Use cases:

Title: CRUD Member
Actors: President
Main success scenario:

1. President opens system
2. President chooses to Create/Read/Update/Delete member
3. System asks for relevant member data
4. President inputs relevant data
5. System prints out a receipt of the request and changes where relevant

Title: Retrieve debtors
Actors: Cashier
Main success scenario:

1. Cashier opens system
2. Cashier chooses to view list of current debtors
3. System retrieves list from database
4. System prints out list of current debtors

Title: Pay membership dues
Actors: Cashier
Main success scenario:

1. Cashier opens system
2. Cashier chooses to pay dues
3. System prints out list of current members with due-relevant data
4. Cashier chooses any number of members from the list
5. System asks for confirmation
6. Cashier confirms
7. System prints out a receipt for the payment updates

Title: Add swimmer to team
Actors: Coach, (President)
Main success scenario:

1. Coach opens system
2. Coach chooses to edit team
3. System displays a list of current teams
4. Coach chooses to add new swimmer
5. System asks for relevant data
6. Coach inputs data
7. System prints out a receipt with the updates

Title: Retrieve top 5 swimmers
Actors: Coach
Main success scenario:

1. Coach opens system
2. Coach chooses to display leaderboards
3. System asks for which discipline to display
4. Coach chooses a discipline
5. System prints out the top 5 for the chosen discipline along with relevant data

Projektrapport

Vi har afgrænset os til to centrale elementer fra vores use case. Vores projektrapport er derfor konstrueret ud fra følgende use cases.

Title: CRUD Member
Actors: President
Main success scenario:

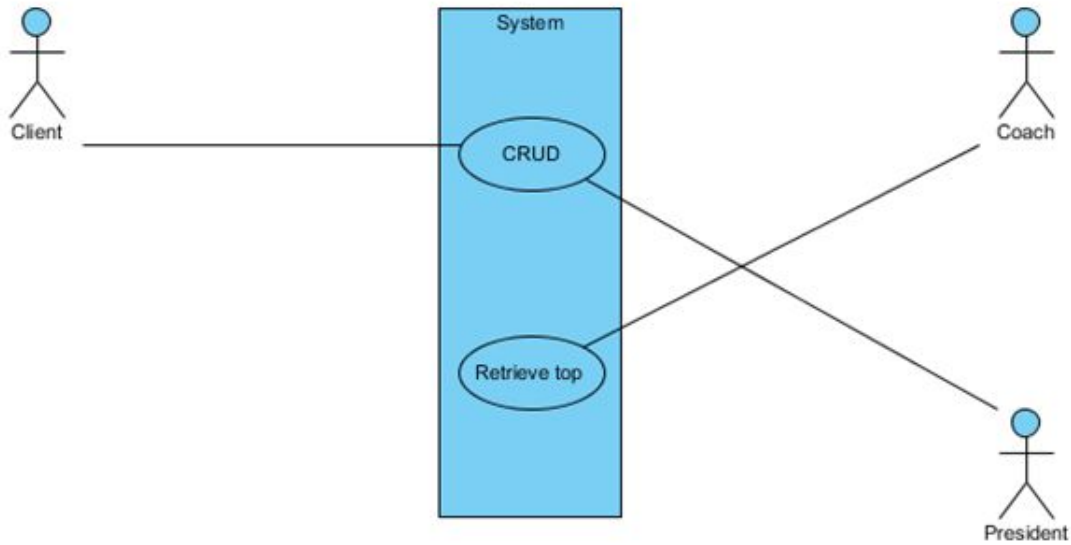
6. President opens system
7. President chooses to Create/Read/Update/Delete member
8. System asks for relevant member data
9. President inputs relevant data
10. System prints out a receipt of the request and changes where relevant

Title: Retrieve top 5 swimmers
Actors: Coach
Main success scenario:

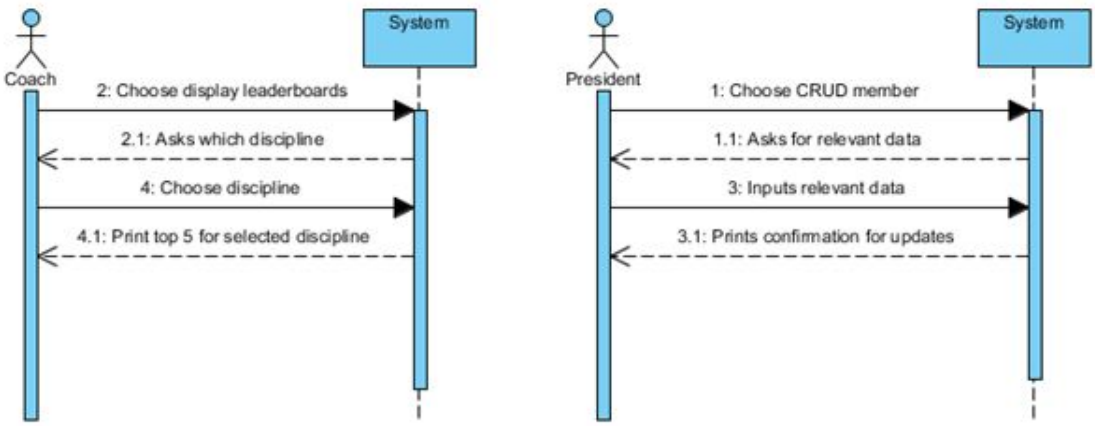
6. Coach opens system
7. Coach chooses to display leaderboards
8. System asks for which discipline to display
9. Coach chooses a discipline
10. System prints out the top 5 for the chosen discipline along with relevant data

Ud fra vores use cases har vi valgt at fokusere på oprettelse af medlemskaber i svømmeklubben, da vi mener at dette vil være en central forudsætning for programmet. Derudover har vi fokuseret på, at træneren vil kunne læse top fem af svømmerne i svømmeklubben. Disse to use use cases vil danne grundlag for udarbejdelsen af et demo program.

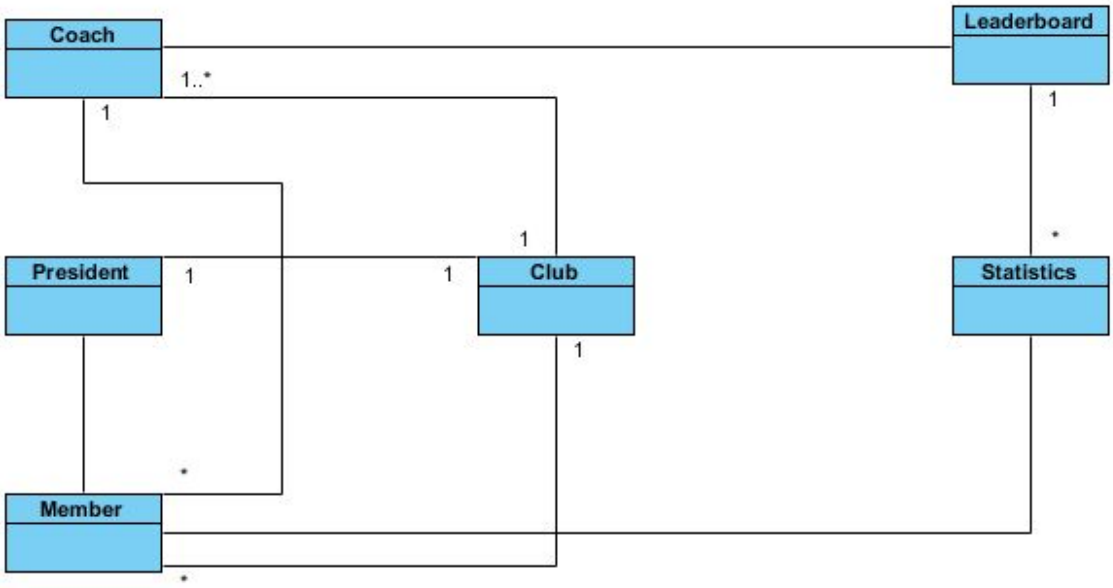
Vores use cases har vi stillet op i et use case diagram, for bedre at kunne danne overblik.



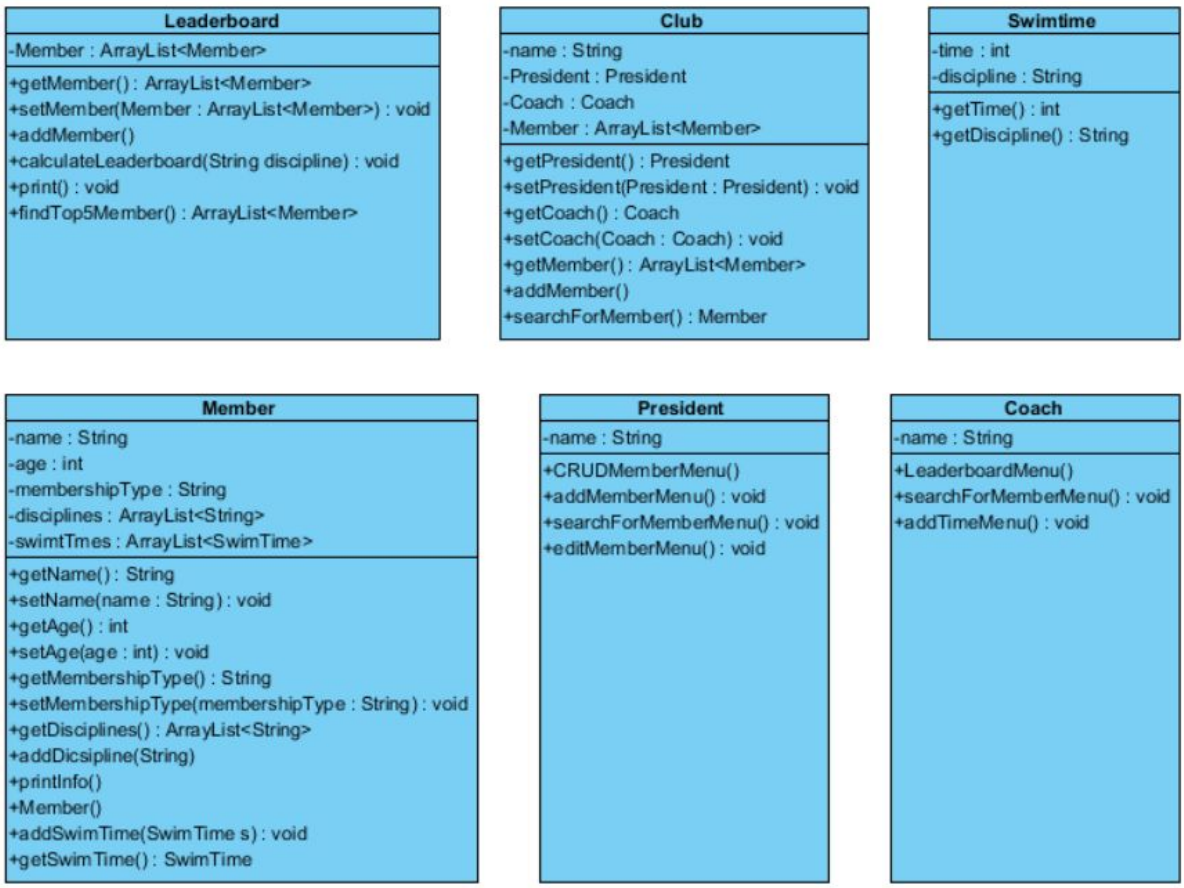
Ud fra vores centrale use cases har vi udarbejdet et SSD diagram. Dette er med til at give et detaljeret overblik over vores udvalgte use cases. Efter dette diagram vil CRUD kun indeholde Create. Vores program er derfor bygget op omkring Create Member og Retrieve top 5.



Ud fra ovenstående, hvor vi har udledt mange informationer omkring demo modellen er vi kommet frem til følgende domænemodel. Domænemodellen har vi udledt, ved at bruge vores use cases til at konstruere en noun og verb list og derefter konstrueret sammenhængen mellem elementerne i vores demo model. Vores udarbejdede domænemodel er afbilledet herunder.



Efter vores udarbejdelse af ovenstående, er vi kommet frem til, at vores demo model skal bygges op om følgende klassediagram.



Program detaljer:

Selve programmet starter i en menu, hvori man vælger om man vil fortsætte som hhv. President, Coach eller Cashier.
Valg af president vil tage én til en side hvor man kan udføre standard CRUD funktioner på Member databasen (OBS! Dette er kun defineret som en ArrayList i vor demo program, men burde implementers som en seperat file så members gemmes imellem program sessions.).
Valg af coach tager en til coach menuen, hvor man har mulighed for at se leaderboard, force update af leaderboard og tilføje svømme tider til Members.
(OBS. for at se leaderboard skal man først force update, hvilket så vil lave et leaderboard ud fra de implementerede test data.)
De implementerede testdata er med discipline “100 m”.
Valg af Cashier gør intet, da vi har valgt at holde det ude for vores demo.

OBS! Da Scanner klassen er lidt mærkelig i forhold til når man bruger f.eks. nextInt(), er der indsat nogle Scanner.nextLine() kald bare for at flushe scanneren. Dette gør dog at man nogle gange bliver nød til at lave et ekstra tastetryk for at komme videre.