

# *Platformă integrată de management al sălilor fitness*

## **Documentul de proiectare**

---

# Cuprins

## 1. Introducere1

### 1.1 Scopul documentului1

## 2. Prezentare generală și abordări de proiectare2

### 2.1 Prezentare generală2

### 2.2 Presupuneri/ Constrângeri/ Riscuri2

#### 2.2.1 Presupuneri2

#### 2.2.2 Constrângeri4

#### 2.2.3 Riscuri4

## 3. Considerații de proiectare6

### 3.1 Obiective și linii directoare (ghiduri)6

### 3.2 Metode de dezvoltare6

### 3.3 Strategii de arhitectură6

## 4. Arhitectura Sistemului și Proiectarea Arhitecturii8

### 4.1 Vedere logică9

### 4.2 Arhitectură hardware9

### 4.3 Arhitectură software10

### 4.4 Arhitectura informațiilor10

### 4.5 Arhitectura de comunicații interne**Error! Bookmark not defined.**

### 4.6 Diagrama de arhitectură a sistemului11

## 5. Proiectarea sistemului12

### 5.1 Proiectarea bazei de date12

#### 5.1.1 Obiecte de date și structuri de date rezultante12

#### 5.1.2 Fișiere și baze de date**Error! Bookmark not defined.**

### 5.2 Conversii de date**Error! Bookmark not defined.**

### 5.3 Interfețe utilizator13

5.3.1	Intrări	14
5.3.2	Ieșiri	15
5.4	Proiectarea interfețelor cu utilizatorul	16
<b>6.</b>	<b>Scenarii de utilizare</b>	<b>17</b>
<b>7.</b>	<b>Proiectare de detaliu</b>	<b>18</b>
7.1	Proiectare hardware de detaliu	18
7.2	Proiectare software de detaliu	19
7.3	Proiectare detaliată de securitate	19
7.4	Proiectare de detaliu pentru performanța sistemului	20
7.5	Proiectare detaliată a comunicațiilor interne (între componente)	21
<b>8.</b>	<b>Controale pentru verificarea integrității sistemului</b>	<b>21</b>
<b>Anexa A: Gestiunea modificărilor documentului</b>		<b>22</b>
<b>Anexa B: Acronime</b>		<b>23</b>
<b>Anexa C Documente la care se face referire</b>		<b>24</b>

# 1. Introducere

---

**Numele proiectului:** No Limits - Platformă de Fitness și Sănătate

**Acronim:** NLFS

Evoluția documentului: Documentul de Proiectare a Sistemului marchează o etapă crucială în dezvoltarea proiectului No Limits, tranziționând de la cerințele funcționale și non-funcționale la specificațiile tehnice de proiectare a sistemului. Este o prezentare detaliată a modului în care cerințele identificate în Documentul de Cerințe sunt transformate într-o arhitectură detaliată și specificații tehnice pentru implementarea sistemului.

Considerații referitoare la securitate și confidențialitate: În timpul proiectării sistemului, se acordă o atenție deosebită aspectelor de securitate și confidențialitate, asigurând protejarea datelor utilizatorilor și a informațiilor sensibile privind sănătatea. Aceasta include implementarea măsurilor adecvate pentru protejarea datelor personale și respectarea reglementărilor privind confidențialitatea.

## 1.1 Scopul documentului

Scopul acestui Document de Proiectare a Sistemului (SDD) este de a oferi o descriere detaliată a arhitecturii și designului sistemului No Limits, platformă dedicată fitness-ului și sănătății. Acesta are rolul de a ghida echipa de dezvoltare în procesul de implementare a sistemului, asigurând înțelegerea clară a cerințelor și specificațiilor de proiectare.

Documentul urmărește să ofere o viziune comprehensivă asupra arhitecturii sistemului și a componentelor sale, evidențiind modul în care cerințele funcționale și non-funcționale sunt traduse într-un design tehnic. Publicul țintă al acestui document include managerii de proiect, echipa de proiect și membrii echipei de dezvoltare, care vor utiliza informațiile prezentate pentru a ghida procesul de implementare.

Pe parcursul ciclului de dezvoltare a sistemului, acest document va fi actualizat și completat în mod incremental, în funcție de evoluția proiectului și de cerințele specifice ale fiecărei etape.

## 2. Prezentare generală și abordări de proiectare

---

### 2.1 Prezentare generală

Sistemul "No Limits Gym" este o aplicație web dedicată promovării sănătății și fitnessului, oferind utilizatorilor o platformă interactivă pentru gestionarea rutinei lor de antrenament și nutriție. Scopul său este de a facilita accesul la informații utile despre fitness, de a oferi planuri personalizate de antrenament și nutriție, și de a motiva utilizatorii să își atingă obiectivele de sănătate și fitness.

### 2.2 Presupuneri/ Constrângeri/ Riscuri

#### 2.2.1 Presupuneri

Pentru a asigura implementarea și funcționarea eficientă a aplicației "Digital Fitness Assistant", există o serie de presupuneri și dependențe care trebuie considerate. Acestea includ aspecte tehnologice, caracteristicile utilizatorilor și flexibilitatea în adoptarea noilor funcționalități sau tehnologii

##### 1.Hardware și Software Asociat:

Se presupune că utilizatorii vor avea acces la dispozitive compatibile cu internetul (smartphone-uri, tablete, laptopuri).

Sala de fitness va necesita hardware adecvat pentru rularea backend-ului aplicației, inclusiv servere fizice sau virtuale.

##### 2.Sisteme de Operare:

Aplicația va fi optimizată pentru cele mai populare sisteme de operare, inclusiv iOS, Android pentru dispozitive mobile și Windows, macOS pentru desktop.

Dependența de compatibilitatea cu versiunile curente și populare ale acestor sisteme de operare este crucială.

##### 3.Conexiune la Internet:

Funcționalitatea completă a aplicației depinde de o conexiune stabilă și rapidă la internet, atât pentru utilizatori cât și pentru infrastructura backend.

#### **4.Caracteristicile Utilizatorilor Finali:**

Se presupune că utilizatorii au competențe digitale de bază necesare pentru navigarea și utilizarea eficientă a aplicației.

Utilizatorii ar trebui să fie familiarizați cu conceptele de bază ale rezervărilor online și gestionării conturilor prin intermediul aplicațiilor web.

#### **5.Software de Dezvoltare și Management al Proiectului:**

Dezvoltarea se bazează pe disponibilitatea continuă a unor framework-uri și limbaje de programare specifice, cum ar fi React pentru frontend și Node.js sau Python pentru backend.

Se presupune disponibilitatea unor tool-uri pentru colaborare și versionare, cum ar fi Git, GitHub și sisteme de tracking al eroilor.

#### **6.Securitate:**

Dependința de soluții moderne de securitate pentru protecția datelor, inclusiv firewalls, sisteme de detecție și prevenire a intruziunilor, și protocoale de criptare.

Presupunem că utilizatorii vor respecta recomandările de securitate oferite prin aplicație pentru protejarea datelor lor personale.

#### **7.Modificări ale Funcționalității:**

Aplicația trebuie să fie suficient de flexibilă pentru a permite adăugarea de noi funcționalități sau modificarea celor existente în funcție de feedback-ul utilizatorilor și de schimbările din industria de fitness.

Dependența de o abordare modulară în design-ul software pentru a facilita actualizările și scalabilitatea.

## 2.2.2 Constrângeri

### 1. Mediu Hardware și Software

- **Constrângere:** Aplicația trebuie să fie compatibilă cu dispozitivele și sistemele de operare pe care le utilizează majoritatea clienților și personalului sălii de fitness.

### 2. Mediu Utilizator Final

- **Constrângere:** Utilizatorii pot avea competențe digitale variate, ceea ce necesită o interfață simplă și intuitivă

### 3. Conformitatea cu Standardele

- **Constrângere:** Aplicația trebuie să respecte reglementările privind protecția datelor personale, cum ar fi GDPR în Europa.

### 4. Cerințe de Interfață/Protocol

- **Constrângere:** Interfețele cu alte sisteme trebuie să fie sigure și eficiente.

### 5. Cerințe pentru Depozitarea și Distribuția Datelor

- **Constrângere:** Datele trebuie să fie stocate în mod securizat și să respecte legile privind confidențialitatea.

### 6. Cerințe de Performanță

- **Constrângere:** Aplicația trebuie să fie responsive și să aibă timpi de încărcare rapizi.

## 2.2.3 Riscuri

### 1. Riscul Tehnologic

- **Descriere:** Tehnologiile selectate pentru dezvoltare pot deveni depășite sau pot fi înlocuite de alternative mai eficiente.

## **2. Riscul de Securitate a Datelor**

- **Descriere:** Posibilitatea de expunere a datelor sensibile ale utilizatorilor din cauza vulnerabilităților de securitate.

## **3. Riscul de Non-Conformitate**

- **Descriere:** Nerespectarea reglementărilor legale, cum ar fi GDPR pentru protecția datelor.

## **4. Riscul de Întreruperi ale Serviciului**

- **Descriere:** Întreruperi ale serviciului datorate defecțiunilor hardware, erorilor software sau atacurilor cibernetice.

## **5. Riscul de Acceptare de către Utilizatori**

- **Descriere:** Rezistența utilizatorilor la adoptarea noii aplicații, posibil din cauza complexității sau neîncrederii în tehnologie.

## **6. Riscul de Performanță Sub Așteptări**

- **Descriere:** Aplicația s-ar putea să nu îndeplinească așteptările de performanță, ducând la o experiență de utilizare slabă.



## 3. Considerații de proiectare

---

### 3.1 Obiective și linii directoare (ghiduri)

Înainte de a elabora o soluție de design completă, trebuie să abordăm următoarele probleme:

**Completabilitatea și claritatea cerințelor:** Este esențial să ne asigurăm că cerințele sistemului sunt complete, clare și lipsite de ambiguități. Orice neclarități sau lacune în cerințe ar putea duce la implementări incorecte sau la neîndeplinirea așteptărilor utilizatorilor.

**Gestionarea riscurilor:** Este important să identificăm și să evaluăm riscurile potențiale asociate cu proiectul, cum ar fi riscurile tehnice, financiare și de securitate. Aceasta include identificarea riscurilor, evaluarea impactului și probabilității acestora și dezvoltarea unui plan de gestionare a riscurilor.

**Securitatea și confidențialitatea datelor:** Trebuie să abordăm aspectele legate de securitatea și confidențialitatea datelor pentru a proteja informațiile sensibile ale utilizatorilor împotriva accesului neautorizat sau a pierderii.

### 3.2 Metode de dezvoltare

Pentru dezvoltarea sistemului și a software-ului, vom adopta o abordare agilă și orientată pe obiecte, cu utilizarea extensivă a limbajului UML (Unified Modeling Language) pentru documentarea și comunicarea design-ului. Această abordare ne permite să ne concentrăm pe identificarea și dezvoltarea obiectelor semnificative din domeniul aplicației, facilitând astfel o înțelegere clară a structurii și funcționării sistemului.

De asemenea, vom utiliza tehnici de prototipare pentru a valida și itera rapid asupra conceptelor și funcționalităților cheie ale sistemului, permitându-ne să obținem feedback-ul timpuriu al utilizatorilor și să ajustăm design-ul în consecință.

### 3.3 Strategii de arhitectură

Pentru arhitectura sistemului, am ales următoarele strategii de design:

Utilizarea limbajului de programare PHP pentru backend: Alegerea limbajului PHP pentru partea de backend a sistemului a fost motivată de familiaritatea și versatilitatea acestuia în

dezvoltarea aplicațiilor web, permițând implementarea eficientă a funcționalităților și a logicilor de business.

Folosirea Figma pentru designul interfeței utilizator (UI): Am utilizat Figma pentru a crea interfața utilizator (UI), datorită capacității sale de a oferi un mediu colaborativ și ușor de utilizat pentru proiectarea și prototiparea interfețelor web.

Tehnologii web tradiționale pentru frontend (HTML, CSS, JavaScript): Pentru partea de frontend, am optat pentru utilizarea tehnologiilor web tradiționale, cum ar fi HTML, CSS și JavaScript, pentru a asigura compatibilitatea și accesibilitatea interfeței utilizator pe diverse platforme și dispozitive.

Interconectarea frontend-ului și backend-ului prin intermediul API-urilor: Am proiectat sistemul pentru a interconecta frontend-ul și backend-ul prin intermediul API-urilor (Interfețe de Programare a Aplicațiilor), facilitând comunicarea și schimbul de date între cele două părți ale aplicației.

Implementarea modulară și extensibilă: Arhitectura sistemului este proiectată pentru a fi modulară și extensibilă, permițând adăugarea și modificarea ușoară a funcționalităților în viitor, în funcție de evoluția cerințelor și nevoilor utilizatorilor.

## 4. Arhitectura Sistemului și Proiectarea Arhitecturii

---

Arhitectura sistemului este dezmembrată în următoarele componente principale:

### **Frontend-ul (Interfața Utilizator):**

Responsabilitate: Interacționează direct cu utilizatorii și furnizează o interfață grafică intuitivă și ușor de utilizat pentru accesarea funcționalităților sistemului.

Tehnologii utilizate: HTML, CSS, JavaScript, împreună cu framework-uri precum React sau Angular pentru gestionarea stării și interacțiunilor interactive.

### **Backend-ul (Logica de Afaceri):**

Responsabilitate: Procesează cererile de la frontend și gestionează logica de afaceri a aplicației, inclusiv autentificarea utilizatorilor, validarea datelor, gestionarea sesiunilor și accesul la baza de date.

Tehnologii utilizate: PHP pentru gestionarea logică a serverului și MySQL pentru gestionarea bazei de date.

### **Baza de Date:**

Responsabilitate: Stocarea și gestionarea datelor sistemului, inclusiv informații despre utilizatori, programări, informații despre clase și abonamente.

Tehnologii utilizate: MySQL sau alte sisteme de gestionare a bazelor de date relaționale (RDBMS) pentru asigurarea persistenței și integrității datelor.

### **API-uri (Interfețe de Programare a Aplicațiilor):**

Responsabilitate: Furnizarea unui set de endpoint-uri și metode pentru a permite comunicația între frontend și backend, precum și integrarea cu alte aplicații terțe sau servicii externe.

Tehnologii utilizate: Utilizarea limbajului PHP pentru implementarea API-urilor RESTful sau GraphQL pentru gestionarea cererilor și răspunsurilor HTTP.

## 4.1 Vedere logică

Vederea logică a sistemului este gestionată de backend-ul aplicației, care include o serie de componente și module pentru procesarea și gestionarea cererilor de la frontend și pentru accesarea și manipularea datelor din baza de date. Aceste componente includ:

**Controlerul utilizatorilor:** Responsabil pentru gestionarea operațiilor legate de autentificarea și autorizarea utilizatorilor, precum și pentru gestionarea sesiunilor și a accesului la resursele protejate.

**Controlerul programărilor:** Gestionează operațiile legate de programări, inclusiv crearea, actualizarea și ștergerea programărilor, precum și furnizarea informațiilor relevante despre programări pentru utilizatori.

**Controlerul claselor și abonamentelor:** Se ocupă de gestionarea claselor disponibile în sala de fitness și a abonamentelor disponibile pentru utilizatori, inclusiv adăugarea, actualizarea și ștergerea acestora.

**Controlerul de interfață API:** Furnizează o interfață de programare a aplicației (API) pentru comunicarea între frontend și backend, gestionând cererile HTTP și furnizând datele solicitate de către frontend.

## 4.2 Arhitectură hardware

Arhitectura hardware a sistemului este centralizată și include următoarele componente:

**Serverul de prezentare:** Acesta este un singur server care găzduiește frontend-ul aplicației web. Este responsabil pentru furnizarea paginilor web și a interfeței utilizatorului către browserul utilizatorilor.

**Serverul de aplicație:** Acest server găzduiește backend-ul aplicației, inclusiv logica de afaceri, controlerele și gestionarea bazelor de date. Aici sunt procesate cererile primite de la frontend și sunt returnate rezultatele corespunzătoare.

**Serverul de bază de date:** Acest server găzduiește baza de date a aplicației, care stochează toate datele relevante, cum ar fi informațiile despre utilizatori, programările, informațiile despre clase și abonamente.

## 4.3 Arhitectură software

Componentele software ale sistemului includ:

### Backend-ul aplicației:

Limbaj de programare PHP: Folosit pentru a dezvolta logica de afaceri a aplicației și pentru a interacționa cu baza de date.

Bază de date MySQL: Folosită pentru stocarea datelor utilizatorilor, programărilor, claselor și abonamentelor.

Serverul web Apache: Utilizat pentru a servi paginile web și pentru a gestiona cererile HTTP.

### Frontend-ul aplicației web:

HTML, CSS, JavaScript: Folosite pentru a crea interfața utilizatorului și pentru a gestiona interacțiunea cu utilizatorul.

Framework-ul Figma: Utilizat pentru a proiecta și a prototipa interfața utilizatorului.

### Alte componente software:

Sistemul de operare Linux (de exemplu, Ubuntu Server): Folosit ca platformă pentru a rula serverul web și alte servicii necesare.

JWT (JSON Web Tokens): Folosit pentru autentificarea și autorizarea utilizatorilor.

Composer (pentru PHP) și npm (pentru JavaScript): Utilizate pentru gestionarea dependențelor și a pachetelor de software.

Git și GitHub: Folosite pentru controlul versiunilor și colaborarea în echipă asupra codului sursă.

## 4.4 Arhitectura informațiilor

În cadrul sistemului, vor fi stocate diverse tipuri de informații, inclusiv:

### Informații despre utilizatori:

Nume, adrese de e-mail, parole criptate și alte detalii de autentificare.

Informații de contact, cum ar fi adresele fizice și numerele de telefon.

Detalii despre contul utilizatorului, cum ar fi abonamentele și istoricul de achiziții.

#### Informații despre programări:

Date și ore ale programărilor.

Detalii despre profesori și clasele rezervate.

Locația și durata claselor programate.

#### Informații despre plăți:

Detalii despre tranzacțiile financiare, cum ar fi sumele plătite și metodele de plată utilizate.

Informații despre facturi și istoricul plăților.

#### Informații despre cursuri și clase:

Detalii despre cursuri disponibile, inclusiv titluri, descrieri și costuri.

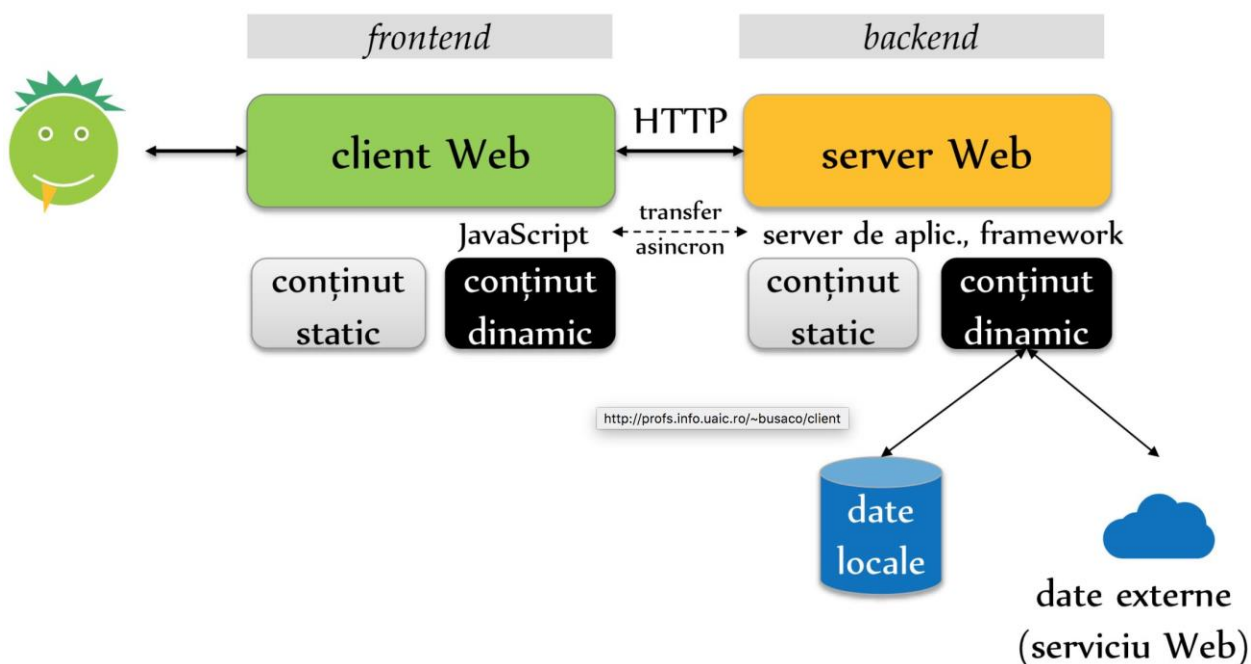
Orarele claselor și disponibilitatea profesorilor.

#### Informații administrative:

Date despre personalul administrativ, cum ar fi profesorii și angajații școlii.

Documente administrative, cum ar fi rapoarte de activitate și politici ale școlii.

## 4.5 Diagrama de arhitectură a sistemului



## 5. Proiectarea sistemului

### 5.1 Proiectarea bazei de date

Acest dicționar de date oferă o imagine de ansamblu asupra elementelor de date implicate în sistem, împreună cu caracteristicile lor asociate și descrieri relevante. Asemenea informații sunt esențiale pentru proiectarea și implementarea corectă a bazelor de date și a întregului sistem.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	<b>grupa_musculara</b>	enum('Cvadriceps', 'Triceps', 'Biceps', 'Spate', '...', '...')	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	<b>denumire</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	<b>video1</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	<b>video2</b>	varchar(244)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 6	<b>pas1</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 7	<b>pas2</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 8	<b>pas3</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More

#### 5.1.1 Obiecte de date și structuri de date rezultante

Utilizator:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	<b>email</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	<b>name</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	<b>password</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More














































Antrenori:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>InstructorID</b>	int(11)			No	None			Change  Drop  More
<input type="checkbox"/> 2	<b>Name</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 3	<b>Image_url</b>	longtext	utf8mb4_general_ci		Yes	NULL			Change  Drop  More

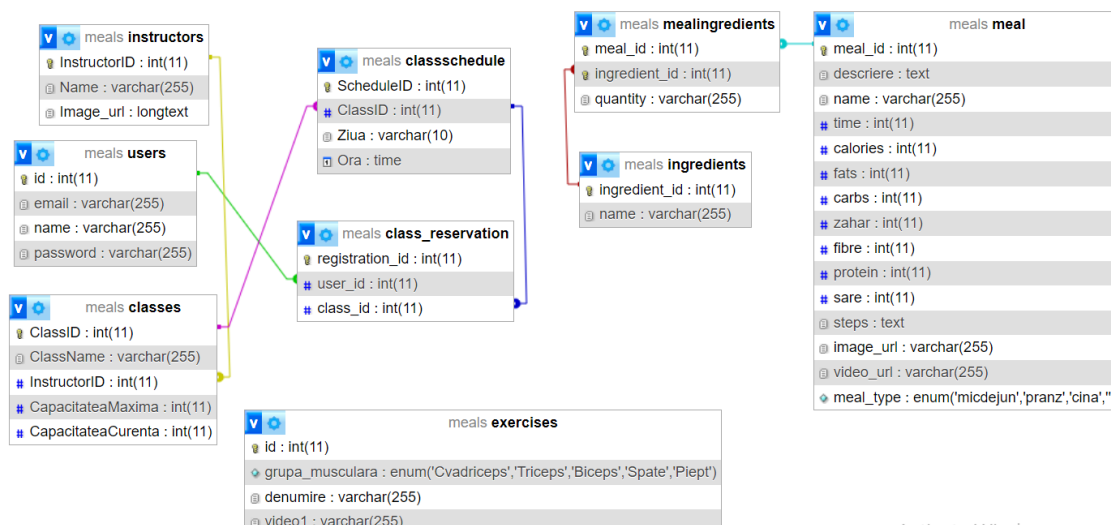
Clase:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>ClassID</b>	int(11)			No	None			Change  Drop  More
<input type="checkbox"/> 2	<b>ClassName</b>	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 3	<b>InstructorID</b>	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 4	<b>CapacitateaMaxima</b>	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 5	<b>CapacitateaCurenta</b>	int(11)			Yes	NULL			Change  Drop  More

Idei de mese:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 meal_id	int(11)			No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2 descriere	text	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3 name	varchar(255)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	4 time	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	5 calories	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	6 fats	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	7 carbs	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	8 zahar	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	9 fibre	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	10 protein	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	11 sare	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	12 steps	text	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	13 image_url	varchar(255)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	14 video_url	varchar(255)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	15 meal_type	enum('micdejun', 'pranz', 'cina', '')	utf8mb4_general_ci		No	None			 Change  Drop  More

#### 5.1.1.1 Baze de date



### 5.1.1.2

## 5.2 Interfețe utilizator

### Utilizatori Obisnuiti:

Descriere: Această clasă de utilizatori sunt persoanele care utilizează aplicația pentru a accesa informații despre clase, pentru a verifica disponibilitatea și pentru a face rezervări.

Responsabilități: Accesarea aplicației, căutarea și vizualizarea detaliilor claselor, realizarea de rezervări.

Competențe: Utilizatori obișnuiți, familiarizați cu utilizarea aplicațiilor web și procesul de rezervare online.



Număr anticipat de utilizatori: Aproximativ 500 de utilizatori.

Număr maxim de utilizatori simultani: 50 de utilizatori.

Utilizatori externi: Nu sunt prevăzuți utilizatori externi pentru această clasă.

#### **Administratori de Clase:**

Descriere: Această clasă de utilizatori sunt responsabili de administrarea și gestionarea claselor disponibile, actualizarea programelor, a capacității și a altor detalii relevante.

Responsabilități: Administrarea claselor, actualizarea informațiilor despre clase, gestionarea capacității și programelor.

Competențe: Cunoștințe avansate despre procesul de gestionare a claselor și a programelor, capacități de gestionare a bazelor de date.

Număr anticipat de utilizatori: Aproximativ 10 administratori.

Număr maxim de utilizatori simultani: 3 administratori.

Utilizatori externi: Nu sunt prevăzuți utilizatori externi pentru această clasă.

#### **Suport Tehnic:**

Descriere: Această clasă de utilizatori sunt responsabili pentru furnizarea de suport tehnic pentru utilizatorii sistemului în cazul unor probleme tehnice sau întrebări legate de funcționalitățile aplicației.

Responsabilități: Furnizarea de asistență tehnică și rezolvarea problemelor utilizatorilor, clarificarea întrebărilor legate de funcționalitate.

Competențe: Cunoștințe tehnice avansate despre aplicație și capacitatea de a comunica eficient cu utilizatorii.

Număr anticipat de utilizatori: Aproximativ 5 agenți de suport tehnic.

Număr maxim de utilizatori simultani: 2 agenți de suport tehnic.

Utilizatori externi: Nu sunt prevăzuți utilizatori externi pentru această clasă.

### **5.2.1 Intrări**

#### **Interfață Utilizator Grafică (GUD):**

Utilizare: Interfața utilizatorului este principalul mijloc de intrare pentru utilizatori. Aceasta include ferestre și formulare interactive.

Descriere: Utilizatorii introduc date prin completarea câmpurilor din formularele interactive sau prin selectarea opțiunilor din meniuri și butoane.

Elemente de date asociate: Nume, dată de naștere, adresă, număr de telefon, e-mail etc.

Criterii de editare: Verificare a formatului corect al datelor (de exemplu, verificarea formatului de e-mail), asigurarea completării câmpurilor obligatorii.

Mesaje asociate: Mesaje de eroare pentru date nevalide sau câmpuri necompletate.

### **Scanere de Coduri de Bare:**

Utilizare: Utilizat pentru scanarea codurilor de bare de pe carduri de membru sau produse.

Descriere: Utilizatorii scanează codurile de bare folosind scanerele dedicate, care sunt conectate la sistem prin intermediul interfeței USB sau wireless.

Elemente de date asociate: Coduri de bare asociate cu produse sau carduri de membru.

Criterii de editare: Verificarea validității codului de bare scanat, asigurarea că produsul sau cardul asociat este înregistrat în sistem.

Mesaje asociate: Mesaje de confirmare a scanării reușite sau a erorilor de scanare.

### **Interfață de Comandă Textuală (CLI):**

Utilizare: Utilizat pentru introducerea comenzilor și parametrilor de către utilizatorii avansați sau administratorii de sistem.

Descriere: Utilizatorii introduc comenzi și argumente folosind tastatura, iar acestea sunt interpretate de sistem pentru a efectua acțiuni specifice.

Elemente de date asociate: Comenzi, argumente, opțiuni.

Criterii de editare: Verificarea corectitudinii sintaxei comenzilor, interpretarea corectă a parametrilor.

Mesaje asociate: Confirmare a succesului sau a eșecului unei comenzi, mesaje de eroare pentru comenzi invalide.

## **5.2.2 ieșiri**

Raportul de Rezervări Săptămânale:

Cod/Nume: REP001\_Raport\_Rezervari\_Saptamanale

Descriere: Acest raport prezintă rezumatele rezervărilor făcute pentru fiecare clasă de antrenament în decursul unei săptămâni. Include detalii precum numărul total de rezervări pentru fiecare clasă, numele clasei, data și ora fiecărei clase etc.

Scopul ieșirii: Acest raport este destinat managerilor de sală și instructorilor pentru a avea o imagine de ansamblu asupra ocupării claselor și pentru a planifica eficient programul de antrenamente.

## 5.3 Proiectarea interfețelor cu utilizatorul

Pentru proiectarea interfețelor cu utilizatorul, următoarele sunt exemple de ecrane și elemente de interfață prevăzute în sistem:

Ecranul de Autentificare:

Descriere: Acest ecran permite utilizatorilor să introducă numele de utilizator și parola pentru a accesa sistemul.

Elemente de interfață: Câmp pentru introducerea numelui de utilizator, câmp pentru introducerea parolei, buton de autentificare.

Tabloul de Bord al Utilizatorului:

Descriere: Acesta este primul ecran pe care îl vede un utilizator după autentificare. Afișează o privire de ansamblu asupra informațiilor relevante, cum ar fi clasele rezervate, programul de antrenament personalizat, notificări etc.

Elemente de interfață: Panou de navigare, secțiuni pentru clase rezervate, notificări, program de antrenament etc.

Ecranul de Căutare a Claselor si Rezervare:

Descriere: Acest ecran permite utilizatorilor să caute și să filtreze clasele disponibile pe baza criteriilor precum dată, oră, tip de clasă, antrenor etc. și utilizatorii pot selecta o clasă din lista rezultatelor căutării și pot efectua o rezervare folosind acest formular.

Elemente de interfață: Câmpuri de căutare și filtre, rezultatele căutării afișate într-o listă.

Formularul de Rezervare a Clasei:

Descriere: Utilizatorii pot selecta o clasă din lista rezultatelor căutării și pot efectua o rezervare folosind acest formular.

## 6. Scenarii de utilizare

---

Scop: Utilizatorul dorește să rezerve o clasă în sistem.

Utilizatorul se autentifică în sistem folosind numele de utilizator și parola.

După autentificare, utilizatorul este direcționat către tabloul de bord al utilizatorului.

Utilizatorul navighează către opțiunea "Căutare Clase".

În ecranul de căutare a claselor, utilizatorul introduce criteriile de căutare, cum ar fi dată, oră și tip de clasă.

După aplicarea filtrelor, sistemul afișează lista claselor disponibile conform criteriilor introduse.

Utilizatorul selectează clasa dorită din lista rezultatelor.

Utilizatorul revizuieste detaliile clasei, cum ar fi numele antrenorului, data și ora, și apasă butonul "Rezervă Clasa".

Sistemul confirmă rezervarea și actualizează lista de clase rezervate a utilizatorului.

Utilizatorul primește o confirmare a rezervării prin intermediul unei notificări pe tabloul său de bord și, dacă este activată, și printr-un e-mail.

## 7. Proiectare de detaliu

---

### Proiectare Detaliată a Componentelor Hardware

Servere:

Server de Aplicație: Un server puternic, cu suficientă capacitate de procesare și memorie pentru a susține aplicația web și baza de date.

Server de Baze de Date: Un server optimizat pentru operațiuni de bază de date, cu stocare rapidă și performantă.

Server de Securitate: Un server dedicat pentru funcțiile de securitate, inclusiv firewall și alte mecanisme de protecție.

Dispozitive Periferice:

Scanere de Coduri de Bare: Pentru scanarea cardurilor de membru sau codurilor de rezervare.

Imprimante: Pentru tipărirea confirmărilor de rezervare sau a rapoartelor.

Dispozitive de Intrare: Tastaturi și mouse-uri pentru interacțiunea utilizatorului cu interfața.

### Proiectare Detaliată a Componentelor Software

Backend:

Limbaj de Programare: PHP pentru logica de afaceri.

Framework-uri: Utilizarea framework-ului Laravel pentru dezvoltarea rapidă și structurată a aplicației.

Bază de Date: MySQL pentru stocarea datelor.

Frontend:

HTML, CSS, JavaScript: Pentru structura, stilurile și interactivitatea paginilor web.

Figma: Pentru proiectarea interfeței utilizatorului, care va fi ulterior transpusă în cod.

### Integrarea Componentelor Hardware și Software

Configurare și Instalare: Echipa de dezvoltare va instala și configura serverele și dispozitivele periferice conform specificațiilor de proiectare.

Interconectarea Componentelor: Se vor stabili conexiunile necesare între servere și dispozitivele periferice pentru a permite comunicarea corespunzătoare între ele.

### 7.1 Proiectare hardware de detaliu

Pentru a furniza detalii despre fiecare componentă hardware și pentru a asigura construcția sau achiziționarea corectă a acestora, voi prezenta mai jos specificațiile pentru fiecare componentă, împreună cu detaliile de alimentare, semnal și conectori, acolo unde este relevant.

#### Server de Aplicație:

Cerințe de Alimentare: 220V AC, 50-60 Hz

Procesor: Dual-core sau mai mare, cu frecvență de minim 2.0 GHz

Memorie RAM: Minim 8 GB, recomandat 16 GB sau mai mult

Stocare: Minim 256 GB SSD

Conectori: Ethernet (RJ45), USB, VGA/HDMI pentru monitor

Spațiu Rack: 1U sau 2U, în funcție de configurația serverului

#### **Server de Baze de Date:**

Cerințe de Alimentare: 220V AC, 50-60 Hz

Procesor: Quad-core sau mai mare, cu frecvență de minim 2.0 GHz

Memorie RAM: Minim 16 GB, recomandat 32 GB sau mai mult

Stocare: Minim 1 TB SSD pentru sistemul de operare și 2 TB pentru bazele de date

Conectori: Ethernet (RJ45), USB, VGA/HDMI pentru monitor

Spațiu Rack: 1U sau 2U, în funcție de configurația serverului

#### **Server de Securitate:**

Cerințe de Alimentare: 220V AC, 50-60 Hz

Procesor: Dual-core sau mai mare, cu frecvență de minim 1.5 GHz

Memorie RAM: Minim 4 GB, recomandat 8 GB sau mai mult

Stocare: Minim 128 GB SSD

Conectori: Ethernet (RJ45), USB, VGA/HDMI pentru monitor

Spațiu Rack: 1U sau 2U, în funcție de configurația serverului

#### **Scanere de Coduri de Bare:**

Alimentare: Alimentare prin USB sau alimentator extern, în funcție de model

Interfață: USB

Tehnologie: Laser sau CCD, în funcție de aplicație

Compatibilitate: Interfață HID pentru utilizare plug-and-play

## **7.2 Proiectare software de deatliu**

### **7.2.1 Proiectare detaliată de securitate**

#### **1. Autentificare**

Descriere:

Autentificarea este procesul de confirmare a identității unei entități, cum ar fi un utilizator sau un dispozitiv, înainte de a le permite accesul la sistem.

Componente:

Biometria: Utilizarea amprentelor digitale sau a scanărilor de retină pentru autentificarea utilizatorilor.

Token-uri de autentificare: Dispozitive fizice (cum ar fi chei USB sau dispozitive mobile) care generează coduri unice pentru autentificare.

## 2. Autorizare

Descriere:

Autorizarea este procesul de acordare a accesului la resursele sistemului după ce o entitate a fost autentificată.

Componente:

Liste de control al accesului (ACL): Specifică permisiunile de acces pentru utilizatori sau grupuri de utilizatori la fișiere, directoare sau alte resurse.

Politici de autorizare: Seturi de reguli și restricții care stabilesc cine poate accesa anumite resurse și ce acțiuni pot fi efectuate asupra acestora.

## 3. Jurnalizare și auditare

Descriere:

Jurnalizarea și auditarea implică înregistrarea activităților de utilizator și a evenimentelor de sistem pentru urmărire și analiză ulterioară.

Componente:

Jurnale de securitate: Înregistrări detaliate ale activităților de autentificare și acces la sistem.

Instrumente de monitorizare a securității: Software specializat care analizează și interpretează datele din jurnalele de securitate pentru a identifica anomalii sau potențiale amenințări.

## 7.2.2 Proiectare de detaliu pentru performanța sistemului

### Cerințe/Estimări de Capacitate și Volum

Descriere:

Identificarea cerințelor de capacitate și volum pentru fiecare componentă hardware, inclusiv estimările pentru volumul de date și traficul de rețea.

Exemplu:

Pentru serverele de aplicații, este posibil să fie necesară o capacitate de procesare suficientă pentru a gestiona un număr specific de cereri pe secundă.

### Așteptări de Performanță

Descriere:

Definirea așteptărilor de performanță pentru fiecare componentă hardware, cum ar fi timpul de răspuns al serverului sau lățimea de bandă a rețelei.

Exemplu:

Un server de baze de date ar trebui să ofere un timp de răspuns sub 1 secundă pentru interogările de bază și să poată gestiona un volum ridicat de tranzacții simultane.

### **Cerințe de Disponibilitate**

Descriere:

Definirea cerințelor de disponibilitate pentru fiecare componentă hardware, inclusiv procentul de timp în care sistemul trebuie să fie operațional.

Exemplu:

Un switch de rețea ar trebui să fie disponibil 99,99% din timpul de funcționare, asigurând conectivitatea neîntreruptă la rețea.

### **Proiectare de Performanță pentru a Îndeplini Cerințele de Capacitate**

Descriere:

Identificarea și implementarea măsurilor necesare pentru a asigura că fiecare componentă hardware poate satisface cerințele de capacitate stabilite.

Exemplu:

Adăugarea de procesoare suplimentare și memorie RAM pentru serverele de aplicații pentru a gestiona un volum crescut de cereri.Proiectare detaliată a comunicațiilor interne (între componente)Controale pentru verificarea integrității sistemului



## Anexa A: Gestiunea modificărilor documentului

*Instrucțiuni: Furnizați informații despre modul în care dezvoltarea și distribuția documentului va fi controlată și urmărită. Utilizați tabelul de mai jos pentru a furniza numărul de versiune, data versiunii, autorul/deținătorul versiunii și o scurtă descriere a motivului pentru crearea versiunii revizuite.*

**Tabel 1 – Înregistrarea modificărilor asupraa documentului curent**

versiune	Data	Autorul/Deținătorul	Descriere
<X.X>	<ZZ/LL/AAAA>	<nume autor>	<Descrierea modificării>
<X.X>	<ZZ/LL/AAAA>	<nume autor>	<Descrierea modificării>
<X.X>	<ZZ/LL/AAAA>	<nume autor>	<Descrierea modificării>

## Anexa B: Acronime

*Instrucțiuni: Furnizați o listă de acronime și traduceri literale asociate utilizate în cadrul documentului. Enumerați acronimele în ordine alfabetică folosind un format tabular, așa cum este ilustrat mai jos.*

**Tabel 2 - Acronime**

Acronim	Forma completă
<Acronim>	<Forma completă>
<Acronim>	<Forma completă>
<Acronim>	<Forma completă>

## Anexa C Documente la care se face referire

*Instrucțiuni: Sintetizați relația acestui document cu alte documente relevante. Furnizați informații de identificare pentru toate documentele folosite pentru a ajunge la și/sau referite în acest document (de exemplu, documente conexe și/sau asociate, documente prealabile, documentație tehnică relevantă, etc.).*

**Tabel 3 – Documente la care se face referire**

Nume document	Locație sau URL	Data emitere document
< Nume document >	<Locație sau URL>	<ZZ/LL/AAAA>
< Nume document >	<Locație sau URL>	<ZZ/LL/AAAA>
< Nume document >	<Locație sau URL>	<ZZ/LL/AAAA>