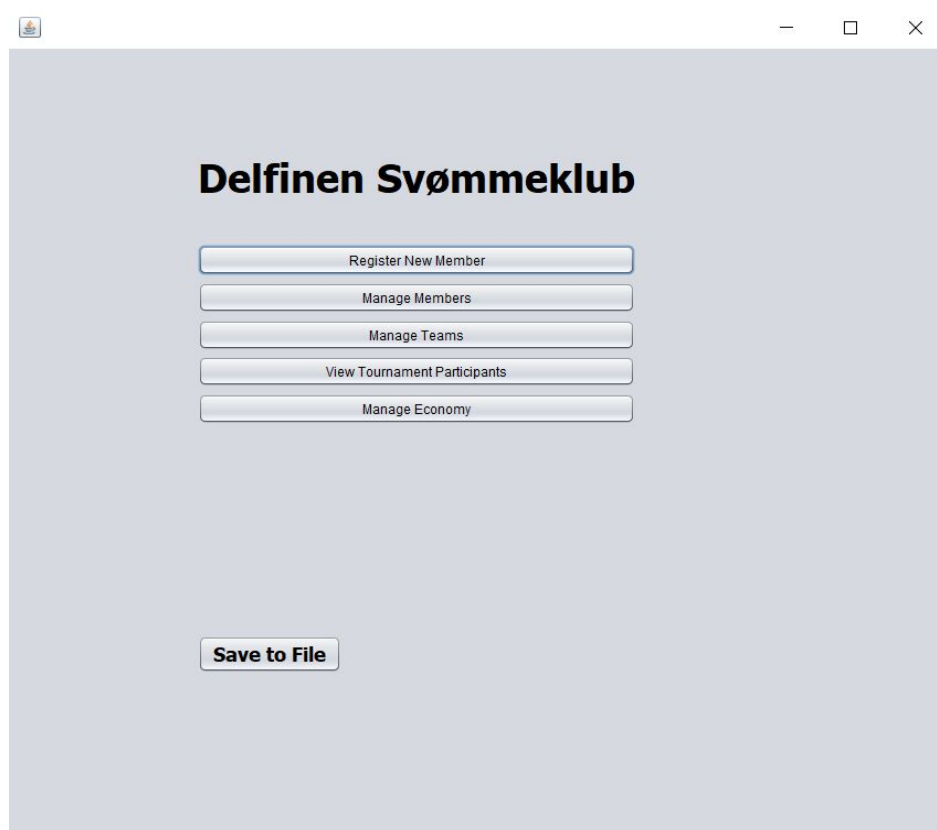


Som beskrevet i vores use-case havde vi tænkt os at inkludere tre aktører, der hver især skulle have en act-funktion: formand(chairman), der skulle kunne oprette og redigere medlemsinfo (member info); træner (trainer), der skulle kunne oprette og redigere hold og kasserer (treasurer), der skulle kunne holde øje med kontingentbetaling (membership price payment) for hvert medlem.

Vi har formået til dels at opfylde de krav vi har sat os selv; der mangle visse funktioner, men de grundlæggende er implementeret.

I forhold til casen mangler der specifikt at kunne se top 5 lister over konkurrencesvømmere (competitive swimmers). Koden er skrevet, men vi mangler at implementere den i GUI'en. Vi har også visse andre mangler i form af diverse mindre funktionaliteter, som vi vil beskrive nærmere:



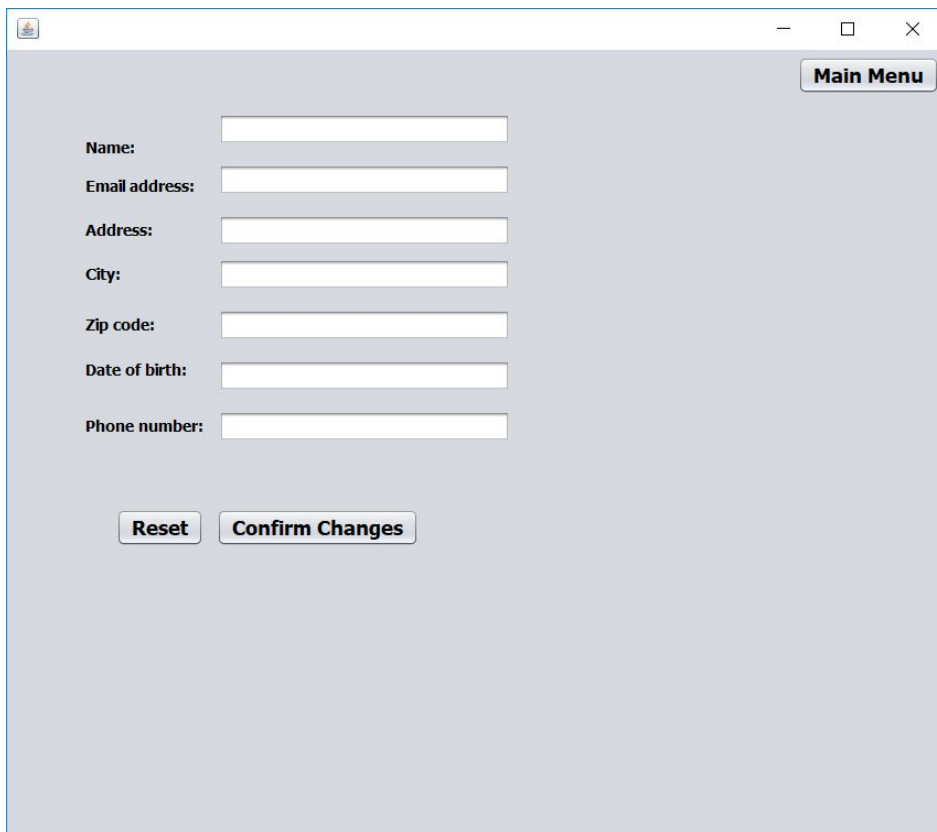
Der er blevet implementeret en main menu. Denne fungerer som navigatør i programmet. I forhold til de funktionelle krav er der en knap, der fører til "View Tournament Participants", men tabellen i det menupunkt mangler at blive kodet til at populere tabellen. Ligeledes mangler main menuen en knap, der viser en top 5 liste.

Knappen "Save to File" gemmer som beskrevet dataerne til en fil. Denne fil er implementeret gennem et interface (DataAccessor), så i tilfælde af at man vil tilkoble en anden form for data opbevaring, skal knappen også renames. I alle metoder er det beskrevet som "Source" istedet for File.

Vores DataAccessorFile (implements DataAccessor) har funktionalitet til at gemme ArrayLists (skulle have været Lists, men nåede ikke at ændre det) af henholdsvis <Member>, <Team> og <TournamentEvent> objekter. Vi nåede kun at implementere de to første, men de virker også.

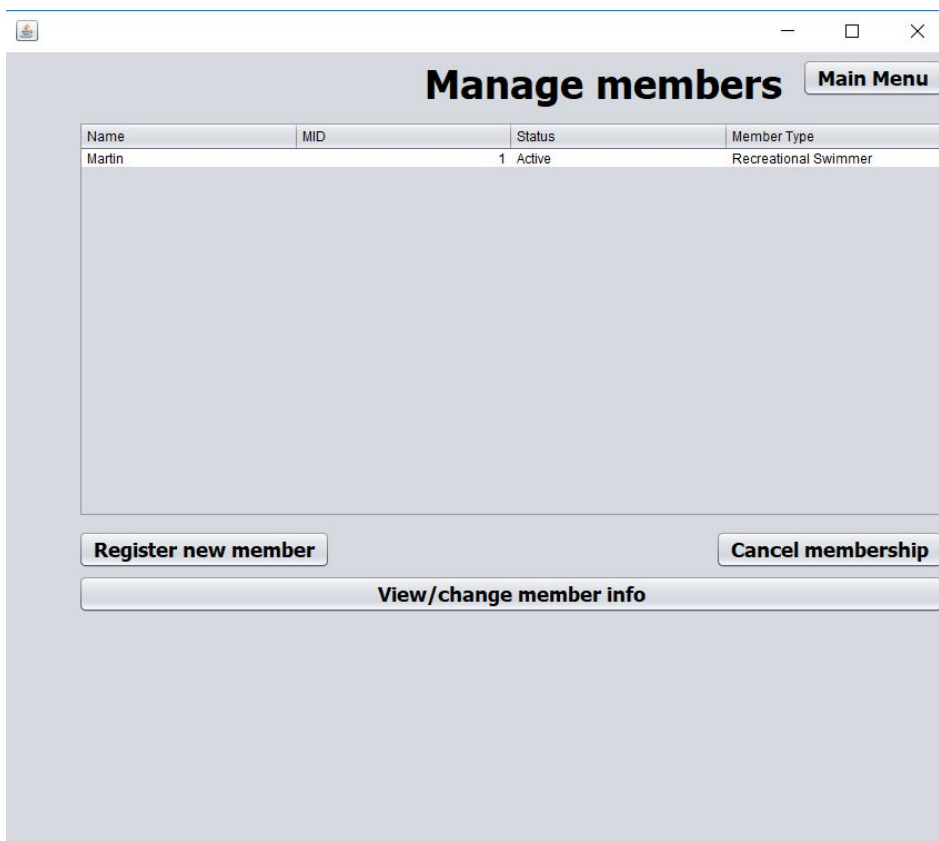
De fungerer ved at vi implementerer et Serializable interface på vores objekter, og så gemmer vi alle objekter fra en ArrayList direkte til en fil gennem en toString metode (som kun er Override't i Member).

Knappen "Register New Member" fører videre til et vindue, hvor der dukker nogle tekstfelter op, som skal udfyldes.



A screenshot of a registration form window. The window has a title bar with a small icon and standard window controls. In the top right corner, there is a button labeled "Main Menu". The form contains several input fields with labels to their left: "Name:", "Email address:", "Address:", "City:", "Zip code:", "Date of birth:", and "Phone number:". Below these fields are two buttons: "Reset" and "Confirm Changes".

Når man klikker på knappen "Confirm Changes" kontrollerer programmet om inputtet i tekstfelterne er gyldigt. Hvis inputtet ikke er gyldigt, popper en messagebox op, der specificerer hvilket tekstfelt, der har ugyldigt input, ellers, hvis oplysninger er gyldige, bekræfter messageboxen, at medlemmet er blevet registreret. Dernæst sættes alle tekstfelterne til at være tomme, så de er klar til at man kan indregistrerer næste medlem. Knappen Reset sætter felterne til tomme.



A screenshot of a "Manage members" window. The window has a title bar with a small icon and standard window controls. The title "Manage members" is centered at the top, and a "Main Menu" button is in the top right corner. Below the title is a table with four columns: "Name", "MID", "Status", and "Member Type". The table contains one row with the data: "Martin", "1", "Active", and "Recreational Swimmer". Below the table are three buttons: "Register new member", "Cancel membership", and "View/change member info".

Name	MID	Status	Member Type
Martin	1	Active	Recreational Swimmer

Knappen "Manage Members" fører videre til et vindue, der viser en tabel over medlemmer i klubben.

Knappen "Cancel membership" er ikke blevet programmeret.

Knappen “Register New Member” fører videre til vinduet beskrevet ovenover. Knappen “View/change member info” fører videre til et vindue, hvor man kan se og ændre oplysninger på det medlem, som man havde markeret, da man klikkede på knappen.

The screenshot shows a web application window with a title bar and standard window controls. The main content area is divided into several sections:

- Member Information:** A form with fields for Name (Martin), Email Address (martin.l.brandstrup@gmail.com), Address (Fiktiv), City (Farum), Zip code (3520), Date of birth (18/05/1993), and Phone number (12345678). A "Confirm changes" button is located below these fields.
- Financials:** Fields for Activity status (Active), Member ID (1), Member type (Recreational Swimmer), Date of joining (23/11/2018), Arrears balance (0 kr.), Next payment date (23/11/2018), and Membership price (1600,00 kr.).
- Tournament eligibility:** A checkbox labeled "Tournament eligibility".
- Team membership:** A table with columns: Team name, Swimming style, Date of joining, and Date of leaving.
- Tournaments participants:** A table with columns: Event name, Event date, Ranking, Swimming style, and Time.

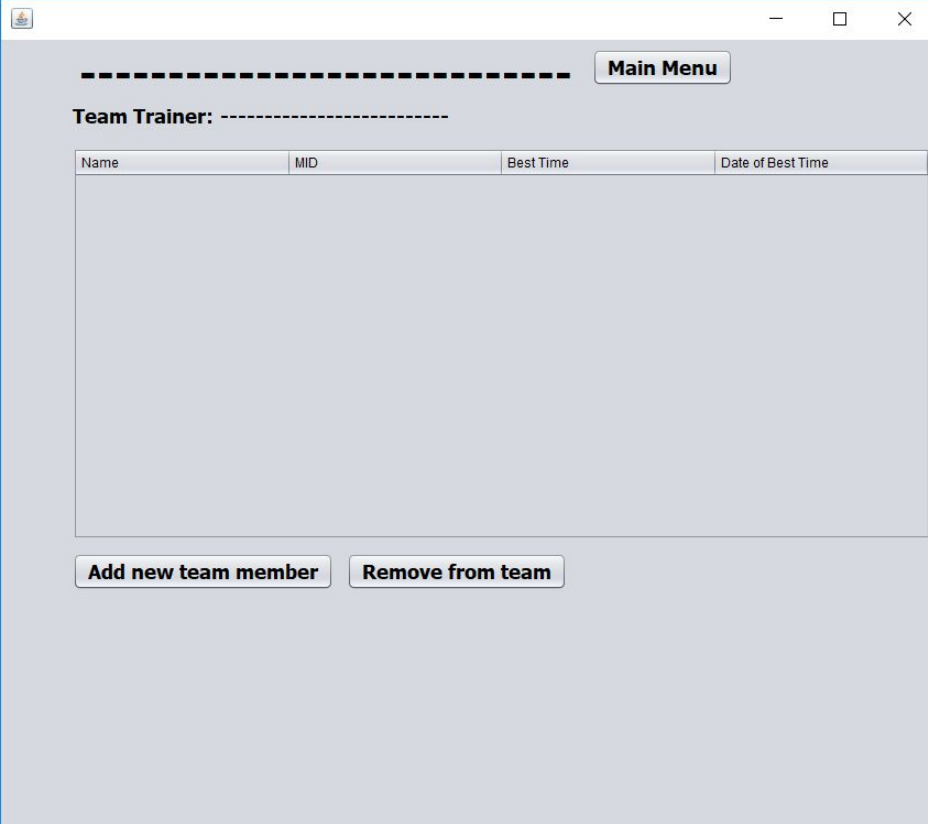
A "Main Menu" button is located in the top right corner.

I dette vindue, er der to tabeller: “Team membership og Tournaments participants”, som ligeledes ikke er programmeret. Derudover er der knappen “Confirm changes”, der gemmer ændringer foretaget, og dette bliver bekræftet ved at der popper en messagebox op.

Knappen “Manage Teams” fører videre til et vindue, bestående af otte tabeller. Fire for junior, med hver deres svømmestil, og ligeledes fire for senior.

The screenshot shows a web application window titled "Junior teams:" with a "Main Menu" button in the top right corner. The content is organized into two main sections: "Junior teams:" and "Senior teams:". Each section contains four columns representing different swimming styles: Breasstroke, Crawl, Back crawl, and Butterfly. Each column has a "Teams" table with a list of teams (e.g., Team A) and a "View members" button below it. At the bottom of the window, there is a "Create a new Team" button.

Her kan man se alle hold i klubben, vælge et hold, og klikke på den tilhørende “View members” knap. Dette fører videre til et nyt vindue, hvor man kan se medlemmerne i det valgte hold.



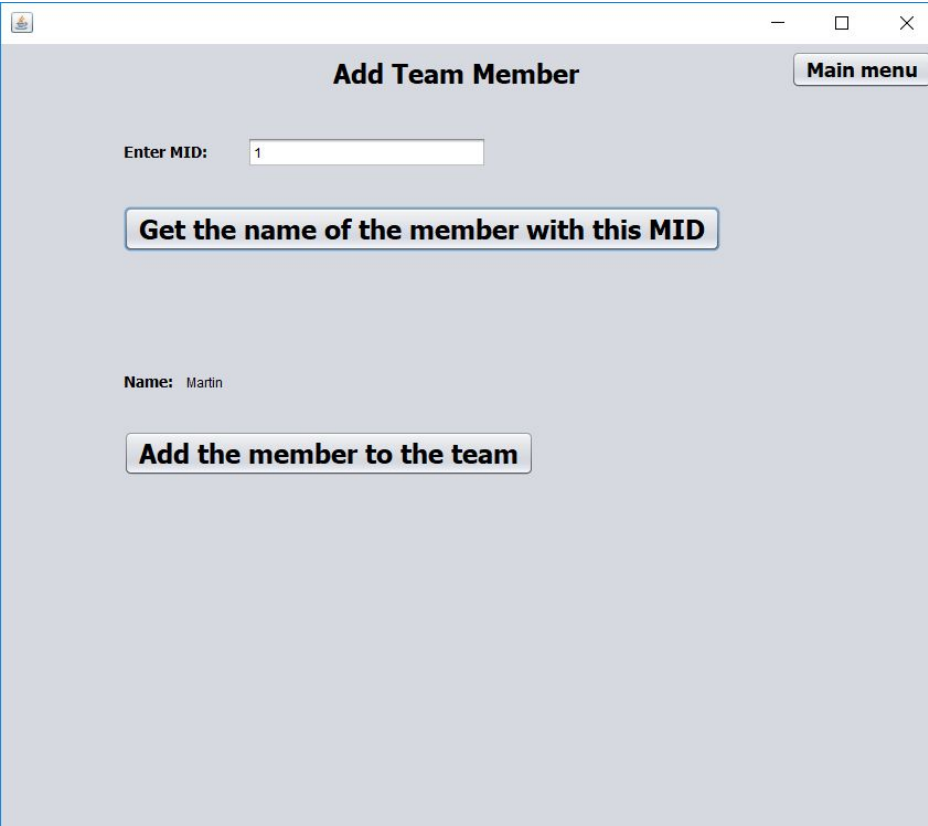
----- **Main Menu**

Team Trainer: -----

Name	MID	Best Time	Date of Best Time
------	-----	-----------	-------------------

Add new team member **Remove from team**

Der er yderligere to knapper “Add new team member” og “Remove from team”. Den sidste af de to knapper, er ikke blevet programmeret. “Add new team member” fører derimod til et vindue.



Add Team Member **Main menu**

Enter MID:

Get the name of the member with this MID

Name: Martin

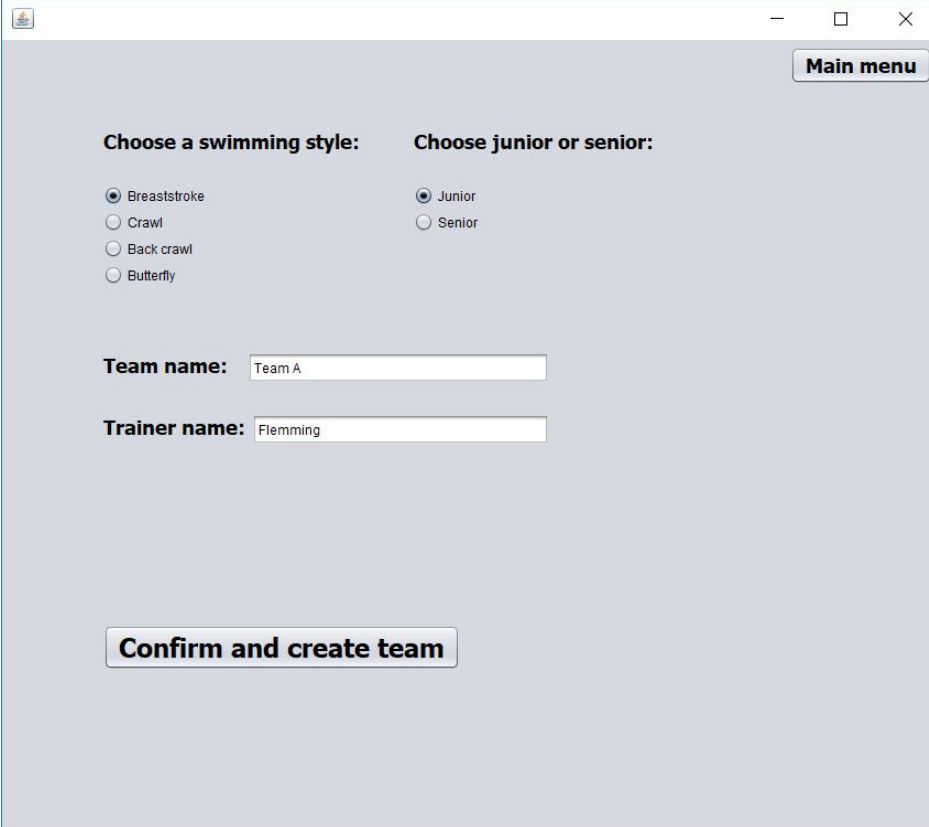
Add the member to the team

Her kan man indtaste et medlemsnummer (MID), og klikke på knappen “Get the name of the member with this MID”, som tjekker, om der er et medlem med det givne medlemsnummer.

Hvis der ikke er gyldigt match, popper en fejlmeddelelse op, der skriver, at der intet medlem er med det indtastede medlemsnummer. Hvis der derimod er et gyldigt medlem med det givne medlemsnummer, bliver navnet skrevet ud for feltet "Name: " og knappen "Add the member to the team" bliver nu klikbar.

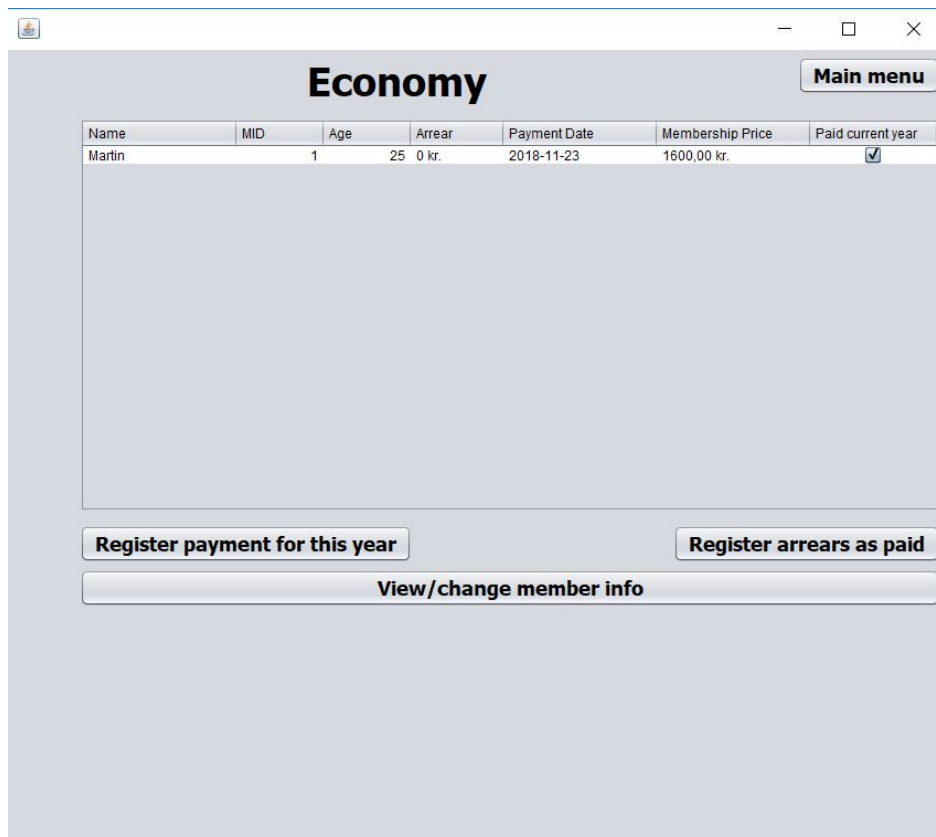
Hvis man klikker knappen popper en box op, der bekræfter at medlemmet er blevet tilføjet til holdet.

I vinduet bestående af de otte tabeller er der desuden en anden knap "Create a new team", som fører videre til et vindue, hvor man skal vælge et svømmestil, om holdet skal være junior eller senior, samt navnet på holdet, og holdets træner.



The screenshot shows a web application window with a light blue background. In the top right corner, there is a button labeled "Main menu". The main content area is divided into two columns. The left column is titled "Choose a swimming style:" and contains four radio button options: "Breaststroke" (selected), "Crawl", "Back crawl", and "Butterfly". The right column is titled "Choose junior or senior:" and contains two radio button options: "Junior" (selected) and "Senior". Below these columns, there are two text input fields. The first is labeled "Team name:" and contains the text "Team A". The second is labeled "Trainer name:" and contains the text "Flemming". At the bottom of the form, there is a large button labeled "Confirm and create team".

Ved at klikke på knappen "Confirm and create team" laves holdet hvis oplysningerne er udfyldt korrekt, og dette bliver også bekræftet ved, at der popper en boks op, der skriver om holdet er blevet oprettet.



Manage Economy tager brugeren til Economy menuen. Economy vinduet har følgende knapper: “Main Menu”, “Register payment for this year”, “Register arrears as paid” og “View/change member info”. Desuden indeholder denne undermenu en tabel med en oversigt over medlemmerne. Det fremgår heraf hvor meget medlemmerne mangler at betale klubben.

“Register payment for this year” markerer kontingentet som betalt for det medlem, som er blevet valgt fra tabellen.

“Register arrears as paid” nulstiller gælden for det valgte medlem.

“View/change member info” fører til vinduet som beskrevet under manage members afsnittet.

I vores domain-model har vi beskrevet Member som havende en underklasse, CompetitiveSwimmer. Under selve kodningen endte vi med at fjerne den extended class, som var CompetitiveSwimmer, da vi havde tidsmangel og ikke kunne finde gode løsninger på alle de problemer der opstod. Hovedsageligt at vi kun arbejdede med en List<Member>, som ikke kunne udnytte en Subclass af Member på en praktisk måde, hvis dette objekt (subclass) igen skal gemmes og bruges i samme ArrayList.

Vi endte med at skelne forskellen mellem de to typer af medlemmer ved hjælp af en boolean, der bestemmer, om mange af de parametre og funktioner de besidder kan bruges. Dette medførte dog desværre at vores objekt Member har en ekstrem lang parameterliste; dette kunne godt forbedres, hvis vi havde mere tid.

Der er blevet lavet javadoc api og tests til de fleste af de mest relevante metoder, men vi må nok erkende, at der er en mangel på flere unit tests.

I generelt omfang vil vi mene at det meste af vores kode er genbrugsvenligt. Mange af metoderne gør kun en ting, der kan bruges i flere sammenhænge. Der er nogle få undtagelser i GUI'en, da vi begyndte at løbe tør for tid, og bare skulle have noget funktionelt kode, så vi kunne præsentere programmet. Vi er selv klar over, hvor i GUI'en der plads til forbedring.

Programmet indeholder nogle få klasser, der ikke er beskrevet i vores klassediagram (vedhæftet UML). Disse er ikke inkluderet, da de indeholder static metoder, der generelt kan bruges i diverse programmer, og vi anser dem derfor ikke som værende en del af dette specifikke projekt. To comparators har sin egen package, men klassen `FormatterAndValidator` fik vi ikke flytte ind i sin egen package.

I forhold til koden i klassen `GUI` er de mange tabeller, knapper, labels og tekstfelter navngivet, således at de er nogenlunde genkendelig og vejledende for den funktion de skal udføre. `GUI`'en er sat op som et `cardLayout`, der har gjort det muligt at have al koden i denne ene klasse, men dette har også gjort det ret uoverskueligt og svært at finde en bestemt metode, som man leder efter, og med dette i betragtning, hvis tiden havde været der, kunne man have opdelt koden i flere hjælpeklasser og eventuelt have overvejet at bruge statiske variabler i sådan et tilfælde.

I `GUI` klassen er der desuden også en del kode vedrørende population af de otte team tabeller (og metoderne `rowSelected()`), som stort set er identisk. Dette er selvfølgelig ikke optimalt, og man burde have fundet en anden løsning til dette.