**ITO:**

Svømmeklubben delfinen er en engageret svømmeklub der beskæftiger sig med alle borgere af alle aldre. Svømmeklubben er baseret på at der betales kontingent af de medlemmer klubben har. Der er forskellige former for medlemskaber der hver har forskellige priser, dette understøtter at klubben kan håndtere alle kunder.

Klubben indeholder:

* Formand
* Kasserer
* Trænere
* Medlemmer (aktiv, passiv/ junior, senior/ motionist, konkurrencesvømmer)

**Design:**

**Functionality:**

Programmet OrgTool kan oprette og redigere medlemmer, for at skabe et overblik over klubbens aktiviteter og medlemmer. Det kan holde styr på hvert enkelt medlem, og hvert enkelt hold. Man kan let se relationer mellem forskellige medlemmer og trænere, f.eks. kan programmet holde øje med svømmernes bedste tider og kontingentbetaling.

**Usuability:**

Let at bruge, man skal ikke 1000 millioner trin igennem for at udføre trivielle funktioner. Let at tilgå redigering af hold, oprette medlemmer osv.

**Reliability:**

Gemme medlemmer og hold i en fil.

**Performance:**

Programmet kører let som en brise.

**Supportability:**

At programmet er nemt at ændre uden der skal redigeres mange forskellige steder.

**Glossary:**

**Use case:** Oprette medlem

**Scope**: OrgTool program.

**Level:** user goal

**Primary actor:** Formand

**Stakeholders and interest:**

* Formand: Ønsker let brug af program, ved oprettelse af medlemmer.
* Kassere: Ønsker et brugervenligt program, der forløber uden fejl. Overblik over medlemmer og betaling.
* Træner: Ønsker nem adgang til oplysninger om tider og hold.
* Medlemmer: Ønsker at blive oprettet så de kan registrere tider og betaling.

**Preconditions:** Formanden har fået en medlemsanmodning.

**Success guarantee (postcondititon):** Medlemmet er oprettet og alle kan tilgå den information de har brug for at se.

**Main success scenario (basic flow):**

1. Formand modtager medlemsanmodning og skal oprette medlem.
2. Starter oprette medlemsfunktionen.
3. Skriver medlemsinfo (alder og navn).
4. Vælger type af aktivitetsform (aktiv, passiv, junior osv.)
5. Gemmer oplysninger.

**Extension (alternative flow):**

1. Skriver forkert information.
2. Vælger rediger medlemsinfo.
3. Retter sin fejl.
4. Programmet crasher på et vilkårligt tidspunkt inden informationen er gemt.
5. Programmet genstartes.
6. Man starter forfra.

**Special requirements:**

* Formand har en pc.
* En form for database over nuværende medlemmer og hold.

**Frequency of occurence:**

Hver gang en ny medlemsanmodning kommer ind.

**Open issues:**

* Gdpr.
* Personer med samme navne.

Use case: se liste over medlemmer med gæld

Main succes scenario: Kasseren åbner programmet og logger ind, efterfølgende åbnes medlemslisten og kasseren vælger at se en list over medlemmer med gæld. Programmet viser en liste med medlemmer med gæld.

Alternate scenarios:

1. Der er ingen medlemmer der har gæld.
2. Programmet udskriver ingen har gæld.
3. Kasseren kan ikke logge ind.
4. Kasseren nulstiller sit password.
5. Kasseren prøver at logge ind igen.

Use case: Tilføj svømmere til konkurrence

Brugeren åbner programmet og logger ind, efterfølgende vælger han print resultater og får en liste af top 5 inden for hver disciplin. Ud fra de resultater han har fået vælger han hvilke svømmere han vil tilføje til konkurrencen.