# Softwarová specifikace požadavků (SRS) – Pocket Gym

Autor: Mykhailo Levurda

## 1. Úvod

### 1.1 Účel dokumentu

Tento dokument definuje softwarové požadavky pro aplikaci Pocket Gym, webovou platformu určenou pro správu fitness centra, členství, lekcí, trenérů, plateb a inventáře.  
Slouží vývojářům, testerům i koncovým uživatelům.

### 1.2 Rozsah systému

Systém umožňuje správu členů a trenérů, rezervaci lekcí, sledování plateb, správu inventáře a odesílání notifikací. Obsahuje frontend (React), backend (Express, MongoDB) a administrační rozhraní.

### 1.3 Definice a zkratky

SRS: Software Requirements Specification  
SDD: Software Design Document  
UI: Uživatelské rozhraní  
BE: Backend  
FE: Frontend  
CRUD: Create, Read, Update, Delete

### 1.4 Související dokumenty

- SDD (Software Design Document)  
- Uživatelská a administrátorská příručka  
- UML diagramy (Use case, Class, Sequence)(SDD)

## 2. Celkový popis

### 2.1 Perspektiva systému

Pocket Gym je webová aplikace založená na architektuře klient-server. Frontend je postaven pomocí Reactu, backend používá Node.js, Express a MongoDB. Komunikace probíhá přes REST API. Nasazení probíhá přes Vercel.

### 2.2 Uživatelské role

- Administrátor: správa systému, uživatelů, lekcí, plateb i inventáře  
  
- Trenér: správa vlastního rozvrhu a komunikace s členy  
  
- Recepční: registrace členů, rezervace, přehled o platbách  
  
- Člen: rezervace lekcí, kontrola plateb, správa profilu

### 2.3 Omezení

- Aplikace vyžaduje moderní prohlížeč (Chrome, Firefox, Edge)  
- Nutné internetové připojení  
- Přístup je umožněn pouze autentizovaným uživatelům

### 2.4 Předpoklady a závislosti

- Frontend: React, React Router, Axios  
- Backend: Node.js, Express, Mongoose  
- DB: MongoDB  
- Deployment: Vercel  
- Prohlížeč: Chrome, Firefox, Edge

## 3. Funkční požadavky

- Registrace a autentizace uživatelů dle role (administrátor, trenér, člen, recepční)  
  
- Správa uživatelských účtů (vytváření, úprava, mazání)  
  
- Rezervace a rušení lekcí členem  
  
- Správa tréninkového rozvrhu a lekcí trenérem  
  
- Sledování plateb a historie transakcí  
  
- Správa inventáře: přidávání, úpravy, zobrazení vybavení  
  
- Notifikace o zpožděných platbách  
  
- Ověření přihlášení, validace vstupních dat a přístupových oprávnění

## 4. Uživatelské scénáře (User Stories)

- \*\*Registrace člena\*\*: Jako recepční mohu přidat nového člena, aby jeho profil mohl být spravován digitálně.  
  
- \*\*Rezervace lekce\*\*: Jako člen mohu rezervovat lekci, abych se mohl zúčastnit tréninku.  
  
- \*\*Sledování plateb\*\*: Jako administrátor mohu sledovat platby členů a zobrazit upozornění při zpoždění.  
  
- \*\*Správa inventáře\*\*: Jako administrátor mohu upravit skladovou položku, aby byl evidován aktuální stav.

## 5. Nefunkční požadavky

- \*\*Výkon\*\*: Odezva systému musí být do 2 sekund při běžné zátěži.  
  
- \*\*Bezpečnost\*\*: Šifrování citlivých dat, autentizace a autorizace podle rolí.  
  
- \*\*Použitelnost\*\*: Intuitivní uživatelské rozhraní přístupné i na mobilních zařízeních.  
  
- \*\*Škálovatelnost\*\*: Podpora rozšíření systému pro více uživatelů a funkcí.  
  
- \*\*Spolehlivost a dostupnost\*\*: 99% dostupnost přes cloudové řešení (Vercel).  
  
- \*\*Udržovatelnost\*\*: Kód dokumentovaný, s podporou unit a integračních testů.

## 6. Požadavky na rozhraní

- Frontend: React, HTML5, CSS, JavaScript  
  
- Backend: Node.js + Express  
  
- Databáze: MongoDB, přístup přes Mongoose  
  
- API: RESTful rozhraní s JSON komunikací  
  
- CORS konfigurace specifikovaná ve `vercel.json`  
  
- Nasazení: Vercel (frontend), lokální nebo cloud běh backendu

## 7. Kvalita a testování

- Jednotkové testy pro klíčové funkce backendu  
  
- Integrační testy API a uživatelského rozhraní  
  
- Testování bezpečnosti a validace vstupních dat  
  
- Uživatelské akceptační testy (UAT) pro ověření přívětivosti rozhraní  
  
- Manuální testování v moderních prohlížečích

## 8. Požadavky na dokumentaci

- Uživatelská příručka pro členy a trenéry  
  
- Administrátorská příručka  
  
- Technická dokumentace (API, datový model, architektura)  
  
- Manuál pro nasazení a údržbu systému

## 9. Přílohy

- Diagramy případů užití (Use Case)  
  
- Sekvenční diagramy (např. přihlášení, rezervace)  
  
- Třídní diagram (struktura modelů MongoDB)  
  
- Uživatelská a administrátorská příručka  
  
- Software Design Document (SDD)