Gym Fitness App

Document de analiza si design

Studenti:

* Colceriu Tudor
* Fîrte Cătălin Emanuel
* Souca Natalia Elena
* Vesa Bianca

**Grupa: 30237**

**Contents**

[I Specificatia proiectului 3](#_Toc59542811)

[1.1 Diagrama Domain Model 4](#_Toc59542812)

[II Modelul Use-Case 4](#_Toc59542813)

[2.1 Utilizatori si stakeholderi 5](#_Toc59542814)

[2.2 Identificare Use-Case-uri 5](#_Toc59542815)

[2.3 Diagrama UML Use-Case 5](#_Toc59542816)

[III Design arhitectural 9](#_Toc59542817)

[3.1 Arhitectura conceptuala 9](#_Toc59542818)

[3.2 Diagrama bazei de date 10](#_Toc59542819)

[IV Specificatii suplimentare 11](#_Toc59542822)

[4.1 Specificatii non-functionale 11](#_Toc59542823)

[4.2 Constrangeri de design 11](#_Toc59542824)

[V Testare 12](#_Toc59542825)

[5.1 Testarea functionalitatii aplicatiei 12](#_Toc59542826)

[5.2 Dezvolatari ulterioare 24](#_Toc59542827)

[VI Bibliografie 24](#_Toc59542828)

# 

# I Specificatia proiectului

Design-ul si implementarea unei aplicatii pentru gestionarea unei sali de fitness.

Aplicatia va avea trei tipuri de utilizatori(client, antrenor, administrator).

Clientul va putea realiza urmatoarele operatii:

-Logarea si inregistrare in aplicatie

-Efectuarea unei programari la antrenamentul dorit si ora dorita

-Creare abonament

-Evaluarea antrenorului clasei la care a participat

-Vizualizare si modificare profil personal

Antrenorul va putea realiza urmatoarele operatii:

-Logarea in aplicatie

-Vizualizare si modificare profil personal (nume, tipuri de antrenamente pe care este specializat, rating-ul primit)

-Vizualizare si modificare orar clase si programari

Administratorul va putea realiza urmatoarele operatii:

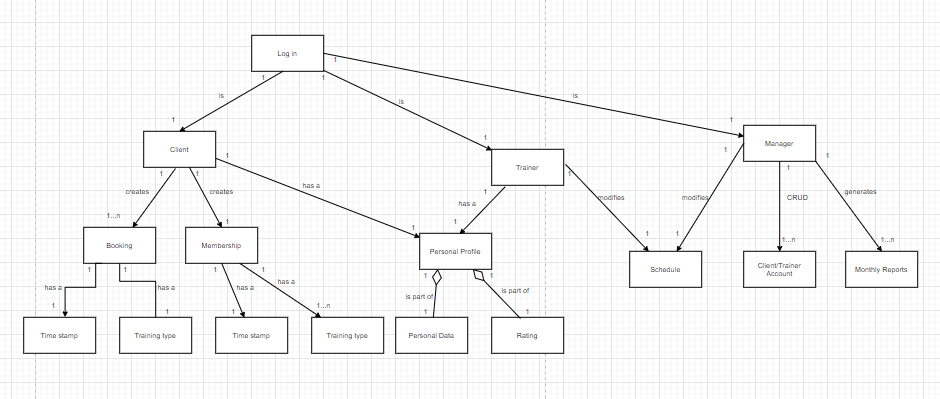
-CRUD antrenori

-Vizualizare si modificare orar clase si programari

-Vizualizare si modificare profil personal

-Generare rapoarte lunare cu cheltuieli

## 1.1 Diagrama Domain Model

**

# II Modelul Use-Case

In ingineria software o diagrama Use Case reflecta interactiunile din lumea reala intre utilizatori si sistem, precum si interactiunile intre diferitele tipuri de utilizatori. Ea modeleaza contextul in care sistemul trebuie privit, cerintele sale, influentele interne, cat si externe pe care acesta le are, dar si obiectivele pe care le indeplineste.

Un actor este reprezentarea unei fiinte din lumea reala, care relationeaza cu sistemul, iar scenariul este actiunea propriu-zisa, pe care acesta o executa, generand un rezultat.

## 2.1 Utilizatori si stakeholderi

Utilizatorii acestei aplicatii sunt:

* Clientul, care reprezinta o persoana obisnuita ce participa la activitatile salii
* Antrenorul, care va fi un angajat al salii
* Managerul, adica persoana care detine sala de fitness

Stakeholderii aplicatiei vor fi:

- Antrenorul si managerul

## 2.2 Identificare Use-Case-uri

Urmatoarele use-case-uri sunt reprezentative pentru aceasta aplicatie:

1. Nume Use Case: Creare cont

Nivel: User-Goal

Actor Principal: Client simplu

Scenariu de success: In fereastra de login utilizatorul apasa pe prompt-ul de inregistrare, sub butonul de login (un mesaj). Fereastra de register va fi deschisa pentru ca utilizatorul sa poata furniza aplicatiei datele necesare. Fereastra contine campuri cu anumite etichete pentru a-i sugera utilizatorului ce date sa introduca in fiecare camp. Dupa completarea tuturor casutelor necesare, se va efectua click pe butonul de register, moment in care aceasta fereastra se va inchide iar utilizatorul este intors la fereastra de login.

Extensie:

E1: Pop-up cu mesaj de succes, in cazul in care toate datele au fost introduse corect de catre utilizatorul ce doreste sa isi faca un cont in aplicatie

E2: Mesaj de eroare

E2.1: Daca utilizatorul introduce pentru credentiale un nume ce este deja existent la alt utilizator, atunci un mesaj de eroare va fi primit

E2.2: Daca utilizatorul va lasa unul din campurile de completat liber, atunci un mesaj de eroare va fi prezentat, in momentul in care acesta doreste sa se inregistreze

## Nume Use Case: Creare abonament

## Nivel: User-Goal

## Actor principal: Client simplu

## Scenariu de succes: Clientul se logheaza in contul sau utilizand credentialele furnizate de catre acesta in momentul inregistrarii. Un buton de creare abonament este prezent in fereastra cu profilul sau. In urma unui click, clientului ii va fi prezentata o alta fereastra in care poate selecta tipul de abonament. Toate informatiile despre tipurile disponibile de abonamente sunt prezente si vizibile in aceasta fereastra. Se executa click pe unul din cele 3 tipuri posibile de abonament (Silver, Gold, Platinum), iar dupa ce se confirma dorinta (click pe ok in fereastra ce apare pentru confirmare), clientului ii va fi atasat un abonament de o luna din momentul efectuarii acestei operatii.

## Extensie:

## E1 : Daca abonamentul a fost creat cu succes, clientul va putea sa isi faca un booking, sau sa dea click pe butonul de vizualizare a booking-urilor existente deja (daca abonamentul e proaspat, nu va gasi nimic acolo, insa nu va primi eroare la operatie)

## 

## Nume Use Case: Creare clasa de fitness

## Nivel: User-Goal

## Actor principal: Antrenor

## Scenariu de succes: Antrenorul intra in contul facut de catre manager, folosind numele si parola in fereastra de login. Pentru a crea o clasa, acesta da click pe butonul de Add Schedule, care apoi il duce la o fereastra pentru efectuarea operatiei dorite. Antrenorul va introduce in campurile prezentate in aplicatie toate datele necesare (vizibil in label-urile de langa fiecare camp). Dupa ce informatiile au fost deja completate, se va face click pe butonul de OK, iar in cazul in care datele introduse au fost corecte, fereastra nu va afisa mesaje de eroare, iar operatia este realizata cu succes.

## Extensie:

## E1 : Mesaj de eroare

## E1.1: Clasa deja existenta in acel moment (cel introdus pentru data si ora clasei ce se doreste sa fie facuta)

## E1.2: Clasa nu corespunde cu certificarea antrenorului (un antrenor de yoga nu poate tine clase de fitness)

## E1.3: Id-ul antrenorului (cel introdus manual de acesta, pe care doar el si-l cunoaste) nu este corect

## E1.3.1: Evitarea situatiei in care un antrenor ii face altuia o clasa, fara ca acesta sa stie acest lucru.

## E2 : Mesaj de success, clasa a fost creata

## Nume Use-Case: Generare rapoarte lunare

## Nivel: Summary

## Actor principal: Manager

## Scenariu de succes: Administratorul intra in contul sau, folosind userul si parola unice pentru administrator. In fereastra de profil, efectueaza click pe butonul de vizualizare a rapoartelor ("View Reports"). O fereatra noua este prezentata, in care administratorul poate alege din 2 casute de drop-down luna si anul pentru care doreste sa vada rapoartele lunare. Dupa selectare, acesta face click pe butunul de Generate, iar rapoartele sunt gasite in format PDF.

## Extensie:

## E1: Mesaj de eroare

## E1.1: Luna selectata incorect (inca nu s-a ajuns in acel moment de timp, in care luna si anul selectate sa fie date anterioare celei curente)

## E2: Succes in generare

## E2.1: Un raport in format PDF este generat in folderul cu aplicatia, in care se afla detaliile cerute in urma interogarii aplicatiei (informatii despre numarul de clienti si abonamente facute, costuri totale)

## 2.3 Diagrama UML Use-Case

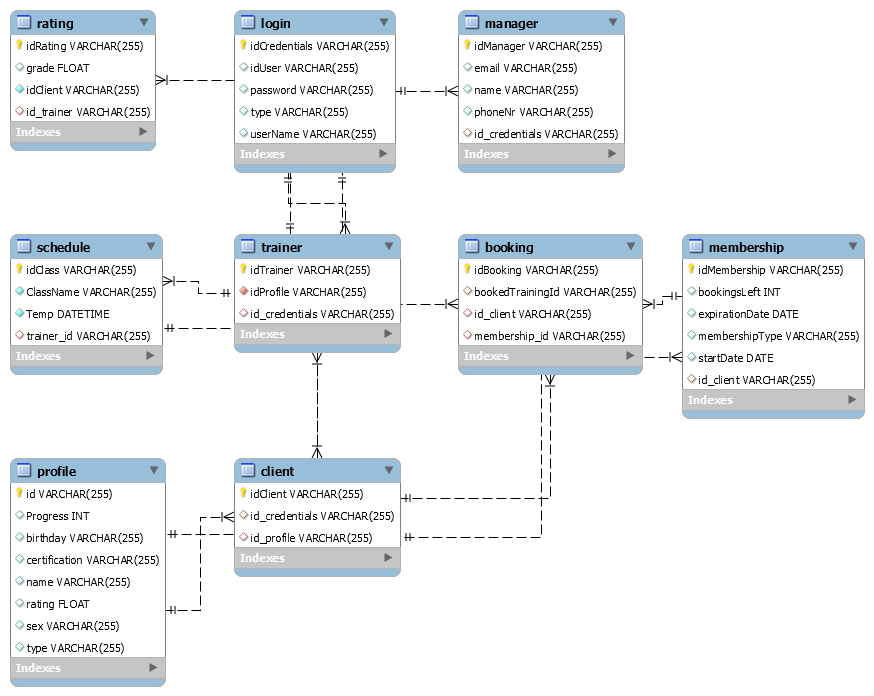
# 

# III Design arhitectural

## 3.1 Arhitectura conceptuala

Proiectul “Gym Fitness App” este o aplicatie desktop, care relationeaza cu o baza de date. Aceasta baza de date pastreaza informatiile legate de cele trei tipuri de utilizatori (Client, Antrenor si Manager). Pentru implementarea proiectului am ales o arhitectura de tip layered. Cele trei componente principale ale acestei arhitecturi sunt : persistence layer, service layer si view-controller layer. Persistence Layer tine legatura intre aplicatie si baza de date, Service Layer prelucreaza datele existente si cele care se vor a fi adaugate, iar View-Controller reprezinta interfata prin care utilizatorul comunica cu aceasta aplicatie.

## 3.2 Diagrama bazei de date

**

# IV Specificatii suplimentare

In cele ce urmeaza vom prezenta specificatiile suplimentare ale aplicatiei, precum si constrangerile pe care le impune.

## 4.1 Specificatii non-functionale

O specificatie non-functionala existenta este timpul de raspuns al aplicatiei. Datorita faptului ca am utilizat pentru implementare tool-ul Hibernate, care comunica in timp real cu baza de date, intervine o intarziere in ceea ce priveste comunicarea utilizatorului cu baza de date prin intermediul interfetei grafice.

O alta constrangere exterioara se manifesta prin obligatia respectarii sigurantei datelor cu caracter personal, cat si a circulatiei lor in cadrul aplicatiei. Toate aceste aspecte sunt prevazute de catre ANSPDCP (UE – 2016/679).

Adaptabilitatea nu este indeplinita datorita faptului ca atat baza de date cat si codul sursa sunt interdependente, iar modificarea uneia nu se propaga si la nivelul celeilalte.

Aplicatia este extensibila, avand posibilitatea de a adauga functionalitati noi (mentionate in capitolul 5.2). De asemenea, functionalitatile deja existente se pot modifica cu usurinta.

## Constrangeri de design

In ceea ce priveste performanta arhitecturii, putem afirma faptul ca legatura intre administrator si utilizatori este mentinuta constant. Atat in cazul indeplinirii cu succes a scenariilor de utilizare, cat si in caz de erori, acestea sunt pastrate in detaliu prin intermediul mesajelor de tip log si a exceptiilor tratate. De asemenea, utilizatorul primeste feedback din partea aplicatiei la fiecare pas dupa executarea unei operatii.

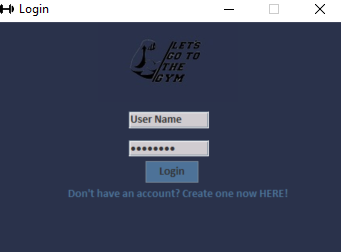
# V Testare

In acest capitol vom exemplifica utilizarea aplicatiei, vazuta prin prisma fiecarui tip de utilizator disponibil.

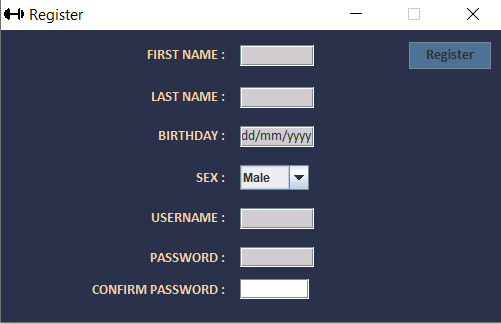
## 5.1 Testarea functionalitatii aplicatiei

Un exemplu de utilizare a aplicatiei pentru crearea unui cont de client:

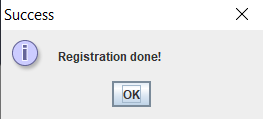
Pentru a crea un cont se selecteaza “Don’t have an account? Create one now HERE!”.



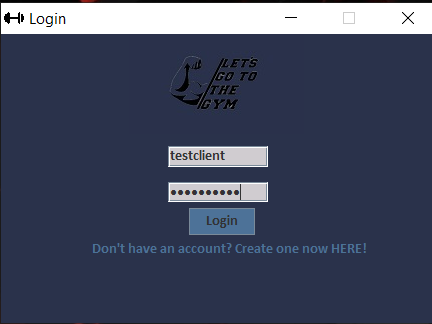
Dupa completarea datelor se apasa butonul Register.



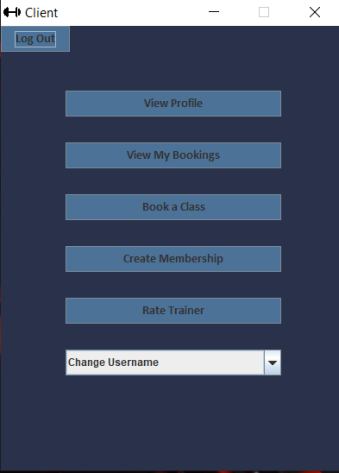
In cazul in care datele introduse sunt valide, va aparea un mesaj de success.



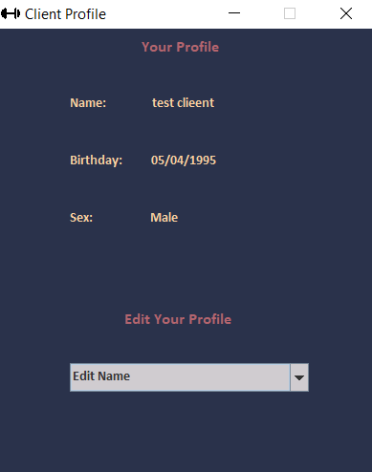
Se revine la pagina de Login unde se introduc credentialele contului si se apasa butonul Login.



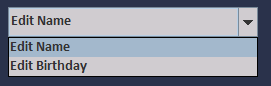
Pentru a vedea detaliile profilului se apasa butonul View Profile.



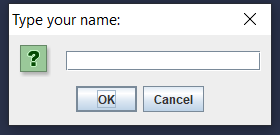
Pentru a-si modifica numele in cazul in care acestea s-au introdus gresit la inregistrare, clientul va apasa pe sageata corespunzatoare optiunii de Edit Profile.



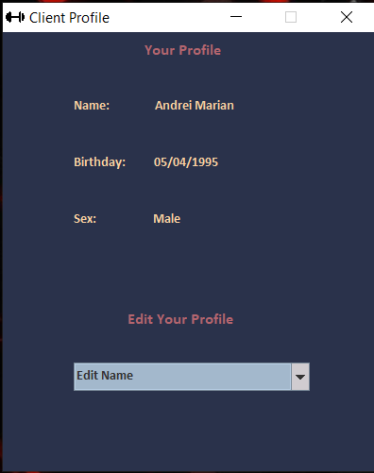
Vor aparea cele doua optiuni posibile de modificare a profilului, selectand-o pe cea de Edit Name.



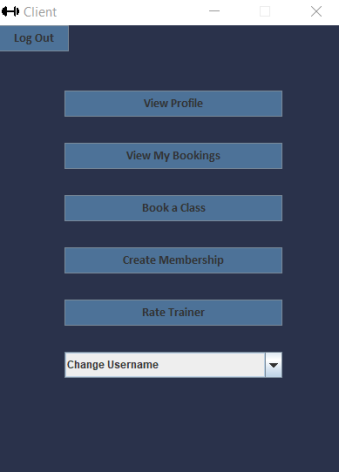
Dupa introducerea numelui, se apasa butonul OK.



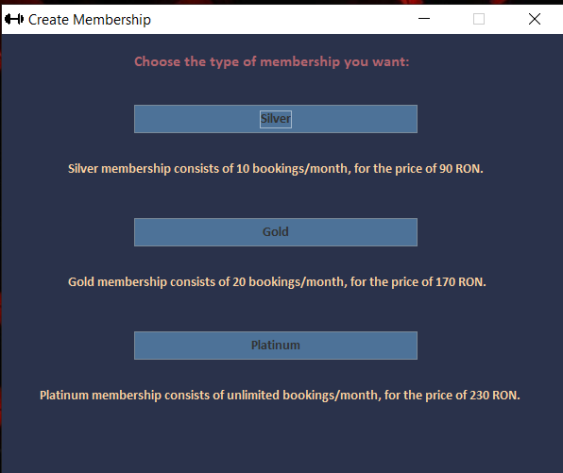
Daca numele este valid, noul nume va aparea in pagina profilului.



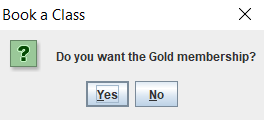
Pentru a crea un abonament, pe baza caruia sa se faca rezervari, se apasa butonul Create Membership.



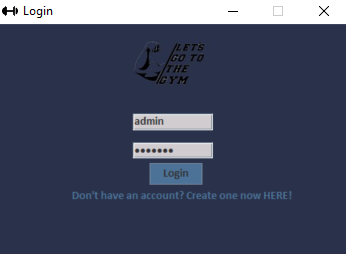
Fereastra Create Membership se va deschide, unde se alege tipul de antrenament dorit, care are detaliile mentionate mai jos.



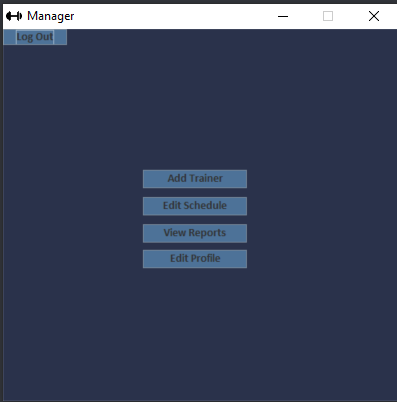
In cazul in care se alege abonamentul de tip Gold, urmatorul mesaj va aparea. La apasarea butonului Yes abonamentul de tip Gold va fi creat.



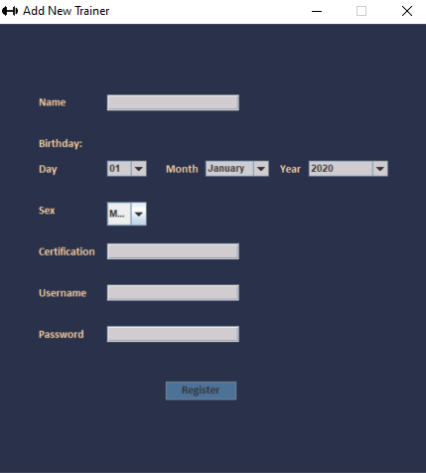
Pentru a intra in contul de manager, se porneste de la fereastra Login, unde se introduce credentialele contului de manager.



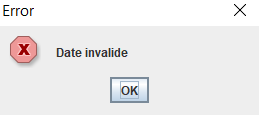
Dupa apasarea butonului Login, contul managerului se va deschide, deci se poate crea un cont de antrenor, selectand butonul Add Trainer.



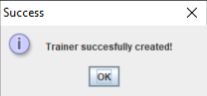
In fereastra care se deschide se completeaza datele personale ale antrenorului, iar apoi se apasa butonul Register.



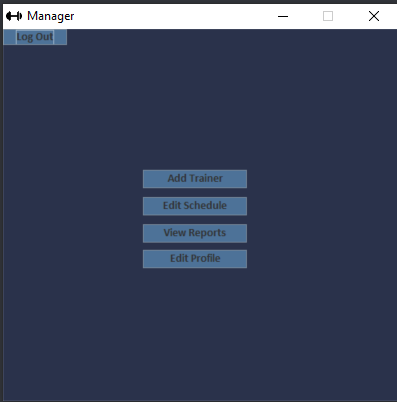
In cazul in care datele introduse nu sunt valide, un mesaj de acest tip va aparea.



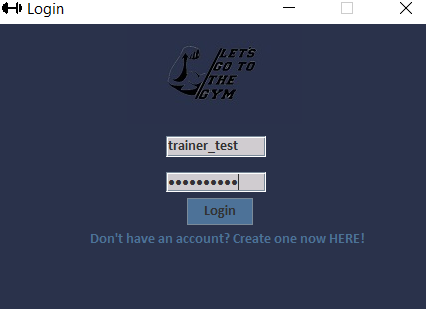
In caz de success, acest mesaj va aparea.



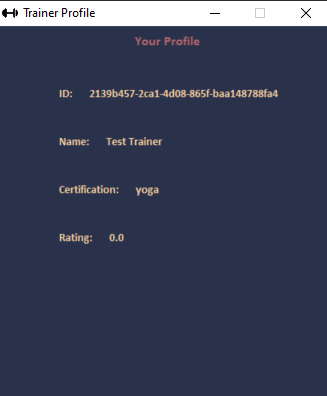
Pentru a intra in contul antrenorului, se apasa butonul Log Out, care revine la fereastra Login.



Se introduc credentialele antrenorului.



Dupa apasarea butonului View Profile, se pot observa datele personale ale antrenorului care au fost introduse corect.



## 5.2 Dezvoltari ulterioare

Ca dezvoltari ulterioare ale acestui proiect se merita mentionate urmatoarele functionalitati:

* Generare bon fiscal si trimiterea lui la mail-ul clientului odata ce s-a creat un abonament la o clasa de sport
* Stergerea din baza de date a abonamentelor care au expirat pana la data curenta
* Posibilitatea de conectare la aplicatie printr-o logica de “forgot password” in cazul in care utilizatorul nu isi mai aminteste parola contului
* Adaugarea unei logici de concediere a antrenorilor, cu dezactivarea contului si imposibilitatea de a-l mai accesa
* Posibilitatea de a face upgrade la un abonament deja existent, intr-unul care ofera mai multe rezervari la clasele de sport

# VI Bibliografie

<https://www.baeldung.com/hibernate-entitymanager>

<https://www.tutorialspoint.com/hibernate/hibernate_tutorial.pdf>

<https://www.baeldung.com/jpa-cascade-types>