

SVØMMEKLUBBEN DELFINEN



DATO

28-05-2024

SEMESTER & ÅRGANG

1 semester - DAT24v3

VEJLEDERE

Lasse Vogelsang

James Hindsgavl Brink

STUDERENDE

Gruppe F:

Caroline Brandt

Solvej Todd

Jasmin Al Aqtash

Indholdsfortegnelse

Produkt Vision	2
Liste med alle krav	2
FURPS	3
Domain Model	3
Glossary	4
Use Case Diagram	4
User Stories	5
Definition of Done	5
Product & Sprint Backlog	6
3 sequence diagrams	9
Class diagram	10
Test Cases	10

Produkt Vision

- “ Svømmeklubben delfinen ønsker at facilitere svømmeoplevelser på alle niveauer for alle aldre “.

Liste med alle krav

Administrativt system:

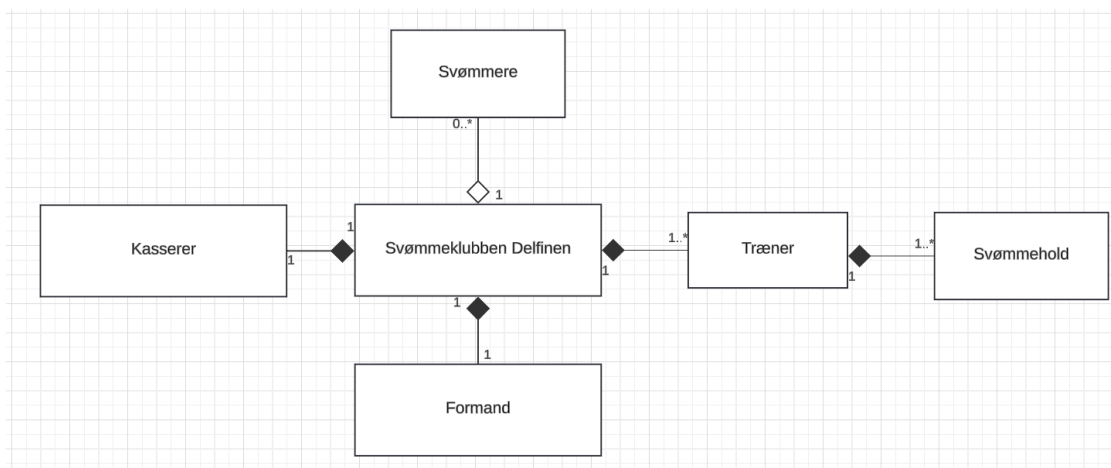
- Nye medlemmer
 - o Stamoplysninger, herunder alder
 - o Ønskede aktivitetsform – aktivt eller passivt medlemskab
 - o Junior – seniorsvømmer
 - o Motionist – eller konkurrencesvømmer
 - o Passivt medlemskab – for 500 kr.
 - § Ingen adgang til svømmehold med motionister og konkurrencesvømmere.
 - § Kan købes adgang til svømmehallen for et mindre beløb.
- Klubbens kasserer:
 - o Kontigent betaling
 - o Medlemsskaber:
 - § Aktive medlemmer under 18 år (Junior) – 1000 årligt
 - § Seniorsvømmere (18 år og over) 1600 årligt.
 - § Over 60 år gives der 25 % rabat på ‘Oversixty’.
 - § Passivt medlemskab – 500 årligt.
 - § Vis oversigt over medlemmer der er i restance.
 - o Konkurrencesvømmerne
 - § Tilknyttet en træner
 - § Inddelt i 2 hold efter alder
 - § Ungdomsholdet < 18 år
 - § Seniorholdet > 18 år
 - § Konkurrencesvømmer inddeles efter hvilke svømmediscipliner medlemmet er aktiv i.
 - § Den enkelte svømmer, der deltager i konkurrencer, registreres stævne, placering og tid.
 - § For hver svømmedisciplin registreres den enkelte svømmers bedste træningsresultat og dato løbende.
 - § Oversigt over klubbens top 5 svømmere inden for hver svømmedisciplin.

FURPS

Functionality, usability, reliability, performance, supportability

F	<ul style="list-style-type: none"> Registrering af enkelte svømmers bedste træningsresultat inden for hver svømmedisciplin Spor bedste træningsresultater og lav en liste med de bedste <p>Formand:</p> <ul style="list-style-type: none"> Registrering af nye medlemmer: alder, aktivitetsform (passivt/aktivt medlemskab), motionist eller konkurrence Tildele medlemmer hold <p>Kasserer:</p> <ul style="list-style-type: none"> Administrering af kontingentbetaling på baggrund af alder Se en oversigt over medlemmer der er i restance Lave faktura <p>Træner:</p> <ul style="list-style-type: none"> Se en oversigt over top 5 svømmere i hver disciplin så de kan udtages til konkurrence <p>Konkurrencesvømmer:</p> <ul style="list-style-type: none"> Ved deltagelse i konkurrencer registreres stævne, placering og tid Inddeles i hold på baggrund af: alder
U	<p>Brugerfladen skal være nem at bruge så:</p> <ul style="list-style-type: none"> formanden nemt kan oprette nye medlemmer træneren har læsbar og overskuelig oversigt over de bedste svømmere kasserer har et overblik over kontingentbetaling
R	
P	Skal ikke bruge ekstrem mange minutter for at gennemføre en simpel task
S	Kan vokse til en større platform

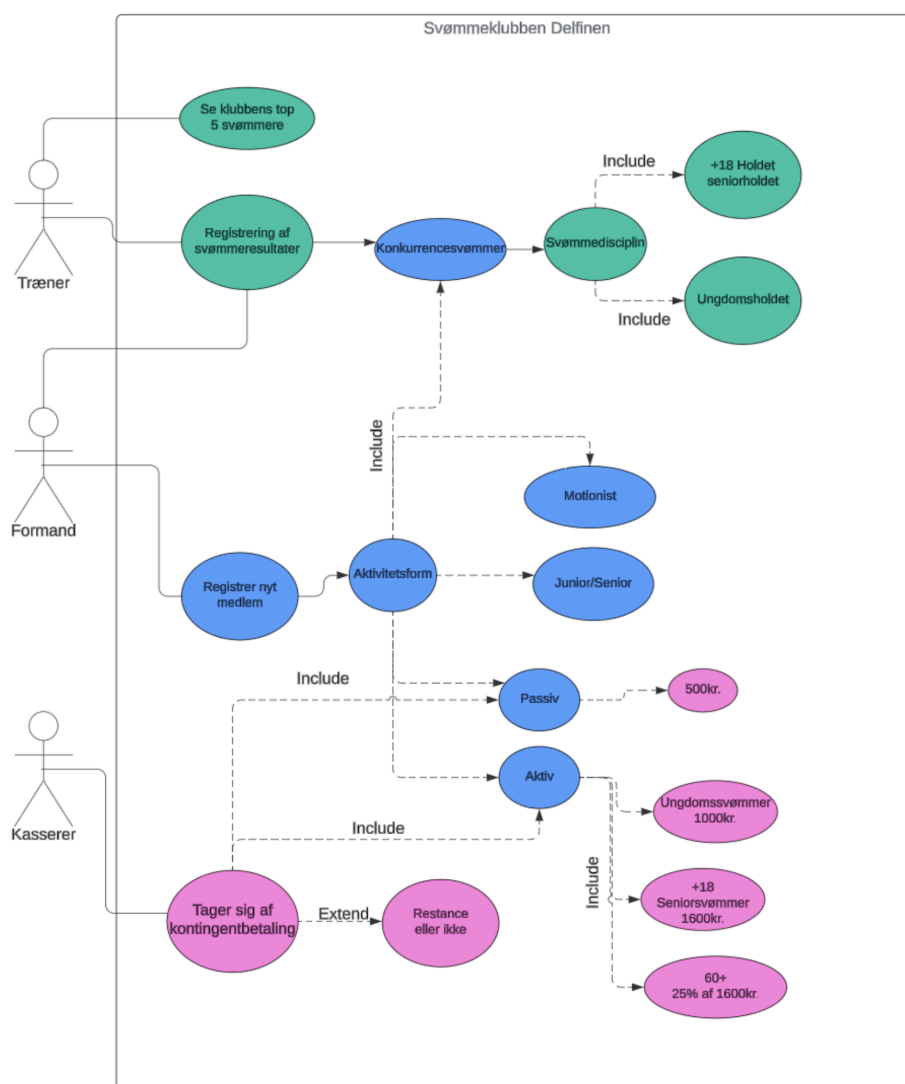
Domain Model



Glossary

	Betydning
Disciplin	De forskellige former for konkurrence svømning i klubben.
Junior, Senior, over sixty	De forskellige aldersgrupper der har forskellige kontingentbetaling
Passiv / aktiv	Enten er medlemmer aktive i klubben eller har et passiv medlemskab, dvs. de for 500 kroner beholder deres plads i svømmeklubben.
Kontingent	Medlemsafgift
Restance	Gæld, når medlemmer ikke har betalt inden fristen.
Junior / Senior	Konkurrencehold

Use Case Diagram



User Stories

Formand	Kasserer	Træner
Som klubbens formand - vil jeg - kunne oprette nye medlemmer med stamoplysninger, herunder alder - for at sikre mig at medlemmer er tilknyttet systemet.	Som kasserer – jeg vil kunne oprette en faktura ift. med oprettelse af nye medlemmer, men også kunne søge efter medlemmer for at se hvis de har betalt. Jeg vil kunne se alle dem i restance – så jeg kan sikre mig at svømmeforeningen har en solid økonomi.	Som træner – vil jeg gerne se mine svømmers resultater – så jeg kan udtage svømmere til deltagelse i konkurrencer.
Som klubbens formand , vil jeg kunne tilføje den ønskede aktivitetsform, junior- eller seniorsvømmer eller om det er passivt eller aktivt - for at – jeg kan få et overblik over vores forening og vores medlemmer.	Som kasserer – vil jeg også kunne se medlemmer og deres medlemskab så jeg sikre mig at de betaler det rigtige beløb.	Som træner – vil jeg gerne registrere træningsresultater for mine elever - så jeg kan se en liste over top 5 indenfor hver disciplin.

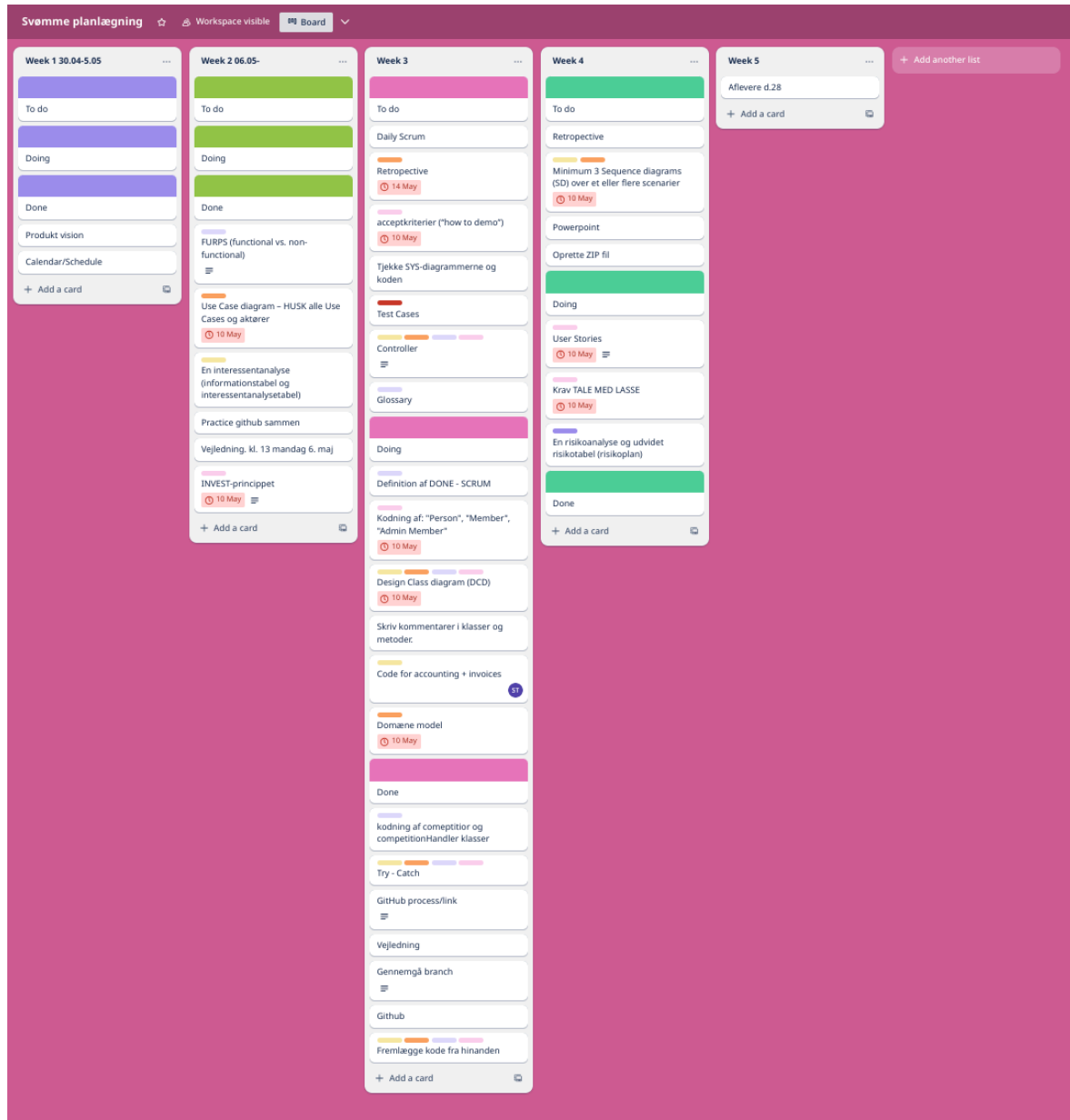
Definition of Done

Vores opgave antages som at være færdig når man kan:

- Oprette et medlem i klubben ved at kunne registrere gennem vores console samt at programmet kan udregne de udvalgte faktorer, som vi har valgt at lægge til hvert enkelt medlem. Hermed lægger medlem klassen en god grund for resten af de informationer som medlemmet får knyttet til sin profil fra de andre klasser.
- Oprette faktura for registreret medlem og teste, om de medlemmer der ikke har betalt automatisk ligger i en restance liste.
- Registrere konkurrencesvømmers resultater fra deres træning og stævner de har deltaget i.
- Se en liste over top 5 svømmere inden for hver svømmedisciplin (og inden for både junior og senior)

Product & Sprint Backlog

Uge 1/2: Fokus var på systemudvikling.



Uge 2/3: Fokus var programmering. Klassediagrammet var opdelt mellem teamet.

Svømme planlægning ☆ Workspace visible Board

Week 1 30.04-5.05

- To do
- Doing
- Done
- Produkt vision
- Calendar/Schedule
- + Add a card

Week 2 06.05-

- To do
- Daily Scrum
- Doing
- FURPS (functional vs. non-functional)
- Glossary
- Definition af DONE - SCRUM
- acceptkriterier ("how to demo") 10 May
- Krav TALE MED LASSE 10 May
- INVEST-princippet 10 May
- User Stories 10 May
- Kodning af: "Person", "Member", "Registre Member" 10 May
- kodning af competition og competitionHandler klasser
- Domæne model 10 May
- Use Case diagram - HUSK alle Use Cases og aktører 10 May
- Design Class diagram (DCD) 10 May
- Minimum 3 Sequence diagrams (SD) over et eller flere scenarier 10 May
- En interessentanalyse (informationstabel og interessentanalysetabel)
- Done
- Practice github sammen
- Vejledning, kl. 13 mandag 6. maj
- + Add a card

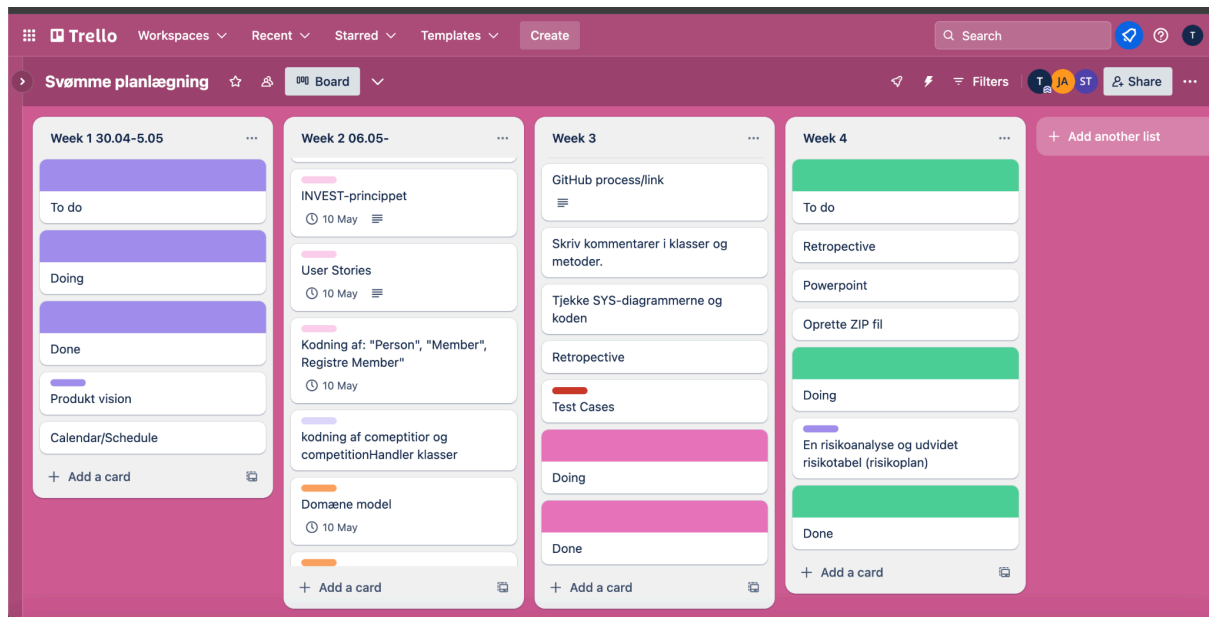
Week 3

- To do
- Retropective 14 May
- Vejledning
- Gennemgå branch
- Github
- Github process/link
- Skriv kommentarer i klasser og metoder
- Tjekke SYS-diagrammerne og koden
- Retropective
- Test Cases
- Doing
- Done
- + Add a card

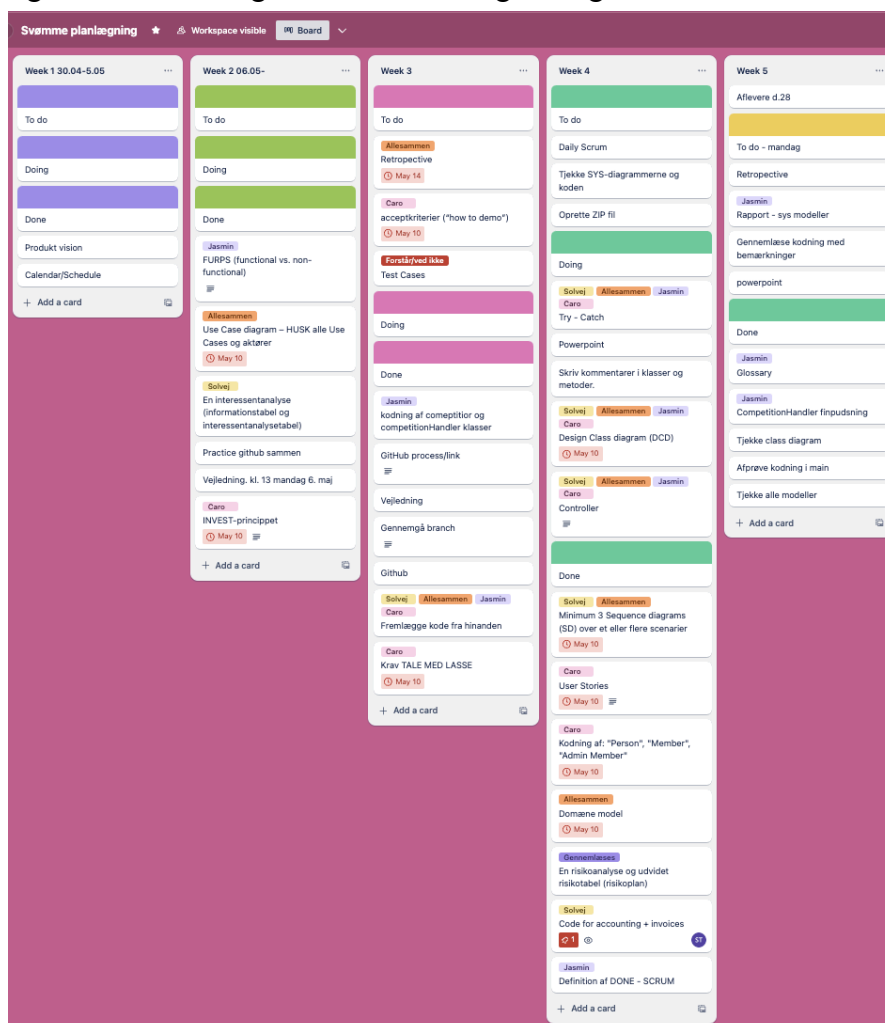
Week 4

- To do
- Retropective
- Powerpoint
- Oprette ZIP fil
- Doing
- En risikoanalyse og udvidet risikotabel (risikoplan)
- Done
- + Add a card

Uge 3: Fokus var på forbedring af kode. Vi indså at vi måtte droppe inheritance og bruge enum, der medførte et spring tilbage mht. ændringer i koden.



Uge 4: Omskrivning/rettelse af kode og testing.

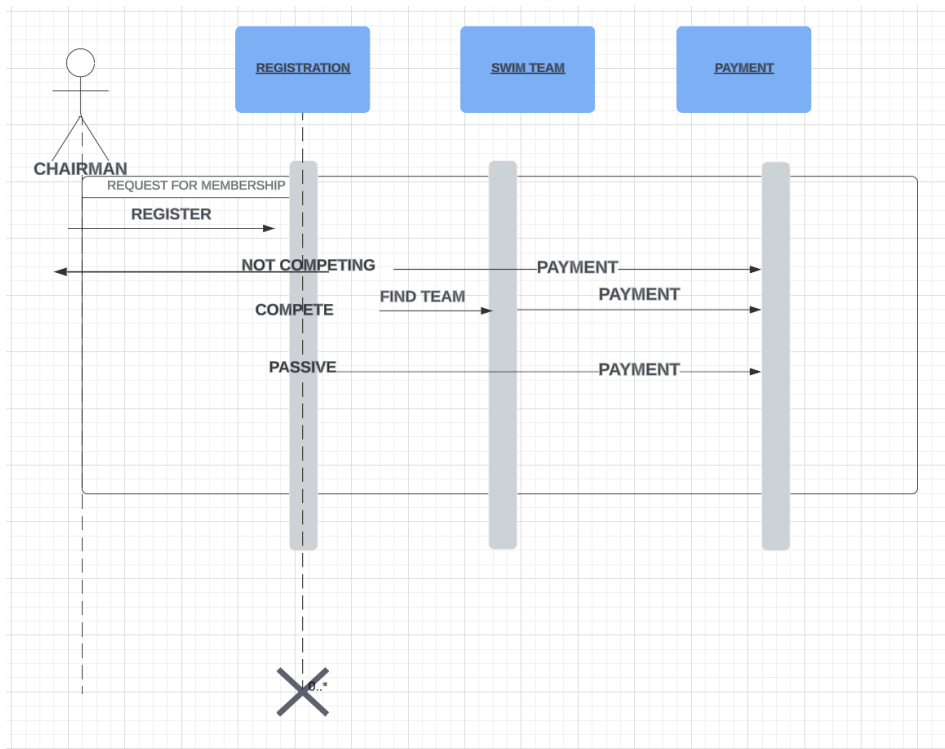


Vi har lært vigtigheden af forarbejdet der laves i systemudvikling ift. programming og hvor vigtigt det er at have en fælles forståelse af hvad systemet skal kunne gøre til sidst.

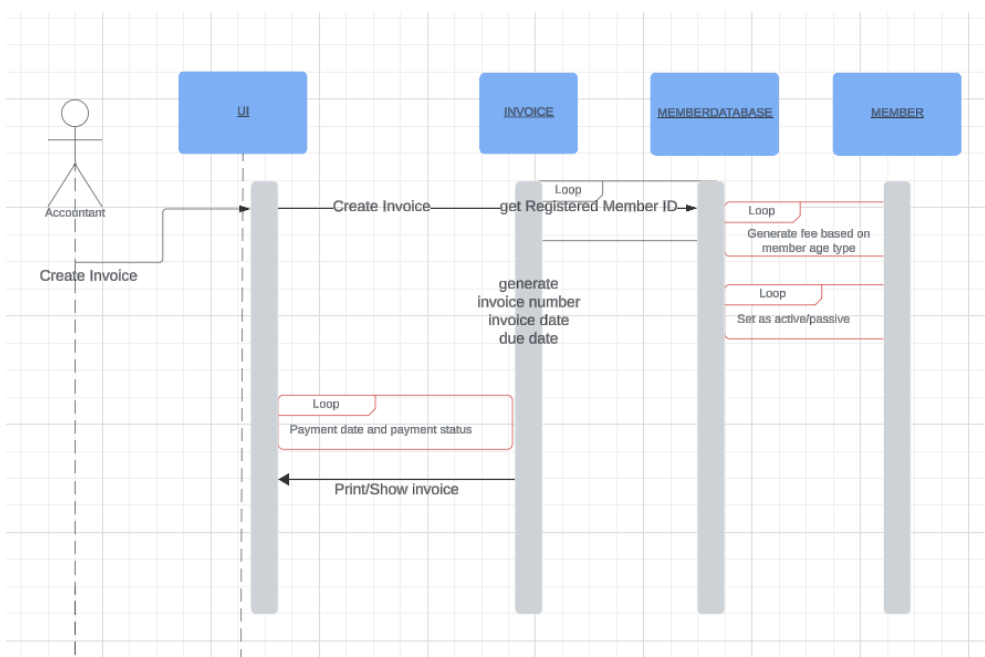
Vi indser at sprint processen egentlig har mere fokus på at færdiggøre enkelte dele og komme videre, hvorpå vores proces har til dels været mere blandet, hvor vi er begyndt på flere dele samtidig.

3 sequence diagrams

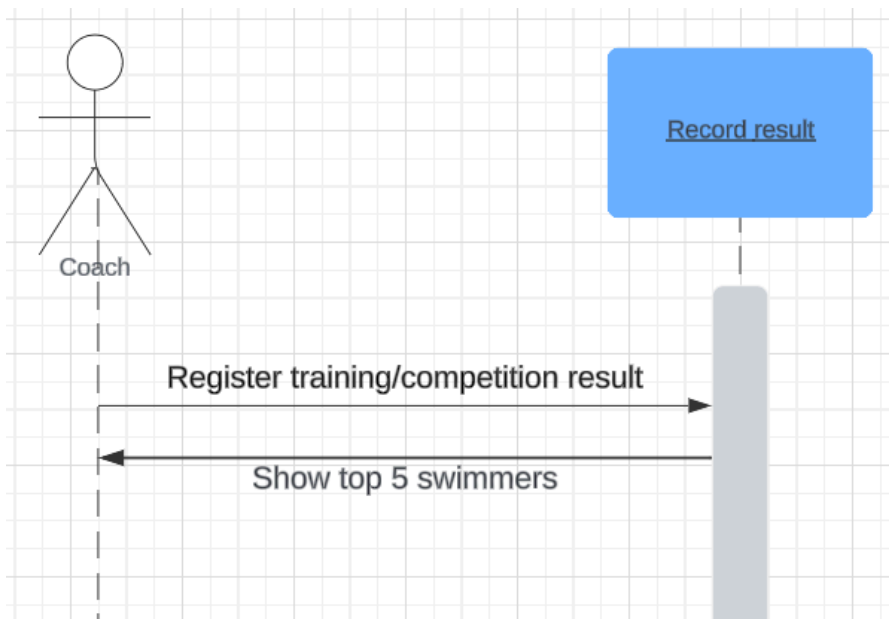
Chairman



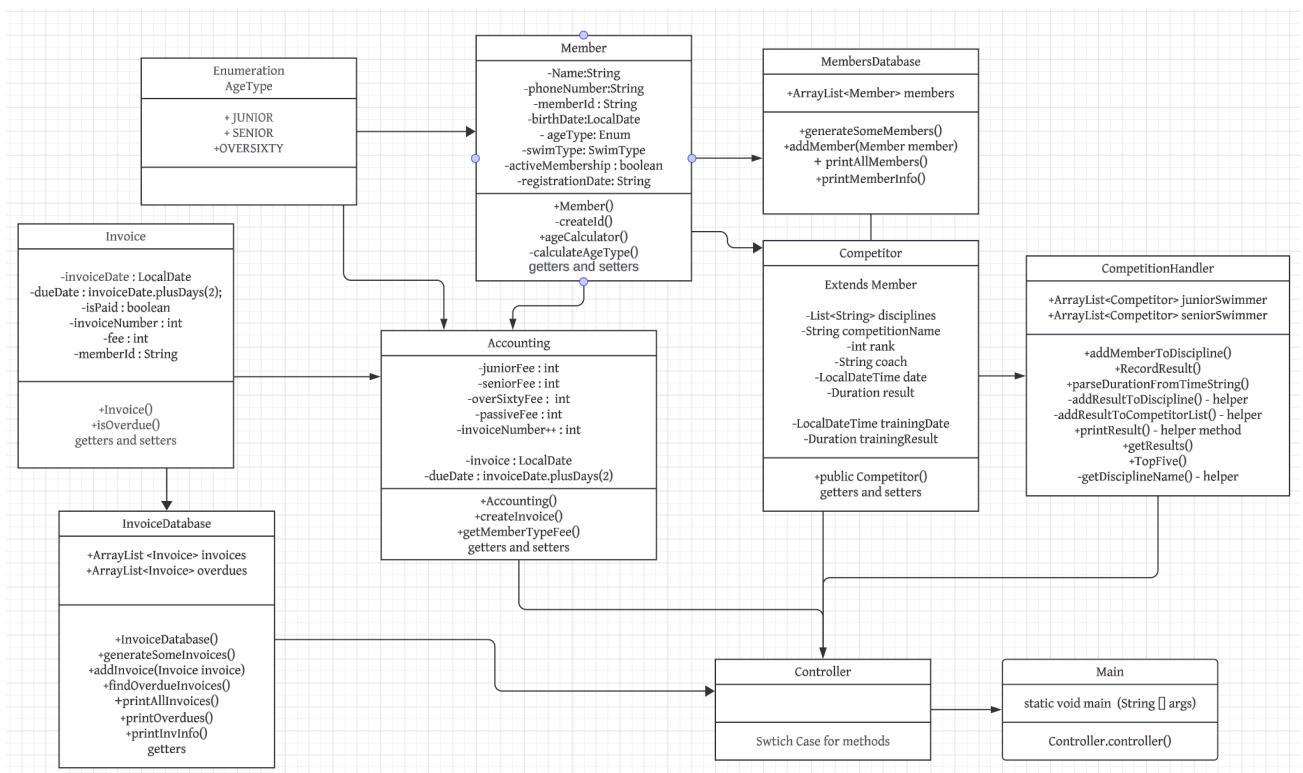
Accountant



Coach



Class diagram



Test Cases

Vi har udført flere test på programmet, men her er nedskrevet to eksempler på dette:

Test 1.

Test Case ID: BirthDate/fødselsdag

Test Scenario: Brugen af Svømmeklubben Delfinen skal kunne taste forkert på tastaturet uden at programmet lukker ned. For lokalisere to succesfulde brugeroplevelser med programmet, så skal følgende ske.

Test trin:

Test data:

Brugerne taster indtastning af fødselsdag.

- 1.Brugeren skal kunne ramme forkert på tastaturet uden at programmet stopper.
2. Hvis man taster forkert, så skal man ikke starte forfra med at oprette et medlem påny.
3. Hvis man taster forkert, så skal programmet bede brugeren om at skrive fødselsdag igen.

Ved succesfuld fødselsdagsinformation i det rigtige format, så skal programmet gå videre til næste spørgsmål ved oprettelse af medlem.

Prerequisites: Registrering af fødseldagsinformation med tastefejl, men uden at programmet stopper og med guide om, at man skal prøve igen.

Browser: JGRASP

Test Status – Pass/Fail: Pass

Brugeren kan taste fødselsdag forkert uden programmeret stopper.

Test Data: Godkendt birthdate format.

Expected/Intended Results: Birthdate bliver oprettet og sidenhen brugt til udregning af alderen og begge bliver til sidst printet, uden at programmet er stoppet, men blot dirigeret brugeren til at skrive birthdate igen ved fejltastning.

Actual Results: Det gik som forventet.

Test Status – Pass/Fail: Pass

Test 2:

Test Case ID: Svømmestatus COMPETITIVE/NONCOMPETITIVE

Test Scenario: Brugen af Svømmeklubben Delfinen skal kunne tæste forkert på tastaturet uden at programmet lukker ned. For lokaliserer to succesfulde brugeroplevelser med programmet, så skal følgende ske.

Test trin:

Test data:

Brugerne taster indtastning af svømmestatus.

1. Brugeren skal blive guidet, hvis der bliver skrevet forkert, med en fejlmelding meddelelse. kunne ramme forkert på tastaturet uden at programmet stopper.

2. Brugeren skal blive guidet med en besked, hvis der bliver skrevet med små bogstaver, med en fejlmelding meddelelse.

3. Hvis man taster forkert, så skal man ikke starte forfra med at oprette et medlem på ny, men blot vises tilbage for at tæste informationen korrekt.

Ved succesfuld svømmestatus i det rigtige format, så skal programmet gå videre til næste spørgsmål i oprettelsen af medlem.

Prerequisites: Registrering af svømmestatus med tastefejl, men uden at programmet stopper og med guide til, at man skal prøve igen.

Browser: JGRASP

Test Status – Pass/Fail: Pass

Brugeren kan tæste små bogstaver eller med små bogstaver, men uden programmet stopper.

Test Data: Godkendt svømmestatus format.

Expected/Intended Results: Svømmestatus (Swimtype) bliver oprettet og sidenhen printet, uden at programmet er stoppet, men med god guide til brugeren om, hvor fejlen findes, hvis en sådan skulle ske, og hvad man skal gøre.

Actual Results: Det gik som forventet.

Test Status – Pass/Fail: Pass