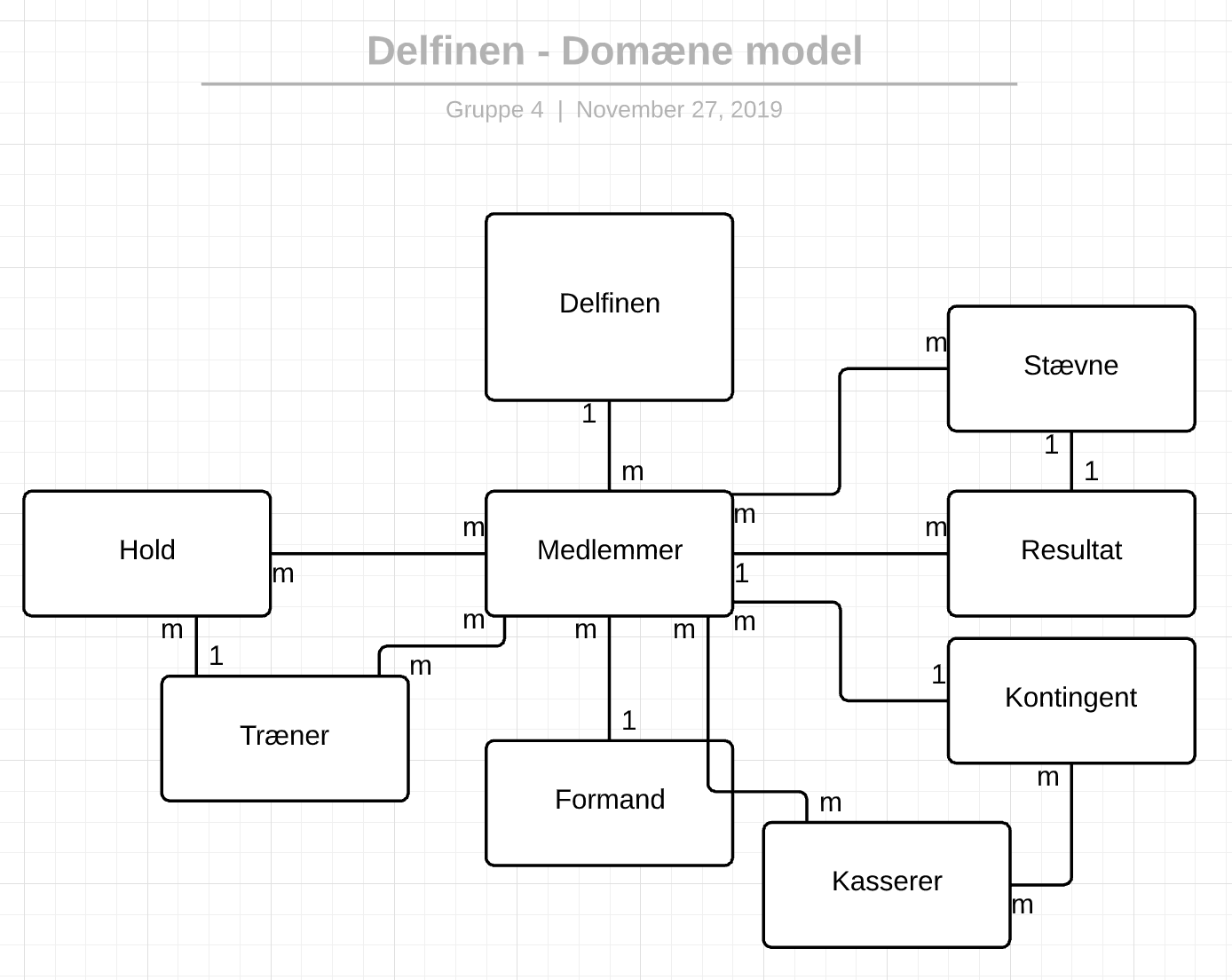
Delefinen

# 1.0 Kravspecifikation

I dette afsnit beskrives de specifikke krav, stillet af kunden, til det stykke software der skal udvikles. I dette tilfælde omhandler det svømmeklubben Delfinen, som efterspørger et system til at administrere medlemmer, kontingenter m.m.

## Domæne model

Nedenfor ses det første udkast til modelleringen af kundens domæne (figur 1).



Figur 1

## Use Case diagram

På nedenstående diagram (figur 2), visualiseres alle de ’use cases’ som der abstraheres fra beskrivelsen af det ønskede stykke software.

Et billede, der indeholder tekst, kort

Automatisk genereret beskrivelse

Figur 2

## Use case diagram beskrivelse

I dette afsnit beskrives de individuelle ’use cases’ som tidligere visualiseret (figur 2).

### 1.2.1 Aktøre

I et ’use case’-diagram forholder man sig til hvordan hvilke aktører agerer med produktet. For svømmeklubben, Delfinen, er der tale om en formand, en kasser og en træner. De interagerer på hver deres måde med programmet, beskrevet i use-cases.

### Use-cases

Hvert use-case beskriver en specifik handling eller funktionalitet i programmet. Hos Delfinen vil man have at programmet har følgende funktioner, beskrevet som use-cases.

|  |  |
| --- | --- |
| Aktør(er) | Use-case |
| Formand | **Nye medlemmer**  Opret nye medlemmer I systemet.  **Preconditions**  Logget ind.Kontakt til database.  **Flow of events**   1. Aktøren vælger at tilføje et nyt medlem. 2. Aktøren indtaster stam-data for det nye medlem (navn, alder, inaktiv/aktiv m.m.). 3. Aktøren vælger at oprette det nye medlem. 4. Programmet viser en oversigt over indtastede data, og beder om bekræftelse fra aktøren. 5. Programmet gemmer informationen i databasen. 6. Programmet bekræfter at data er blevet gemt.   **Alternative flow**  **2a.** Forket data-type indtastet.   * Programmet viser fejl-meddelelse og beder om nyt input.   **4b.** Identiske data med lign. Bruger, indtastet.   * Programmet beder om bekræftelse af oprettelse af denne bruger.   **6a.** Kan ikke gemme data.   * Programmet viser en fejl-meddelelse og beder bruger om at prøve igen. |
| Kasser | **Administrer kontingent**  Ændre kontingent størrelse via. **Extends ’Ændre kontingent’**.  **Preconditions**  Logget ind.  Kontakt til database.  **Flow of events**   1. Aktøren vælger at ændre størrelsen på et givent kontingent. 2. Aktøren indtaster en ny værdi som kontingentet skal tilrettes. 3. Programmet beder om bekræftelse fra brugeren af det indtastede. 4. Programmet gemmer informationen i databasen. 5. Programmet bekræfter at data er blevet gemt.   **Alternative flow**  **2a.** Aktøren indtaster en ugyldig værdi   * Programmet viser en fejl-meddelelse og beder om nyt input.   **5a.** Kan ikke gemme data.   * Programmet viser en fejl-meddelelse og beder bruger om at prøve igen. |
| Kasser | **Vis restence**  Aktøren skal kunne se hvilke medlemmer der er i restance.  **Preconditions**  Logget ind.  Kontakt til database.  **Flow of events**   1. Programmet tjekker betalingsdato mod medlemmers seneste betaling. 2. Programmet viser en oversigt over medlemmer i restance.   **Alternative flows**  N/A. |
| Træner | **Registrer resultat**  Aktøren kan registrere et givent resultat.  **Preconditions**  Logget ind.  Kontakt til database.  **Flow of events**   1. Programmet spørger hvilket medlem der skal registreres resultat på. 2. Aktøren vælger medlem. 3. Programmet beder om indtastning af typen og resultatet. 4. Aktøren vælger om det er et trænings- eller stævne-resultat. 5. Aktøren vælger svømmedisciplinen. 6. Aktøren indtaster det konkrete resultat. 7. Programmet beder om bekræftelse fra brugeren af det indtastede. 8. Programmet gemmer informationen i databasen. 9. Programmet bekræfter at data er blevet gemt.   **Alternative flows**  **4a.** Aktør vælger stævne-resultat.   * Bruger indtaster dato og navnet på stævnet.   **9a.** Kan ikke gemme data.   * Programmet viser en fejl-meddelelse og beder bruger om at prøve igen. |
| Træner | **Vis top 5**  Aktøren vælger at få vist top 5 af medlemmerne, inden for hver svømmedisciplin.  **Preconditions**  Logget ind.  Kontakt til database.  **Flow of events**   1. Programmet spørger hvilken disciplin der skal vises top 5 for. 2. Aktøren vælger disciplin. 3. Programmet sammenligner medlemmers træningsresultat inden for den givne svømmedisciplin. 4. Programmet viser top 5 af den specifikke disciplin.   **Alternative flows**  **2a.** Bruger vælger ugyldig disciplin.   * Programmet viser en fejl-meddelelse og beder om nyt input. |
|  |  |

## Klasse diagram

Et billede, der indeholder tekst

Automatisk genereret beskrivelse