

Dokumentace projektu

Název projektu: GymManager

Autor: Filip Pištěk

Email: pistek@spsejecna.cz

Datum vypracování: 27.12.2025 - 11.1.2026

Název školy: SPŠE Ječná

1. Specifikace požadavků (Requirements)

Cílem projektu bylo vytvořit informační systém pro správu fitness centra, který umožňuje evidenci klientů, trenérů, lekcí a správu rezervací.

Funkční požadavky (Functional Requirements)

- Správa klientů:** Uživatel může vytvářet, číst, upravovat a mazat klienty (CRUD).
- Správa lekcí a trenérů:** Systém zobrazuje rozvrh lekcí včetně přiřazených trenérů.
- Rezervace (Transakce):** Uživatel může vytvořit rezervaci klienta na lekci. Tato akce musí být atomická (kontrola kreditu, vytvoření vazby, stržení kreditu).
- Reporty:** Systém generuje přehled vytíženosti trenérů.
- Import dat:** Systém umožňuje hromadný import lekcí a trenérů z CSV souborů.

2. Architektura aplikace

Aplikace je navržena jako **Konzolová aplikace** v jazyce C# na platformě .NET Framework. Architektura striktně dodržuje **Repository Pattern (D1)** pro oddělení datové vrstvy od aplikační logiky.

Strukturální diagram (Zjednodušený Class Diagram)

Aplikace je rozdělena do tří hlavních vrstev:

- Entities (Datová vrstva):** Třídy (`Client`, `Trainer`, `Lesson`, `Booking`) reprezentující tabulky v databázi. Jsou to prosté POCO objekty bez logiky.
- Repositories (Přístup k datům):**
 - `BaseRepository`: Zajišťuje připojení k DB.
 - `ClientRepository`: CRUD operace nad tabulkou Clients.
 - `LessonRepository`: Čtení z View, Import dat.
 - `BookingRepository`: Řízení transakcí (BeginTransaction, Commit, Rollback).

3. **Services :**
`ReportService`: Agregace dat pro statistiky.
`ImportService`: Parsing CSV souborů.
 4. **TestingData (testovací data do import funkcí)**
`tlessons.csv`
`trainers.csv`
-

3. Popis běhu aplikace (Behaviorální model)

Klíčovým procesem aplikace je **Vytvoření rezervace**, která probíhá v transakci.

Activity Diagram: Proces Rezervace

1. Uživatel vybere Klienta (ID) a Lekci (ID).
 2. Aplikace zahájí databázovou transakci (`BeginTransaction`).
 3. **Krok 1:** Zjištění aktuálního kreditu klienta (`SELECT`).
 4. **Rozhodnutí:**
 - *Kredit < Cena*: Vyhození výjimky -> **Rollback** (zrušení transakce) -> Výpis chyby.
 - *Kredit >= Cena*: Pokračovat.
 5. **Krok 2:** Vložení záznamu do tabulky `Bookings` (`INSERT`).
 6. **Krok 3:** Aktualizace kreditu v tabulce `Clients` (`UPDATE`).
 7. **Krok 4:** Zápis do auditního logu `Logs` (`INSERT`).
 8. Potvrzení transakce (`Commit`).
 9. Výpis úspěchu uživateli.
-

4. Použitá rozhraní a knihovny třetích stran

Aplikace je závislá na následujících komponentách:

Softwarové závislosti

- **.NET Framework 4.8:** Runtime prostředí aplikace.
- **Microsoft.Data.SqlClient:** Knihovna pro komunikaci s MS SQL Serverem (ADO.NET).
- **System.Configuration:** Knihovna pro čtení konfiguračních souborů.

Hardware a Služby

- **MS SQL Server:** Aplikace vyžaduje běžící instanci SQL Serveru (LocalDB, Express nebo Enterprise) dostupnou přes TCP/IP nebo Named Pipes.
-

5. Právní a licenční aspekty

- **Licence:** Projekt je šířen pod licencí **MIT License**. Je možné jej volně používat, upravovat a šířit pro vzdělávací a komerční účely, za podmínky uvedení autora.
- **Autorská práva:** Veškerý kód je původním dílem autora. Nebyly použity žádné proprietární knihovny vyžadující placenou licenci.
- **Omezení:** Aplikace je školní projekt a je poskytována "tak jak je", bez záruky funkčnosti v produkčním prostředí.

6. Konfigurace programu

Konfigurace probíhá prostřednictvím souboru **App.config** ve formátu XML, který se nachází v kořenovém adresáři aplikace.

Konfigurační volby

- **Connection Strings (`<connectionStrings>`):**
 - Klíč **GymDb**: Definuje připojovací řetězec k databázi.
 - Uživatel musí nastavit parametry: **Data Source** (server), **Initial Catalog** (databáze), **User ID** a **Password** (pokud nepoužívá integrované zabezpečení).

Příklad konfigurace:

```
<connectionStrings>
    <add name="GymDB"
        connectionString="Data Source=ADRESA_SERVERU;
        Initial Catalog=NAZEV_DATABAZE;
        User ID=UZIVATELSKE_JMENO;Password=HESLO;Encrypt=True;
        TrustServerCertificate=True"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

7. Instalace a spuštění

Detailní postup je uveden v souboru **README.md**.

Stručný postup:

1. Na cílovém SQL Serveru spusťte skript **database_setup.sql** pro vytvoření struktury.
2. V souboru **App.config** nastavte správný connection string.
3. Spusťte soubor **GymManager.exe**.

8. Chybové stavy a jejich řešení

Aplikace má implementované ošetření výjimek (`try-catch` bloky) v celé vrstvě UI i Repositories.

Chybový stav	Příčina	Chování aplikace	Řešení
Connection Failed	Špatné heslo nebo server nedostupný.	Vypíše chybu připojení a ukončí akci.	Zkontrolovat <code>App.config</code> a běžící SQL Server.
Insufficient Funds	Klient nemá dostatek kreditu.	Transakce provede Rollback, vypíše chybu.	Dobít kredit (přes DB) nebo zvolit jiného klienta.
FK Constraint Error	Pokus o smazání klienta s rezervací.	DB zamítne smazání, aplikace vypíše varování.	Nejprve smazat rezervace (v DB).
File Not Found	Chybějící CSV soubor pro import.	Vypíše chybu, že soubor neexistuje.	Zkontrolovat cestu k souboru.

9. Testování a validace

Ověření funkčnosti proběhlo formou **manuálních integračních testů** podle předem definovaných testovacích scénářů (viz přílohy PDF).

- **Test 1 (Happy Path):** Úspěšné vytvoření klienta, zobrazení rozvrhu a vytvoření rezervace. -> **Prošlo.**
- **Test 2 (Chybové stavy):** Pokus o rezervaci bez kreditu. Systém správně vrátil DB do původního stavu. -> **Prošlo.**
- **Test 3 (Import):** Import souborů `lessons.csv` a `trainers.csv`. Data byla korektně uložena. -> **Prošlo.**

10. Verze a známé chyby

- **Verze:** 1.0
 - **Známé issues:**
 - Aplikace neobsahuje funkci pro mazání rezervací (musí se řešit přes DB admina).
 - Vstup od uživatele v konzoli není validován na délku řetězce (může způsobit SQL chybu při překročení NVARCHAR(50)).
-

11. E-R Model databáze

Databáze je relační, normalizovaná (3NF).

Schéma tabulek

1. **Clients:** `id` (PK, int), `name` (nvarchar(50)), `surname` (nvarchar(50)), `email` (nvarchar(50)), `credit` (float), `is_active` (bit).
 2. **Trainers:** `id` (PK, int), `name` (nvarchar(50)), `surname` (nvarchar(50)), `specialization` (enum/check).
 3. **Lessons:** `id` (PK, int), `name` (nvarchar(50)), `date_and_time` (datetime), `capacity` (int), `trainer_id` (FK -> Trainers).
 4. **Bookings:** `id` (PK, int), `client_id` (FK -> Clients), `lesson_id` (FK -> Lessons), `date_of_creation`. (*Vazební tabulka M:N*).
 5. **Logs:** `id` (PK, int), `message` (nvarchar(255)), `date` (datetime), `type` (nvarchar(50)).
-

13. Struktura importovaných souborů (Schéma)

Aplikace podporuje import formátu **CSV** (Comma Separated Values). Oddělovačem je čárka ,.

1. Import Lekcí (**lessons.csv**)

Název, Datum a Čas, Kapacita, ID Trenéra

Příklad:

Ranni Yoga,2025-12-01 08:00,15,2
Box,2025-12-01 18:00,10,1

2. Import Trenérů ([trainers.csv](#))

Specializace **musí odpovídat** povoleným hodnotám v DB (CHECK constraint).

Jméno, Příjmení, Specializace

Příklad:

Code snippet

Jan,Novak,Silovy

Petra,Mala,Yoga