

Use cases:

Use case 1: Registration of members

Actors: Chairman

Chairman logs into the system and chooses an option "Registrate members". He enters the new members name, age, sex, address and member type. After finishing the operation he gets a confirmation that member is added to list of members.

Use case 2: Removing from arrears list

Actors: Cashier

Cashier has the contingent payment responsibilities in the club. He logs to the system and gets the options "Remove" a member (which means a member has payed), or "See" the list of members, who have not paid for the membership. He chooses an option "Remove" member searching by his name, who should be removed from arrears list. When the member is removed from arrears list, he gets the confirmation of it.

Use case 3: Overview of members

Actors: Coach

The coach logs into the system and chooses an option "Overview of the swimming disciplines". He chooses then a swimming discipline and gets a list/statistic of the top 5 swimmers of each discipline in the club.

Use case 4: Show training results

Actors: Coach

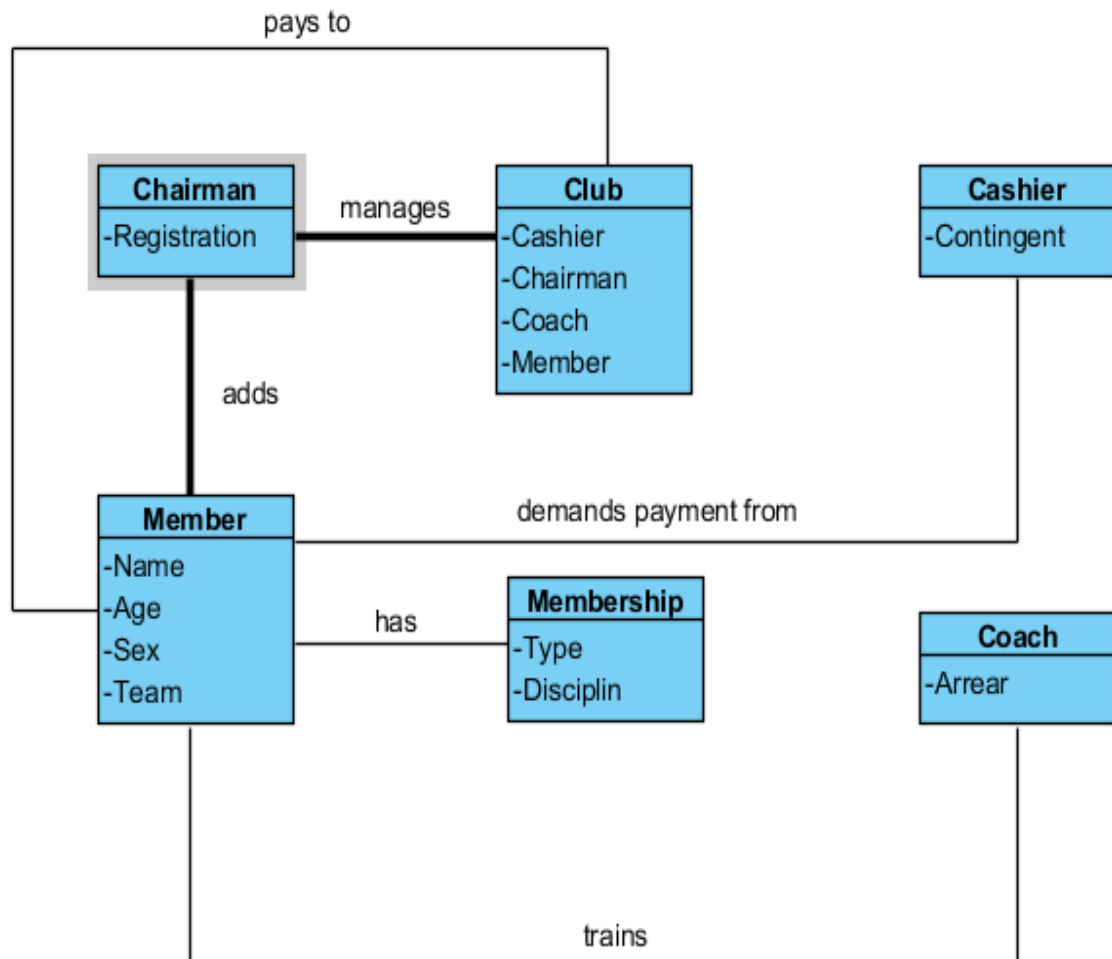
The coach logs to the system and chooses the option "Show best training times". He then chooses the team he is interested in getting results from: junior or senior. After choosing he gets the list of competition swimmers, who have the best training results. He then chooses the swimmers with best results and makes a team to the next competition.

Nouns: Chairman, request, **registration**, member, **membership**, club, information, name, age type, team, discipline, **contingent**, payment, system, options, list, status, arrear, **coach**, cashier, overview, competition, result, swimmer.

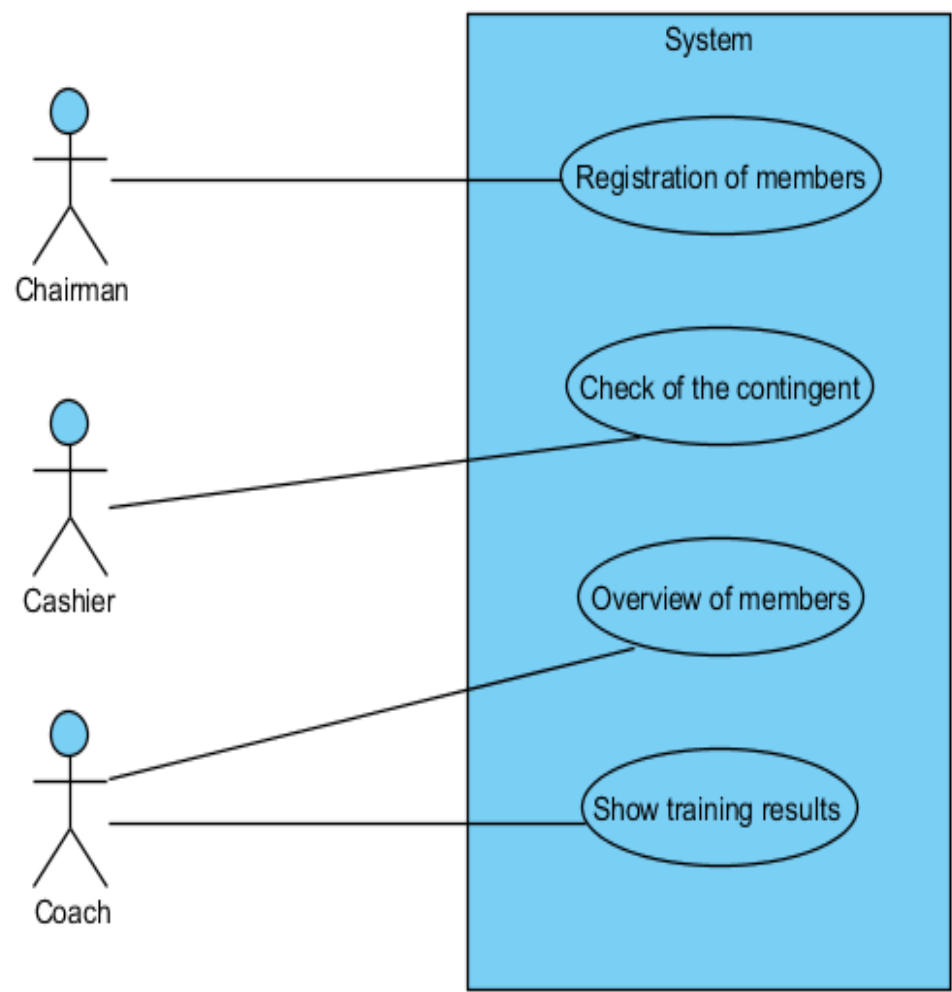
Classes: Chairman, coach, cashier, member, club, membership

Diagrams and models:

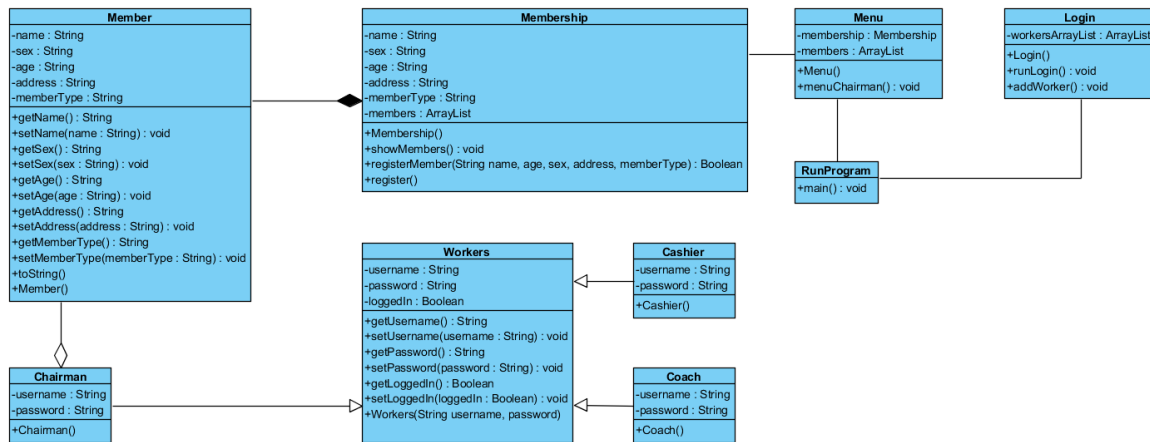
Domain Model:



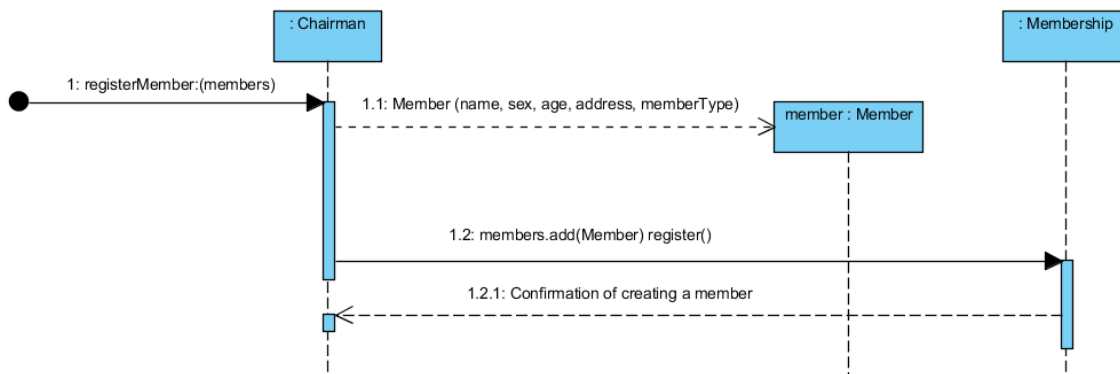
Use case diagram:



Class diagram:

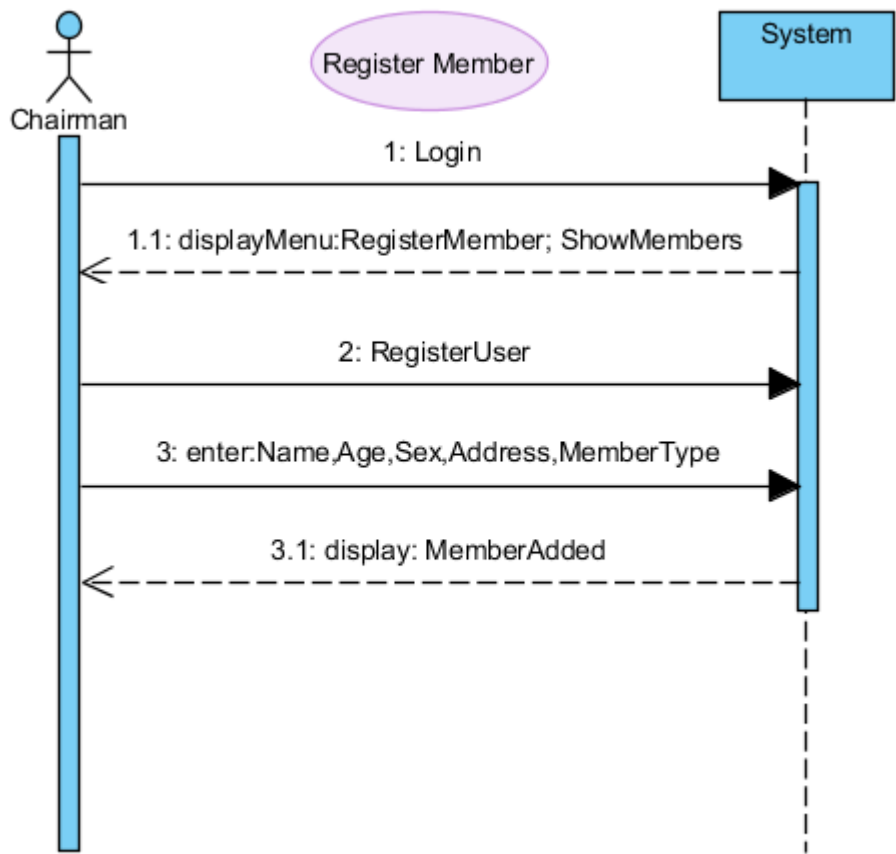


Sequence diagram for UC1:

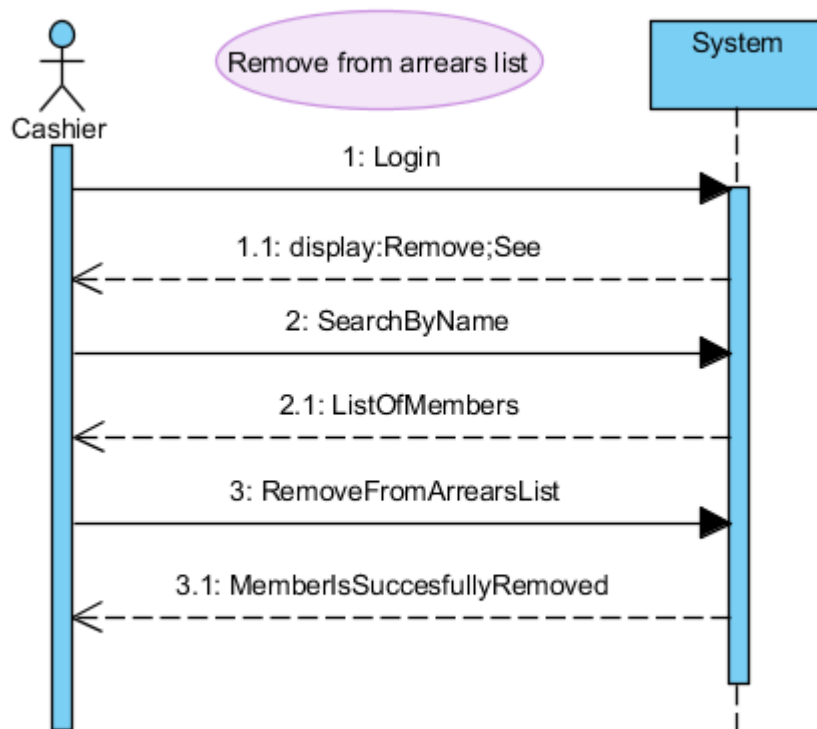


System sequence diagrams:

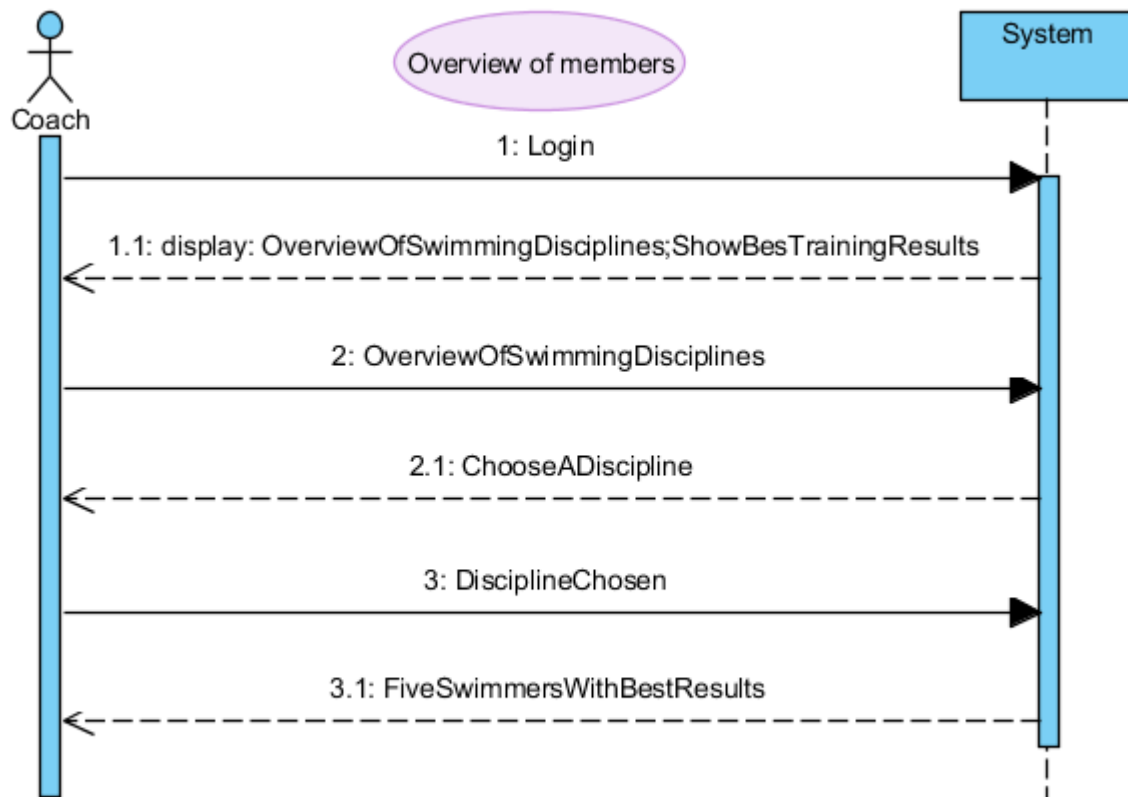
Register Member:



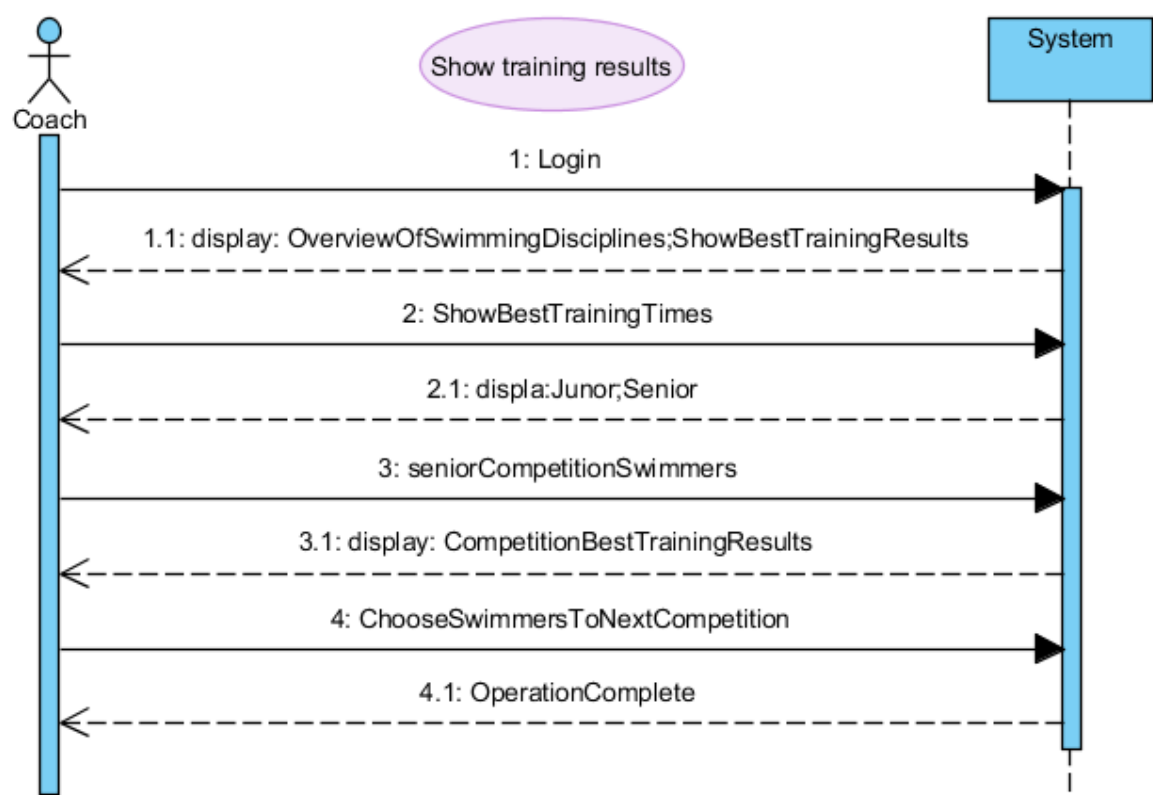
Remove from list:



Overview of members:



Show training result:



IT-organization:

Stakeholders analysis:

To make a stakeholder analysis we first need to establish who and what the stakeholders are.

In our case we have these stakeholders:

- Chairman
- Cashier
- Coach
- Members

High power		Chairman Cashier
Low power		Coach Members
	Low interest	High interest

We have the put them in a diagram with low power high power. The reasons for the different stakeholders' placement:

Chairman: We placed the chairman in high power and high interest since he is the one handling the club. He is the responsible "owner" of the swimming club. The chairman also has the high interest for the club to develop in the direction he wants also, is the one with the must influence in the businesses.

Cashier: The cashier has a lot of power in the swimming club, since he is the one controlling the economic. Cashiers are a part of the management, who stands for the club developing in the right direction since they are the once making sure that the members pay their contingent.

Coach: The coaches are the importance of the daily run of the club. Coaches are the once with interest in the club running smoothly. It's important for them that the club develops in the competition swimming and the administrative so they have optimal options of for filling their job. Since they are employees in the club they don't have a lot to do with the management. Therefore, they have no responsibilities in the clubs deployments or its economy. They simply need to do their jobs and that is the reason we have put them in low power and high interest.

Members: members have been placed in low power and high interest. Members don't have any power in running the club, since they have no influence in how the club will develop. They do however have interest in the club having a good development and that the facilities become better in every way. Especially the competition swimmers will have good benefit of the club becoming more competitive.

We made a SWOT analysis to try to understand the clubs' strategies and its position in relation to other swim clubs.

Strength: Growth Positive economics Good prices	Weaknesses: Unknown club Bad training hours
Opportunities:	Threats:

Better facilities	Competition clubs
Advertising	Others sports

Here we have deliberated the clubs' strengths as we know it will enhance the clubs position and development.

Growth:

Delfin is in growth and is growing. Growth reflects good economic and increases the interest of the club. Both for the potential member and sponsors.

Positive economic:

Delfin revenues becomes bigger and the economic becomes better.

Good prices:

Delfin has cheap memberships compared to other swim clubs.

Here are the factors for the club not moving forward:

Unknown club:

Since it is a little swimming club, it's not that popular or known.

Bad training hours:

The training hours can be unfitting for certain groups of the members. Ex. Members that can make the times because they are at work or studying.

The club could develop on certain ways as in:

Better facilities:

Better facilities could increase the interest of the club. Such as better trainers and swimmers which would increase the sponsorship and the advertising of the club would be bigger.

Advertising:

Advertising in all schools would make it more attractive for younger people.

The club still has threats from outside which can stop the growth and development:**Competition clubs:**

Clubs with either cheaper or better facilities can make a treat for the club.

Other sports:

If a swimmer prefers another sport and chooses to drop the swimming.

About our code:

The work on the code didn't went how it should. We rewrote the code several times, because it did not work properly. The main problem was with the login system, which we could not make to work the way it should. The problem was at giving different users different possibilities to manage the program: show arrears list, register new members, training results etc. The problem was not about making the needed functions, but it was that function did not speak to each other: could not see the getters from another class for example, can handle the text file, conflict between each other. After fighting about the code, watching videos and reading tons of tutorials for how to write some functions, we decided that we should start all over again and make at least one function. We chose it to be Chairman use case, where we could register new members. It went well, though we couldn't make the program show us the list of members due some fail with static. At the end we ended up creating code only for one use case. If we had time, we could finish other use cases too, because we found the right way to make the code work, but it was too late.