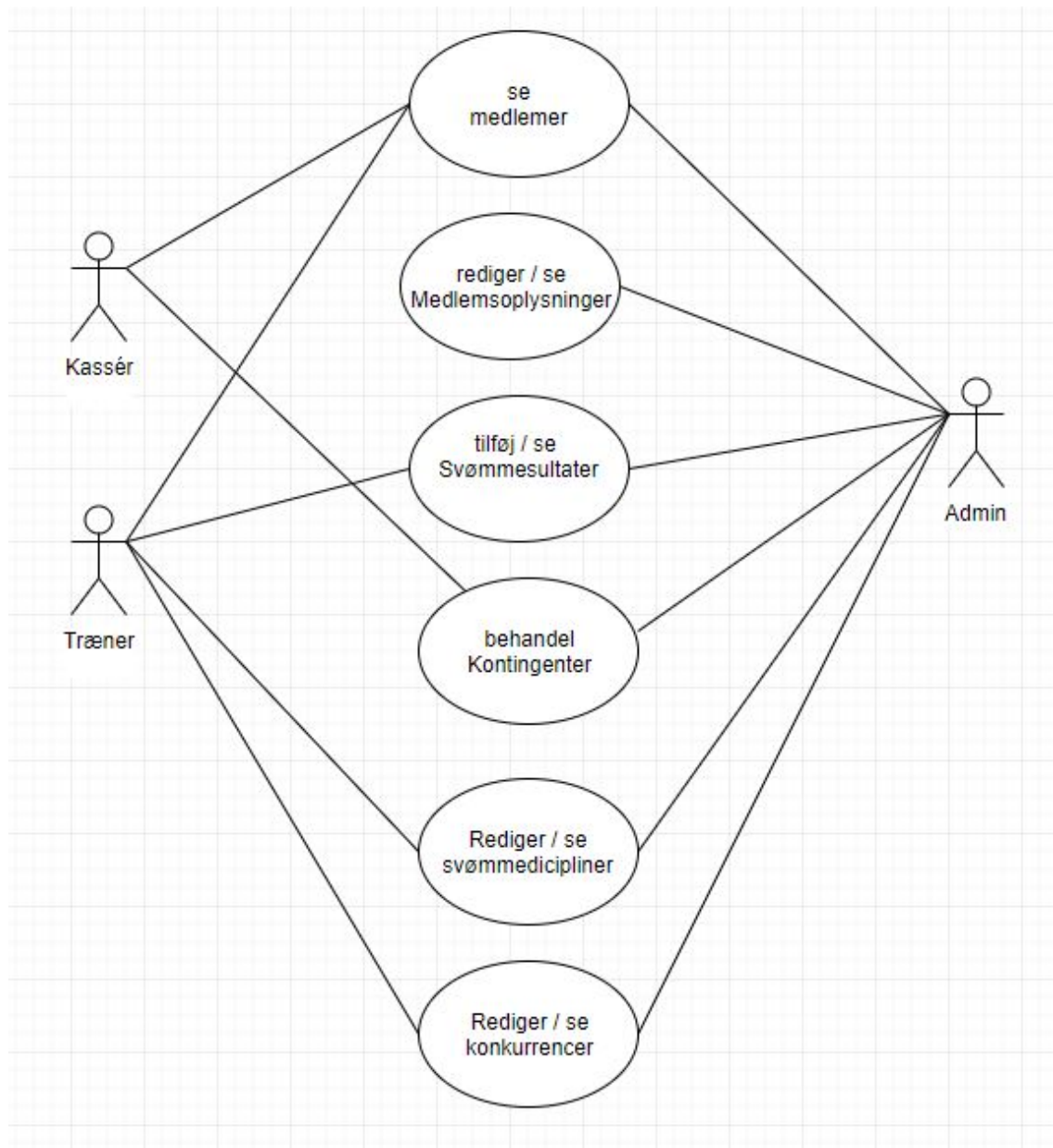


Project Delfinen

Rasmus Hemmingsen, Rasmus Prætorius, Emil Neis, Ditlev Andersen


Implementation af funktionalteter og use cases

Use case diagrammet er en del af det man kalder behavioral UML, og beskriver hvordan brugernes tilgang kunne være til det eventuelle system. Vores use case diagram fra starten af projektet kan ses nedenfor.



Implementationen af use case diagrammet og den tankegang der lå i den kan tydelige ses i den endelige kunde demo hvor adgangsmulighederne for de forskellige brugere er begrænset i relation til deres rolle. Adgangsmulighederne er angivet ved adgangen til de øverste knapper/faner. Dette kan ses på screenshots fra selve den endelige kundedemo nedenfor.

Admin:



Logout

Medlemmer

Svømmersultater

Kontingenter

Svømmediscipliner

Discipliner top 5

Medlemmer

Tilføj ny medlem


Fjern en medlem

Refresh

Database information Medlemmer:

Name	Date of birth	eMail	Disciplin	Status	Konkurrencesvømmer	Payment
Klaus	1989-02-08	klaus@gmail.com	rygcrawl	true	true	true
Stine	1999-08-02	stine@gmail.com	butterfly	true	false	true
Tine	1979-05-02	tine@gmail.com	crawl	false	true	false
Henrik	1989-12-12	henrik@gmail.com	rygcrawl	true	false	true
Anna	1993-04-08	anna@gmail.com	rygcrawl	true	true	true
Freja	1995-02-08	freja@gmail.com	brystsvømning	true	false	false
William	1994-12-27	william@gmail.com	rygcrawl	true	false	false
Oscar	1999-05-12	oscar@gmail.com	brystsvømning	true	true	true
Lucas	1997-03-17	lucas@gmail.com	rygcrawl	true	true	true
Victor	1990-05-19	victor@gmail.com	rygcrawl	true	true	true
Aksel	1997-09-12	aksel@gmail.com	butterfly	true	false	true
Hannah	1999-05-09	hannah@gmail.com	butterfly	true	false	true
Nanna	1998-04-08	nanna@gmail.com	crawl	false	false	true
Naja	1999-09-10	naja@gmail.com	crawl	true	false	true
Magnus	1995-03-14	magnus@gmail.com	brystsvømning	false	false	true
Felix	1999-07-18	felix@gmail.com	butterfly	true	true	true
Eva	1990-04-18	eva@gmail.com	butterfly	true	false	false
Alfred	1988-03-15	alfred@gmail.com	rygcrawl	true	false	true
Bob	1999-03-03	bob@gmail.com	brystsvømning	true	true	true
Sille	2004-04-05	sille@gmail.com	rygcrawl	true	true	true
Emil	1999-04-07	emil@gmail.com	brystsvømning	false	true	true
Ole	1999-05-02	ole@gmail.com	brystsvømning	true	false	false
Elma	2004-03-05	elma@gmail.com	crawl	true	true	false
Lise	2004-03-09	lise@gmail.com	crawl	true	true	true

Træner:



Logout

Medlemmer

Svømmersultater

Svømmediscipliner

Discipliner top 5


Medlemmer

Refresh

Database information Medlemmer:

Name	Date of birth	eMail	Disciplin	Status	Konkurrencesvømmer	Payment
Klaus	1989-02-08	klaus@gmail.com	rygcrawl	true	true	true
Stine	1999-08-02	stine@gmail.com	butterfly	true	false	true
Tine	1979-05-02	tine@gmail.com	crawl	false	true	false
Henrik	1989-12-12	henrik@gmail.com	rygcrawl	true	false	true
Anna	1993-04-08	anna@gmail.com	rygcrawl	true	true	true
Freja	1995-02-08	freja@gmail.com	brystsvømning	true	false	false
William	1994-12-27	william@gmail.com	rygcrawl	true	false	false
Oscar	1999-05-12	oscar@gmail.com	brystsvømning	true	true	true
Lucas	1997-03-17	lucas@gmail.com	rygcrawl	true	true	true
Victor	1990-05-19	victor@gmail.com	rygcrawl	true	true	true
Aksel	1997-09-12	aksel@gmail.com	butterfly	true	false	true
Hannah	1999-05-09	hannah@gmail.com	butterfly	true	false	true
Nanna	1998-04-08	nanna@gmail.com	crawl	false	false	true
Naja	1999-09-10	naja@gmail.com	crawl	true	false	true
Magnus	1995-03-14	magnus@gmail.com	brystsvømning	false	false	true
Felix	1999-07-18	felix@gmail.com	butterfly	true	true	true
Eva	1990-04-18	eva@gmail.com	butterfly	true	false	false
Alfred	1988-03-15	alfred@gmail.com	rygcrawl	true	false	true
Bob	1999-03-03	bob@gmail.com	brystsvømning	true	true	true
Sille	2004-04-05	sille@gmail.com	rygcrawl	true	true	true
Emil	1999-04-07	emil@gmail.com	brystsvømning	false	true	true
Ole	1999-05-02	ole@gmail.com	brystsvømning	true	false	false
Elma	2004-03-05	elma@gmail.com	crawl	true	true	false
Lise	2004-03-09	lise@gmail.com	crawl	true	true	true

Kassér:



Logout

Medlemmer

Kontingenter

Medlemmer

Refresh

Database information Medlemmer:

Name	Date of birth	eMail	Disciplin	Status	Konkurrencesvømmer	Payment
Klaus	1989-02-08	klaus@gmail.com	rygcrawl	true	true	true
Stine	1999-08-02	stine@gmail.com	butterfly	true	false	true
Tine	1979-05-02	tine@gmail.com	crawl	false	true	false
Henrik	1989-12-12	henrik@gmail.com	rygcrawl	true	false	true
Anna	1993-04-08	anna@gmail.com	rygcrawl	true	true	true
Freja	1995-02-08	freja@gmail.com	brystsvømning	true	false	false
William	1994-12-27	william@gmail.com	rygcrawl	true	false	false
Oscar	1999-05-12	oscar@gmail.com	brystsvømning	true	true	true
Lucas	1997-03-17	lucas@gmail.com	rygcrawl	true	true	true
Victor	1990-05-19	victor@gmail.com	rygcrawl	true	true	true
Aksel	1997-09-12	aksel@gmail.com	butterfly	true	false	true
Hannah	1999-05-09	hannah@gmail.com	butterfly	true	false	true
Nanna	1998-04-08	nanna@gmail.com	crawl	false	false	true
Naja	1999-09-10	naja@gmail.com	crawl	true	false	true
Magnus	1995-03-14	magnus@gmail.com	brystsvømning	false	false	true
Felix	1999-07-18	felix@gmail.com	butterfly	true	true	true
Eva	1990-04-18	eva@gmail.com	butterfly	true	false	false
Alfred	1988-03-15	alfred@gmail.com	rygcrawl	true	false	true
Bob	1999-03-03	bob@gmail.com	brystsvømning	true	true	true
Sille	2004-04-05	sille@gmail.com	rygcrawl	true	true	true
Emil	1999-04-07	emil@gmail.com	brystsvømning	false	true	true
Ole	1999-05-02	ole@gmail.com	brystsvømning	true	false	false
Elma	2004-03-05	elma@gmail.com	crawl	true	true	false
Lise	2004-03-09	lise@gmail.com	crawl	true	true	true

Sikring i forhold til de ikke-funktionelle krav

I forhold til de ikke funktionelle krav som indgik i opgaveformuleringen, har vi i mange af tilfældene lavet exception handling eller formateret brugerinput så der ikke kommer for mange ulovlige argumenter. Æstetisk har vi sikret os at alle vinduer der fungere som child og parrent pannels, ikke kommer i konflikt med hinanden og at deres synlighed derfor kommer an på rollen af brugeren. Dette betyder også at vi ikke kommer til at have problemer med visuelle bugs der kunne resultere i begrænset brugeradgang. Login og log bout funktionerne er også tilpasset denne metode og knyttes igen til rollernes adgang til panelerne. Login mekanismen selv henter fra en text pil som gemmer en krypteret version af login'et så programmet kun reagere på denne data. Ud over det har vi fået filereaderen til at ignorerer whitespaces foruden " , " som den splitter Strings til String[] med. Dette resultere i at det at fjerne en medlem og det efterfølgende whitespace ikke kommer i komplikation med implementationen af en ny bruger og det resulterende med med whitespaces mellem brugerne når .txt filen læses med filereaderen.

Med alt dette taget i betragtning er der selvfølgelig nogle ting som vi gerne ville have lagt noget mere vægt på, som for eksempel det at kunne skrive æøå uden at programmet fejler i at indlæse filen, og at vi måske kunne have nogle flere parseToLowerCase funktion på bruger inputtet, så filereaderen har en højere chance for at kunne læse og reagere på sorteringen af brugerinput i tabellerne.. Dog foruden dette føler vi at programmet virker robust og køre rimelig stabilt.

Exception handling og automatiserede unit test

Delfin projektet er et administrativt system som afhænger af at personer som for eksempel en træner eller en admin kan tilføje og fjerne nye medlemmer. I disse tilfælde afhænger hele den fortsatte funktion af programmet af input fra brugeren. I disse scenarier er der implementeret exception handling til at sørge for at alt input som har en stor betydning for programmets funktion som for eksempel fødselsdato bliver skrevet ind korrekt. Da det er et administrativt system og ikke et system til interaktion med medlemmer af svømmeklubben er visse ting som for eksempel indtastning af navn op til brugeren selv at sørge for er korrekt, da det ikke har nogen betydning for programmets funktion om et navn er indtastet forkert.

I programmet er der tests af alle de metoder der sørger for at programmet fortsat fungerer. Disse tests tester funktionaliteten af kontingent beregning og behandling af nye medlemmer hvilket er de to vigtigste metoder.

De metoder udover dem der står for at hive information ud af dokumentet testes. Det er de classes vi i projektet har ment der var størst grund til at have en test af, og dermed også dem som menes at være vigtigst og har mest brug for opretholdelse.

Navngivning af klasser, variable og metoder, dokumentation (javadoc, commits, uml)

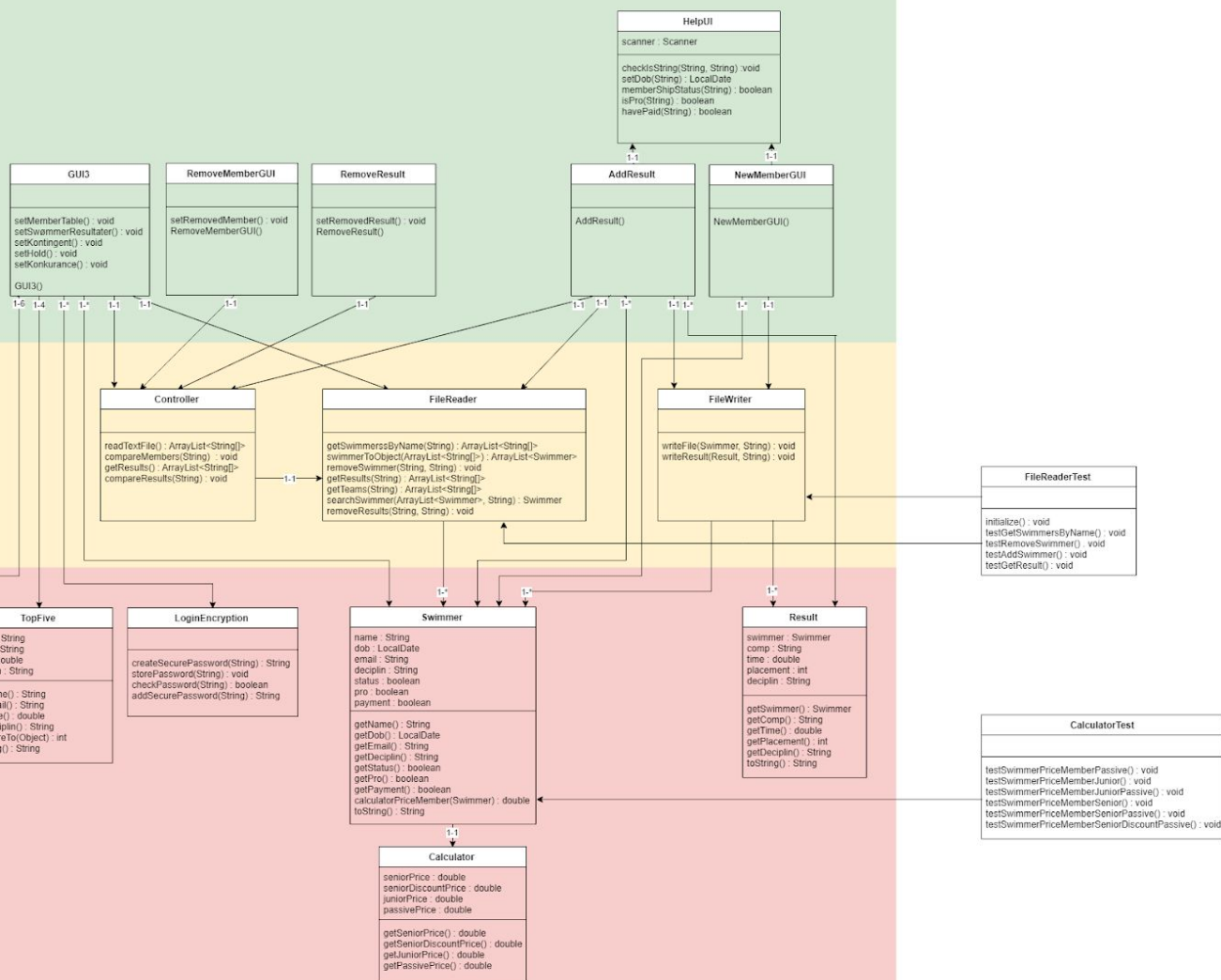
Programmet er opbygget på en tre lags struktur hvor der findes et logik lag, et data-lag og et brugergrænseflade lag. Programmet er opbygget på denne måde for at klargøre og adskille classes baseret på deres funktion. Alle classes der har med ui at gøre lige sammen osv. Dette gør programmet overskueligt og det er en almen struktur at bruge så allerede her er der god læselighed. Alle classes er navngivet sådan at man straks har en ide om hvilken funktion de har og hvilke metoder de indeholder. Vi har i projektet haft use case scenarios og domænemodellen i bagtankerne, det færdige produkt er designet ud fra mockup.

Også efter vi færdiggjorde programmet, dokumenterede vi for vores valg gennem vejledende javadoc kommentare og en mere fuldstændig UML designmodel baseret på de valg vi gjorde i den forberedende UML model i starten af projektet. Den fuldstændige designmodel kan ses nedenunder.

User Interface

Data

Logic



Modulært design, hjælpeklasser

Metoderne som bliver brugt i GUI'en bliver kaldt via vores Controller klasse. Udskiftning eller modifikationer i frontend-laget vil derfor være mindre vanskeligt at gennemføre, da alle metoderne ligger i controller-klassen. Ved udbygning af flere klasse med metoder som skal bruges i frontend-laget ville det være meget fordelagtigt at tilføje dem til controller-klassen. Det gør det hele mere overskueligt. Programmet er opdelt i 3 lag. Datalaget hvor alt information gemmes. Dataaccessorlaget hvor metoderne til at hente data'en ligger og præsentationslaget hvor data'en bliver vist. De tre lag kan ses visuelt i projektet, da vi for overskuelighedens skyld har oprettet en pakke til hver del af programmet. Det vil også gøre det lettere for en udefrakommende programmør at forstå hvordan klassernes opdeling er.