

# Webová aplikácia pre kanoistický klub

## Špecifikácia projektu

### 1. Popis aplikácie

V športových kluboch je potrebné spravovať, organizovať a plánovať viaceré oblasti. Pri väčších športových kluboch toho všetkého môže byť na jedného človeka, najčastejšie trénera, popri ďalších povinnostiach priveľa, či už z časového hľadiska alebo organizačného.

Kanoistika ako každý iný šport má svoje špecifiká, ktoré je nutné v klube riešiť. Tak ako v každom inom športe, je nutné organizovať tréningy, preteky alebo súťaže a trénovať športovcov. Špeciálne treba v kanoistických kluboch riešiť kanoistickú výstroj: kajaky, kanoe, pádla, vesty... Tréningy treba tiež riešiť osobitne, keďže niektoré sa konajú v posilňovni, niektoré vonku a iné na jazere.

V bežnom kanoistickom klube sa nachádza viacero osôb, ktoré vykonávajú rôzne činnosti. Najväčšou skupinou sú športovci patriaci do klubu, ktorí v klube trénujú a zúčastňujú sa pretekov. Ďalšou a asi najdôležitejšou osobou v klube je tréner, ktorý organizuje tréningy a požičiava výstroj. Dôležitá je aj osoba, ktorá ma na starosti opravu lodí a výstroje.

Aby sa všetkým členom kanoistického klubu zefektívnila práca a mohli sa venovať, čo najviac trénovaniu a čo najmenej sa museli zaťažovať správou tréningov a vecami okolo, navrhujem webovú aplikáciu, ktorá by mala jednotlivé oblasti okolo kanoistiky zjednodušiť a spojiť na jedno miesto.

### 2. Používateľské požiadavky

#### I. Tréner

- vedie klub
- plánuje tréningy, dátumovo, obsahovo, lokáciou
- akceptuje/odoberá členov klubu
- vyberá členský poplatok

#### II. Člen klubu (zverenec)

- prihlasuje sa do klubu
- absolvuje tréningy, zapisuje sa na ne
- zapožičiava si výstroj

#### III. Technický správca lodí

- spravuje lode, pridáva odoberá ich
- opravuje poškodené lode

## IV. Administrátor webu

- môže upravovať nastavenia alebo dáta aj ostatných používateľov
- má práva všetkých ostatných používateľov

### User stories

Ako ktorákoľvek roľa, chcem sa vedieť prihlásiť do účtu, aby som vedel iba ja alebo administrátor spravovať moje dáta.

Ako tréner, chcem vedieť plánovať tréningy tak, že zverenci v klube budú tréningy vidieť a budú sa na ne môcť prihlásiť, aby som mal v tréningoch poriadok.

Ako tréner, chcem vedieť, spravovať členov klubu, aby som vedel, koľko má klub členov a informácie o nich.

Ako tréner, chcem vedieť koľko a akých tréningov (posilňovňa/plaváreň....) každý člen klubu absolvoval, aby som vedel, kto môže ísť na preteky.

Ako tréner, chcem mať prehľad o tom, ktorý členovia ešte nezaplatili celé členské a chcem si to vedieť zaznačiť.

Ako člen klubu (zverenec), chcem vedieť kedy sú tréningy, aby som vedel, kedy a kam mám prísť respektíve vedel sa na daný tréning prihlásiť.

Ako člen klubu, si chcem požičiavať výstroj, aby som mal loď s ktorou môžem trénovať na tréningoch na jazere.

Ako technický správca klubu chcem mať prehľad o lodiach v klube, aby som vedel, ktoré je potrebné opraviť alebo zakúpiť nové.

Ako administrátor webu, chcem vedieť upravovať informácie na stránke a používateľov, aby som mohol pomôcť s chodom webu.

## 3. Dátový model

### I. Tabuľka účtov

user id, používateľské meno, oprávnenia/rola (tréner, zverenec, správca lodí, administrátor), heslo, soľ

## **II. Tabuľka lodí**

id lode, rok výroby, kajak/kanoë, veľkosť lode (maximálna váha športovca), mini (je určený pre deti), popis poruchy (optional), váha lode, user id (kto si ho požičiava)

## **III. Tabuľka členov**

user id, meno, priezvisko, vek, výška, váha, súťažná kategória, kajakár/kanoista, zaplatené členské

## **IV. Tabuľka tréningov**

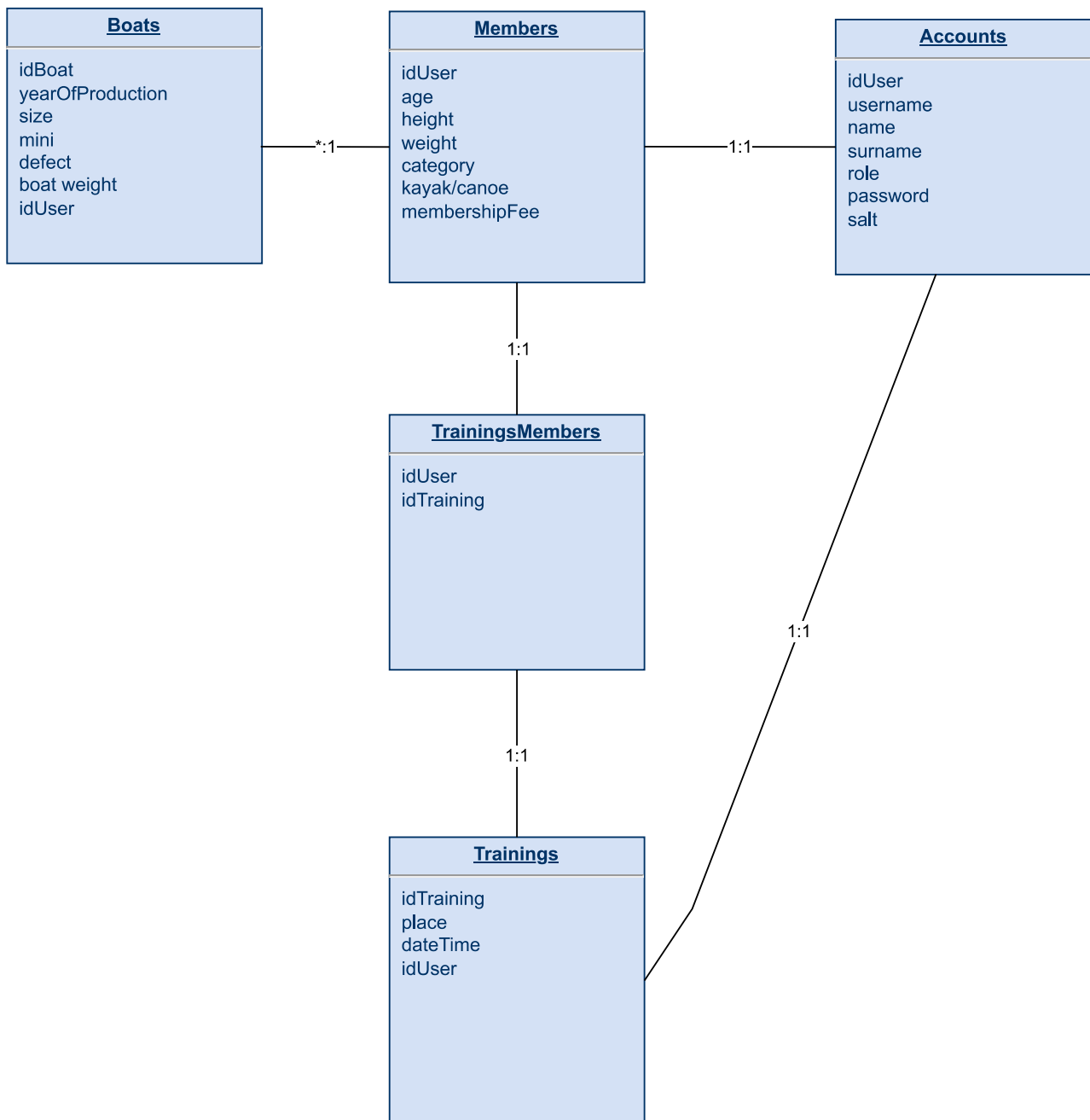
miesto (adresa/ budova), typ tréningu (jazero, posilňovňa, plaváreň, pádlovacie trenažéry), čas, user id (tréner na danom tréningu, ktorý tréning vytvára), tréning id

## **V. Tabuľka spájajúca tréningy a členov**

user id, tréning id

popisuje účastníkov na tréningu

## Vizualizácia tabuliek a vzťahov



## 4. Technologické požiadavky

Na backende plánujem použiť Flask verziu 2.2.3 s pluginom Flask-Login. Ako databázu plánujem použiť MySQL verziu 57.41. Flask bude komunikovať s frontendom pomocou REST API. Na frontende plánujem využiť framework React (18.2.0) a na design stránky chcem použiť Bootstrap (5.3). Aplikácia by mala byť Single Page App a renderovanie by malo byť na strane Reactu, teda použil by som Client Side Rendering. Hosting plánujem použiť pythonanywhere. Z prehliadačov by aplikácia mohla podporovať Chrome a Firefox, keďže plánujem použiť bootstrap niektoré časti stránky by mohli byť vhodné aj pre mobilné zariadenia.

## 5. Časový plán

### 1. Týždeň

Stiahnutie MySQL databázy a jej spustenie a naštudovanie. 2 hodiny

Nainštalovanie Flasku, Reactu, Bootstrapu ich prepojenie aj s DB. 7 hodín

### 2. Týždeň

Preštudovať základy Flasku, prvé REST API. 4 hodiny

Test deploymentu na pythonanywhere. 4 hodiny

### 3. Týždeň

Naštudovanie Flask login a spravenie základného loginu + inicializovať tabuľku účtov. 6 hodín

Základ UI s bootstrapom a spolupráca s Reactom. 2 hodiny

Naštudovanie Reactu + prvé React UI pre login. 3 hodiny

### 4. Týždeň

Inicializovanie tabuliek + obrázkov. 3 hodiny

REST API pre lode. 4 hodiny

### 5. Týždeň

UI v Reacte pre lode. 5 hodín

Deployment na pythonanywhere. 1 hodina

Beta report, finalizácia beta verzie. 2 hodiny

Začať REST API pre tréningy a členov 2 hodiny

### 6. Týždeň

React UI pre tréningy a správu členov. 6 hodín

Dokončenie REST API pre tréningy a členov. 4 hodiny

### 7. Týždeň

Dokončenie chýbajúcich REST API vo Flasku a frontendu v Reacte. 4 hodiny

Dotiahnutie nestihnutých vecí, chýb a problémov. 4 hodiny

Deployment na pythonanywhere. 1 hodina

Final report. 1 hodina

## **6. Future Work**

Ďalšími užitočnými možnosťami by na stránke mohli byť správa pretekov, kto sa kedy zúčastní, na ktorých pretekoch, alebo záznam výkonov na tréningoch, napríklad počet zhybov alebo trvanie 200, 500 a 1000 metrových úsekov. Určite existuje ešte mnoho ďalších funkcionalít, ktoré by bolo možné do webovej aplikácie pridať, ktoré vyplynú z praxe.