2. Formanden vælger forkert undermenu ved indregistrering.
 - a. Formanden benytter muligheden for at forlade nuværende menu og vender tilbage til hovedmenuen
 - b. Formanden lukker programmet og genåbner
3. Formanden indtaster fejlagtige oplysninger ved indregistrering af medlem
 - a. Formanden tilgår den eksterne txt-fil som er lagret på computeren efter oprettelse og ændrer de fejlagtige oplysninger.
 - b. Formanden afbryder oprettelsen og lukker programmet, for herefter at genåbne og starte oprettelsen på ny.
4. Formanden glemmer at overdrage medlemsnummeret til det nye medlem.
 - a. Kunden efterspørger medlemsnummer med det samme.
 - i. Formanden kigger i indtastningshistorikken og finder frem til medlemsnummeret.
 - b. Kunden efterspørger medlemsnummeret på et senere tidspunkt, hvor programmet har været genstartet og kundens medlemsnummer indgår ikke længere i indtastningshistorikken.
 - i. Formanden benytter muligheden for at få en oversigt over alle medlemmer samt deres medlemsnummer og supplerer relevante til medlemmet.

Special requirements:

- Kunden skal acceptere vilkår for at være medlem af svømmeklubben.

- Kunden skal betale kontingent forud for at indmelde sig i svømmeklubben.
- GDPR-behandling af personlig data
- En computer er nødvendig for brug af systemet

Technology and Data Variations List:

5.* Lokal lagring af medlems- og resultatlister (.txt-filer).

4.* Input via konsol.

Frequency of occurrence: Viger enormt meget, men vil formentlig have højdepunkter i højsæsoner.

Asterisk() henviser til trin i Main Succes Scenario*

Fully Dressed use case II - Tjek restance for medlem

Scope: Medlemsoverblik

Level: User goal

Primary actor: Kasser(backend - læs: Cashier)

Stakeholders and interests:

- Potentielle klubmedlemmer: Ønsker at deres data bliver lagret bag et login-system hvilket resulterer i sikkerhed og bidrager til at mindske frygt for menneskelige fejl.
- Kasserer: Ønsker, nemt og enkelt, at kunne registrere og danne sig overblik over kunders mellemværende.
- Klubejer: Ønsker at der digitalt bliver registreret kontingentoplysninger, for at mindske menneskelige fejl og for at kunne overdrage disse informationer til potentielle købere af klubben. Det sikrer værdi.

Preconditions: Kasserer er tilstede og har modtaget indbetaling fra et medlem, samt har deres medlemsnummer.

Success Guarantee (or Postconditions): Kasser og eventuel tredjepart informeres om givent medlems restance..

Main Success Scenario (or Basic Flow):

1. Kasserer tilgår medlemsoverblikket
2. Kasserer indtaster sine login-oplysninger korrekt
3. Kasserer vælger at tjekke beløb i restance for klubmedlem
4. Systemet efterspørger medlemsnummer på klubmedlemmet, hvis restance skal tjekkes
5. Kasserer indtaster klubmedlemmets medlemsnummer
6. Systemet returnerer beløb i restance for medlem, ud fra medlemsnummer.

Extension (or Alternative Flows):

1. Kasserer glemmer login-oplysninger
 - a. Kasserer kontakter udviklerne af systemet
 - i. Systemudviklerne tilgår koden og supplerer denne information til kasserer

- ii. Systemudviklerne tilgår koden og opretter en ny adgangskode og/eller et nyt brugernavn.
2. Kasseren vælger forkert undermenu, ved valg heraf.
 - a. Kasseren benytter muligheden for at forlade nuværende menu og vender tilbage til hovedmenuen
 - b. Kasseren lukker medlemsoverblikket og genåbner
3. Kasseren indtaster forkert medlemsnummer
 - a. Kasseren vender retur til menuen
 - i. Kasseren gentager trin 3-6*
 - b. Kasseren lukker medlemsoverblikket i afmagt
 - i. Kasseren gentager trin 1-6*

Special requirements:

- Kunden skal acceptere vilkår for at være medlem af svømmeklubben.
- Kunden skal betale kontingent forud for at indmelde sig i svømmeklubben.
- GDPR-behandling af personlig data
- En computer er nødvendig for brug af systemet

Technology and Data Variations List:

- 6.* Lokal læsning af medlemslister (.txt-filer).
- 5.* Input via konsol.

Frequency of occurrence: I starten af sæsonen, august måned, med forbehold for eftertilmeldinger samt sene betalinger; september, senest oktober.

Asterisk() henviser til trin i Main Succes Scenario*

Fully Dressed use case III - Registrer konkurrenceresultat i butterfly for seniorsvømmer

Scope: Medlemsoverblik

Level: User goal

Primary actor: Svømmetræner (backend - læs: Coach)

Stakeholders and interests:

- Potentielle klubmedlemmer: Ønsker at deres data bliver lagret bag et login-system hvilket resulterer i sikkerhed og bidrager til at mindske frygt for menneskelige fejl.
- Coach: Ønsker, nemt og enkelt, at kunne registrere svømmeklubbens medlemmers konkurrenceresultater, samt danne sig overblik over disse.

Preconditions: Svømmetræneren er tilstede og har modtaget informationer om konkurrenceresultat fra et medlem.

Success Guarantee (or Postconditions): Medlemmets svømmeresultat registreres i medlemsoverblikket.

Main Success Scenario (or Basic Flow):

1. Svømmetræneren tilgår medlemsoverblikket
2. Svømmetræneren logger ind med korrekte login-oplysninger
3. Svømmetræneren vælger muligheden for at registrere et resultat for klubmedlem
 - a. Vælger registrér en konkurrencetid
 - b. Vælger registrér for seniorsvømmer
 - c. Vælger at indtaste et medlemsnummer
4. Svømmetræneren indtaster rette medlems medlemsnummer
5. Systemet efterspørger relevante oplysninger (navn på stævne, disciplintype, tidsresultat, placering) og svømmetræneren modtager disse fra det medlemmet.
6. Systemet lagrer resultatet i CompetitionTimes
7. Svømmetræneren vælger nej, når denne bliver spurgt om de vil registrere yderligere tider

Extension (or Alternative Flows):

1. Svømmetræneren glemmer login-oplysninger
 - a. Kasseren kontakter udviklerne af systemet

- i. Systemudviklerne tilgår koden og supplerer denne information til kasseren
 - ii. Systemudviklerne tilgår koden og opretter en ny adgangskode og/eller et nyt brugernavn.
2. Svømmetræneren vælger forkert undermenu ved trin 3, 3a, 3b eller 3c*.
 - a. Svømmetræneren lukker programmet
 - i. Svømmetræneren gentager trin 1-3*.
3. Svømmetræneren mangler medlemsnummer for medlem
 - a. Svømmetræneren vælger muligheden for at få overblik over medlemmers navne og ID'er (Trin 3*).
 - i. Svømmetræneren finder frem til relevant ID og fortsætter fra trin 4*.

Special requirements:

- Kunden skal acceptere vilkår for at være medlem af svømmeklubben.
- Kunden skal betale kontingent forud for at indmelde sig i svømmeklubben.
- GDPR-behandling af personlig data
- En computer er nødvendig for brug af systemet

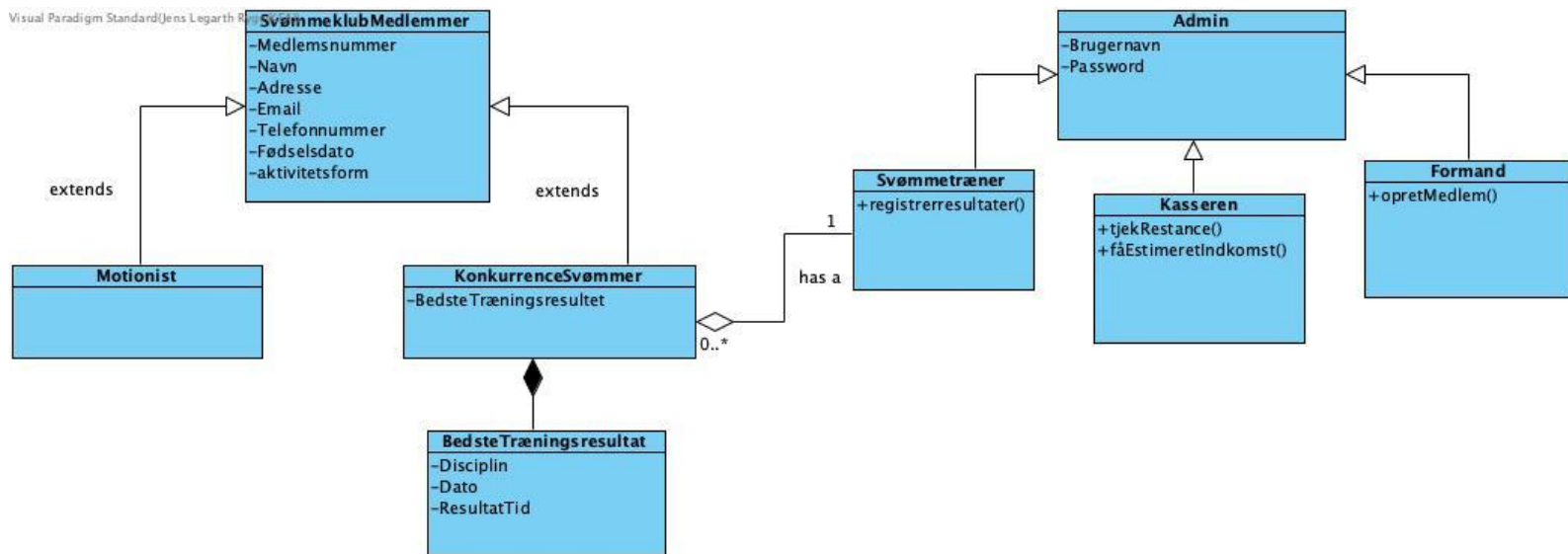
Technology and Data Variations List:

6.* Lokal lagring af medlems- og resultatlister (.txt-filer).

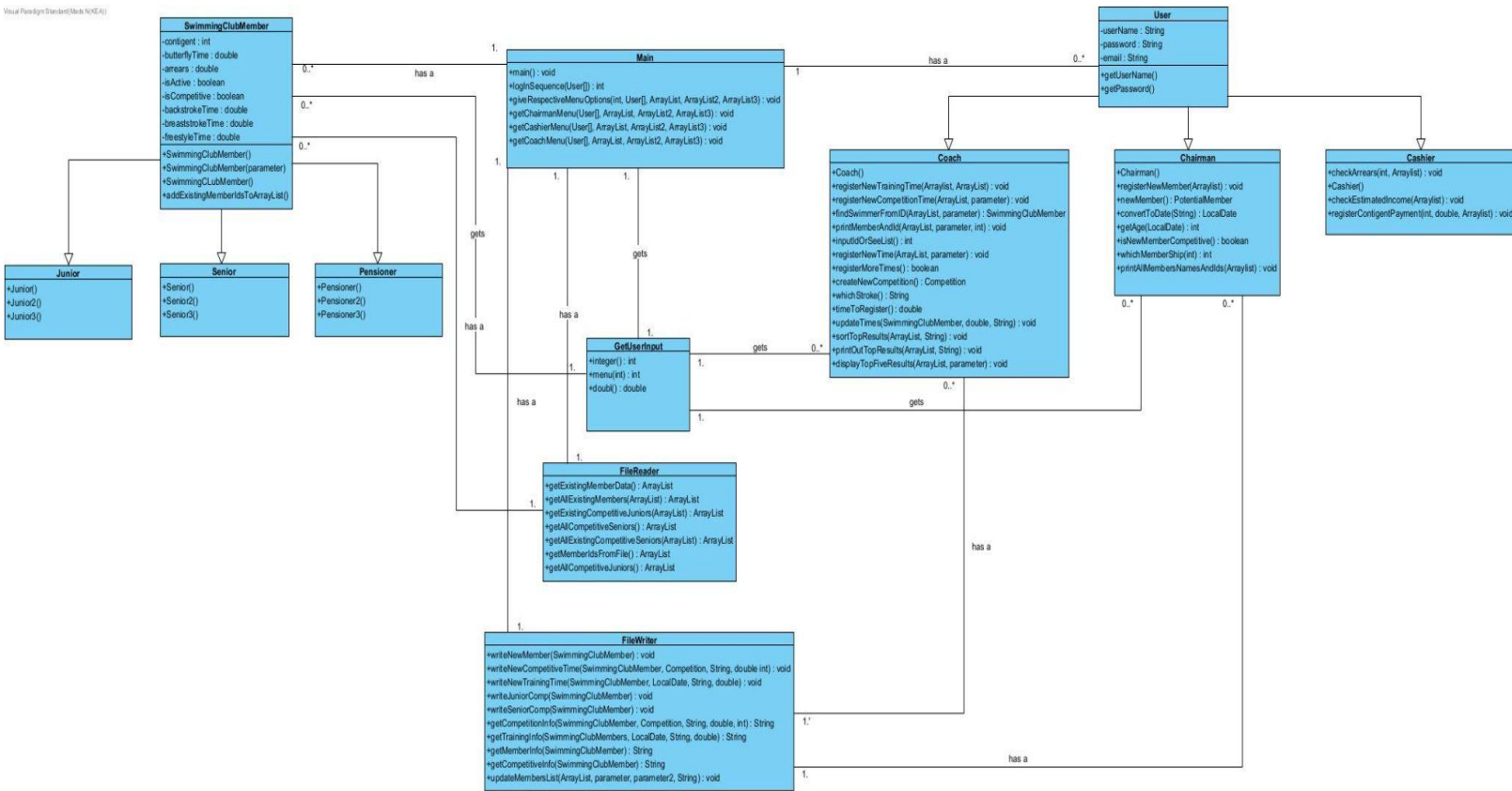
3. -6 . & 7.* Input via konsol.

Frequency of occurrence: Sker efter hvert svømmestævne, hvilket kan varierer fra hver weekend til en gang om måneden.

Asterisk() henviser til trin i Main Succes Scenario*

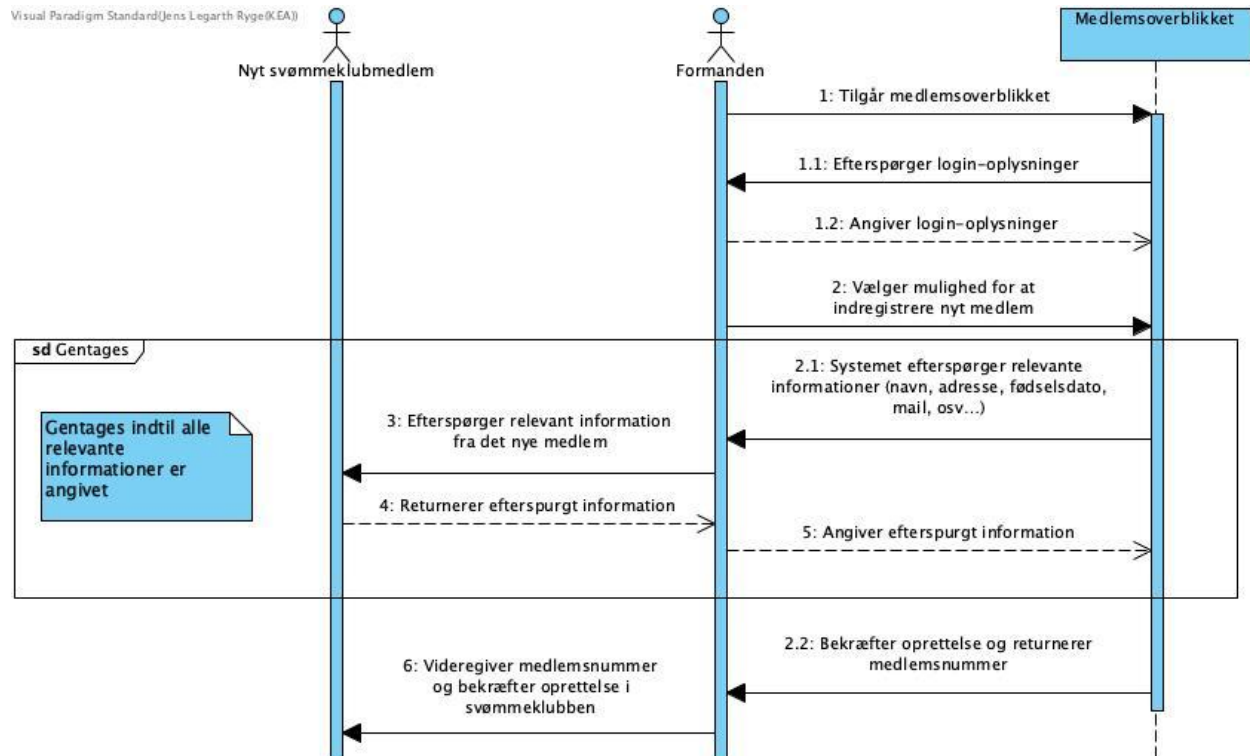
Domain Model

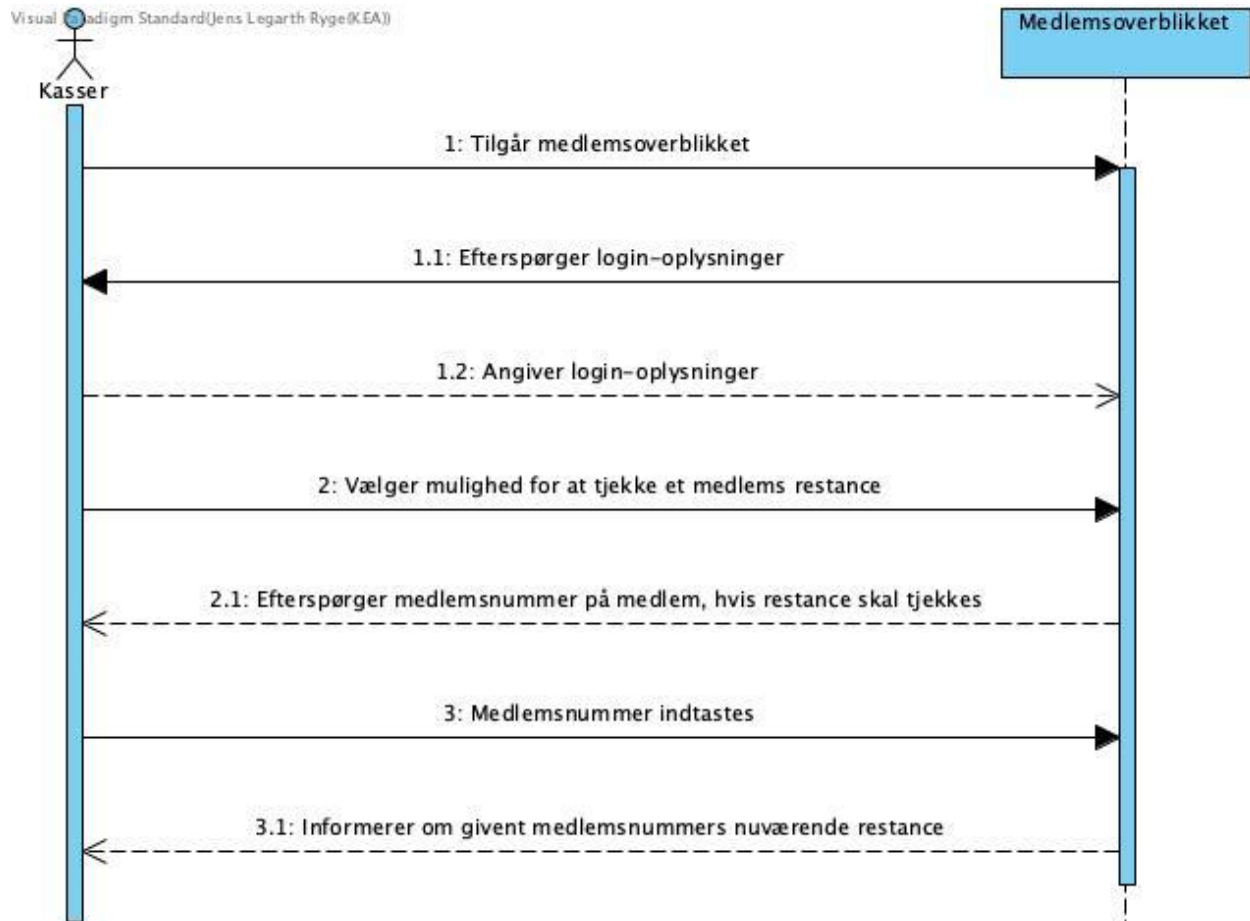
Class Diagram

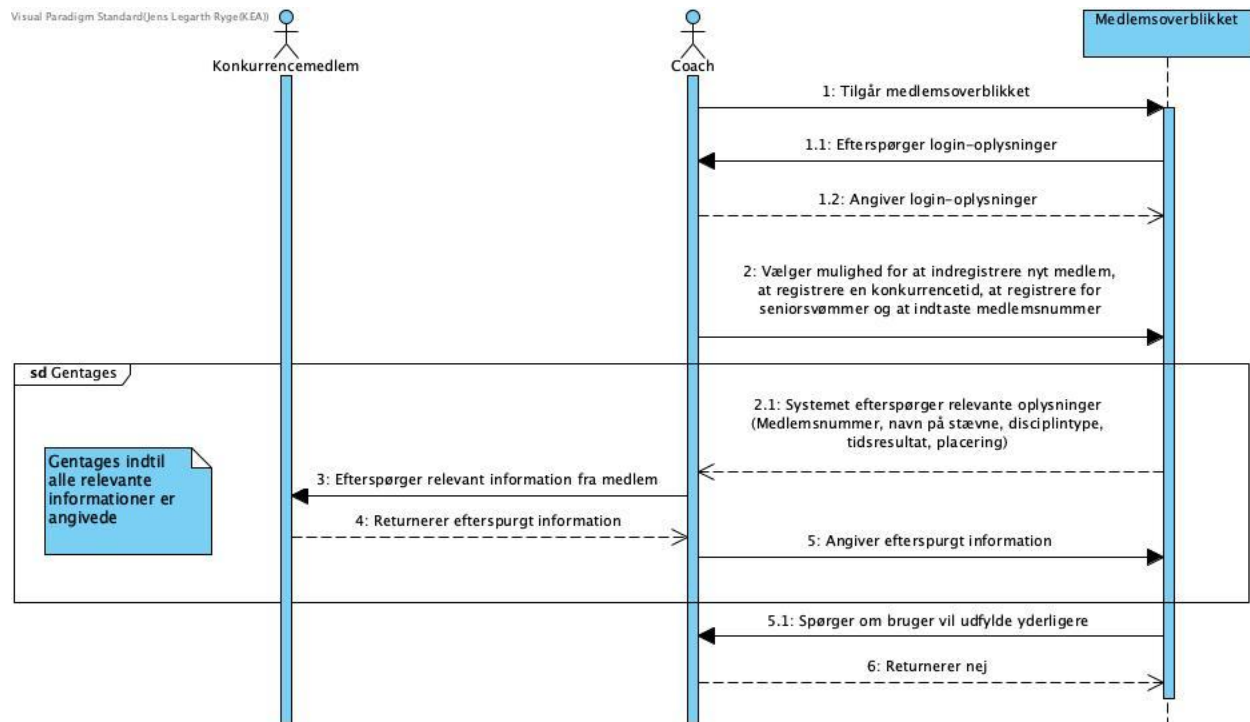


System Sequence Diagrammer

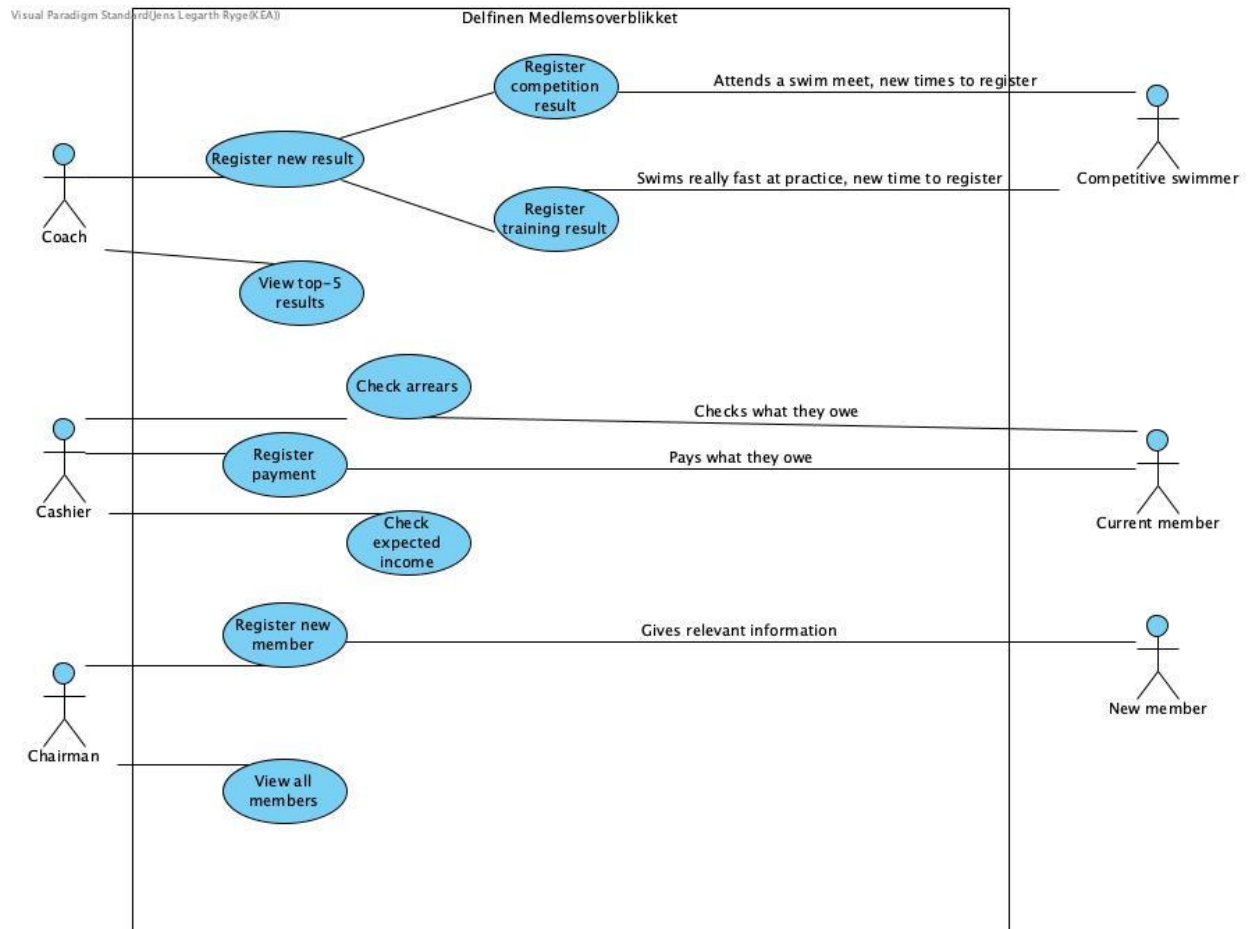
SSD I - Indregistrering af nyt medlem



SSD II - Tjek restance for medlem

SSD III - Registrer konkurrenceresultat i butterfly for seniorsvømmere

Use Case Diagram



Overvejelser ift. GRASP

Information Expert

- I forhold til information expert har vi tænkt over hvilke klasser der har den relevante information og funktionaliteten til at bearbejde det. Vi er blevet begrænset af opgavebeskrivelsen i den forstand at specifikke klasser er beskrevet at skulle have en metode som ville passe bedre i andre klasser. Et eksempel på dette kunne være: `printOutTopResult`, `createNewCompetition` osv. Disse metoder ville passe ind i andre klasser men det er beskrevet at de skal være i `coach`

Creator

- Vores overvejelser i forhold til creator er at de klasser som har informationen står for oprettelsen af vores objekter. Dog med forbehold for hvad vi har antaget ud fra opgavebeskrivelsen.
- Fx; Formanden/Chairman havde til opgave at oprette nye medlemmer, og derfor har vi sat al funktionalitet og informationsindsamling om de nye medlemmer ind i den klasse og klassens metoder. Derudover slutes metoden af oprettelse af et nyt medlems-objekt af med at vi både sender den ind i vores arraylist over "allMembers" og at medlemmet bliver skrevet ind i vores fil.

Low coupling

- I forhold til low coupling har vi prøvet at få vores classes til at være så uafhængige af hinanden som muligt. Dog er de fleste af vores classes afhængige af vores `FileReader` og `FileWriter` class, men det er ikke noget vi kunne komme uden om.
- Hvis man ændre noget inde i vores `Chairman` class vil ikke have nogen effekt på vores `Coach` class osv.

High cohesion

- Vi har prøvet så vidt muligt at holde vores classes simple men der er dog nogle tilfælde hvor at klassen er svær at forstå som feks. `Coach` klassen. Den er forholdsvis uoverskuelig, grundet en overflod af små metoder, men vi lader den være som den er fordi vi, i forhold til information expert, har valgt at holde al den relevante information i klassen.

Polymorphism

- Har vi for eksempel brugt til vores medlemstyper, med at inddele dem i flere konstruktors så vi kunne gemme den information vi gerne ville, når vi skulle bruge dem til forskellige ting. Fx når vi skal oprette tider til konkurrence svømmerne og når vi skal “downloade” dem fra vores fil til arraylister når vi starter vores program op.

Afgrænsninger og antagelser

- Konkurrencesvømmere bliver "bedømt" i tid; det er hvad vi tracker dem på.
- Back-end udvikles på engelsk for at kunne ramme brede i tilfælde af yderligere rekruttering, front-end kommer til at være på dansk.
- Vi snakkede kort om passivt og aktivt medlemskab og dets relevans i backend-arbejdet. Vi vurderer at kun kontingent prisen kun har relevans vedr. Medlemmets status, som værende enten aktiv- eller passivt medlem; resterende detaljer er out-of-scope.
- Der er blevet overvejet hvorvidt en login metode vil forbedre systemet
- ~~Formand behøver ikke adgang til medlemsliste, men kun kunne oprette nye medlemmer i systemet.~~
- Der er overvejet om der skal være disciplinattributter til svømmerne (som crawl, bryst osv.).
- ~~Skal der være en funktion til at afmelde/slette medlemmer i systemet?~~ Det giver logisk mening, men står ikke til at være en del af kravspecifikationerne.
- Tiderne som konkurrencesvømmerne svømmer, bliver tracket i minut.sekund (1.52 fx). Derfor vi bruger double til at tracke dem. (Hvis vi vil have 100 dele af sekunder med, kunne man så bruge float?)
- Vi tracker kun en enkelt distance i hver disciplin lige pt. - Her tænker vi 100 meter.
- Programmet kører hele tiden lige pt. Når man logger ud fra den relevante funktion der har været i brug, så kommer man tilbage til log in. Simulere at programmet kører så længe computeren er tændt.