

# **DOKUMENTACE PROJEKTU**

## **Football Roster Finder**

**Autor:** Jakub Šrámek

**Škola:** Střední průmyslová škola elektrotechnická Ječná, Praha 2, Ječná 30

**Email:** sramek4@spsejecna.cz

**Datum:** 3.1. 2026

**Typ projektu:** Školní projekt - Databázové aplikace D1

## **1. ÚVOD**

### **1.1 Název a účel projektu**

Football Roster Finder je desktopová aplikace pro správu fotbalových týmů, hráčů, smluv a přestupů s využitím relační databáze Microsoft SQL Server.

### **1.2 Technologie**

- Python 3.13.5
- Microsoft SQL Server
- ODBC Driver 17 for SQL Server
- Tkinter (GUI framework)
- pyodbc (databázové připojení)

### **1.3 Splnění požadavků zadání**

Projekt splňuje požadavek **D1 - DAO Pattern**:

- Implementace vlastního DAO patternu (PlayerDAO, TeamDAO, ContractDAO, PositionDAO)
- 6 tabulek v databázi
- 4 pohledy (VIEW)
- M:N vazba mezi tabulkami
- Všechny požadované datové typy
- Transakce nad více tabulkami
- Agregované reporty
- Import CSV a JSON
- Konfigurační soubor
- Kompletní ošetření chyb

## **2. POŽADAVKY UŽIVATELE**

## 2.1 Use Case Diagram

Zobrazit hráče
Zobrazit týmy
Přidat hráče
Upravit hráče
Odebrat hráče
Přidat tým
Zobrazit soupisku
Přidat hráče do týmu
Odebrat hráče z týmu
Změnit pozici hráče
Aktualizovat minuty
Přestup hráče
Smlouvy
Statistický report
Import / Reset

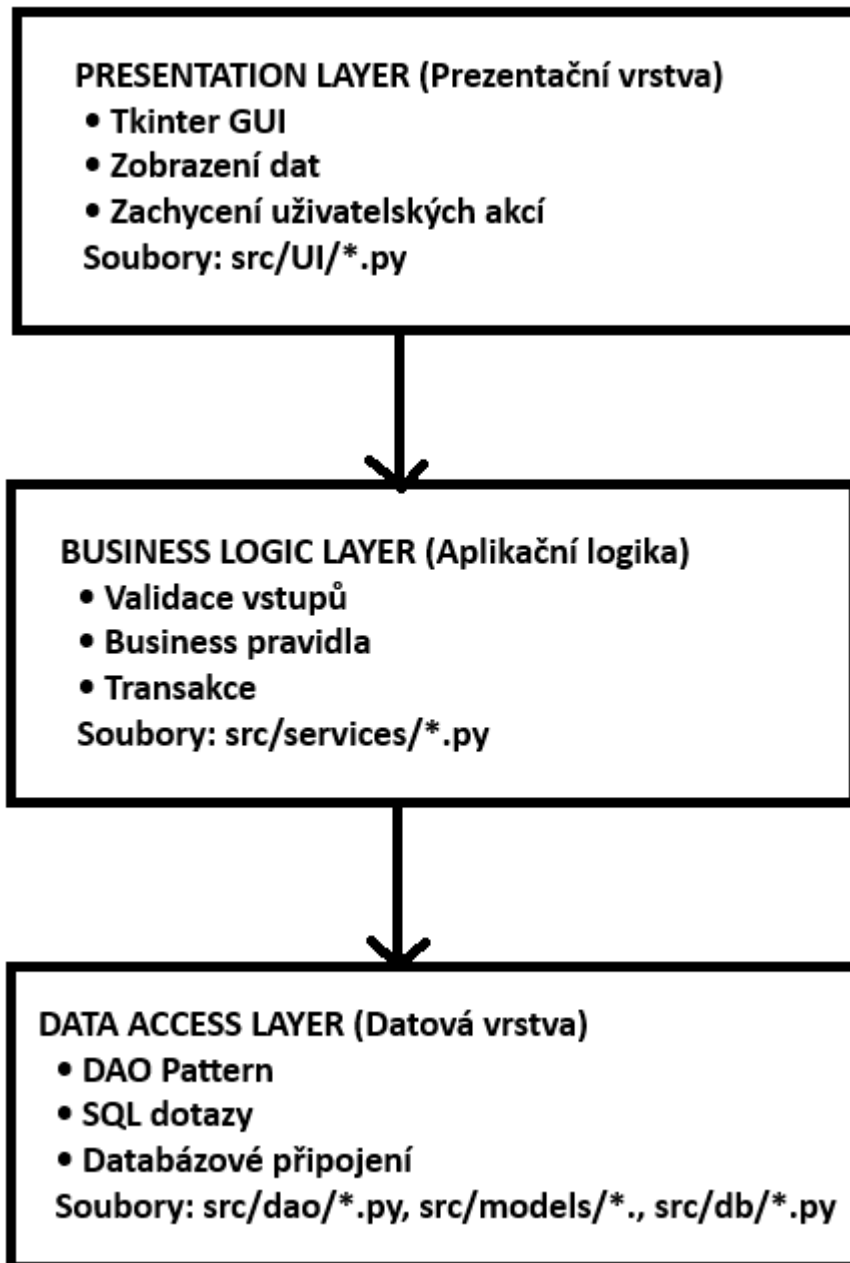
## 2.2 Hlavní funkční požadavky

1. **CRUD operace** pro hráče a týmy
2. **Správa soupiskek** - přiřazování hráčů k týmům s pozicemi
3. **Přestupy hráčů** - atomická operace přesunu mezi týmy
4. **Správa smluv** - vytváření a přiřazování smluv hráčům
5. **Statistické reporty** - agregované údaje z více tabulek
6. **Import dat** - načítání z CSV a JSON
7. **Validace vstupů** - kontrola správnosti dat

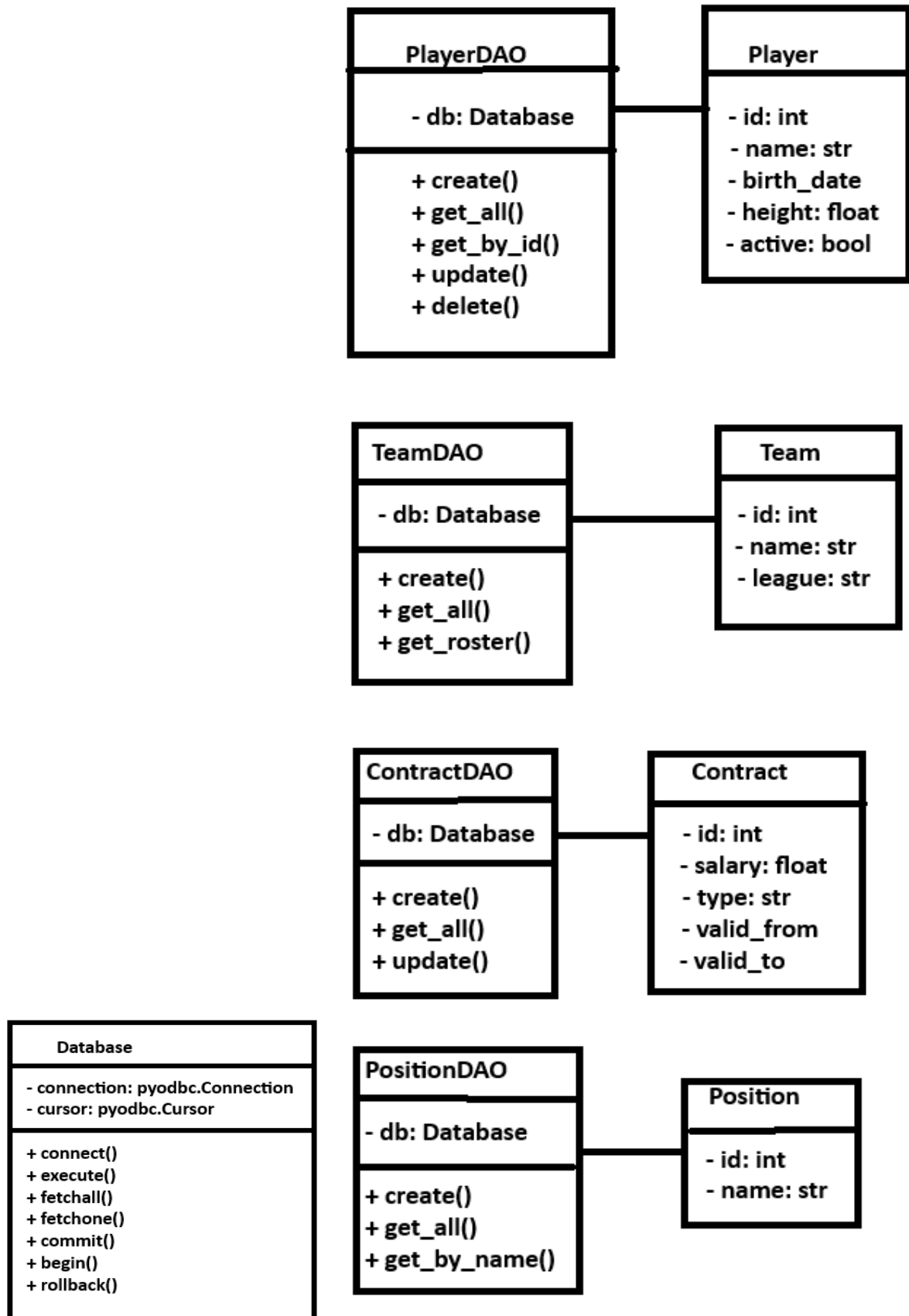
## 3. ARCHITEKTURA APLIKACE

### 3.1 Návrhový vzor - Three-Tier Architecture

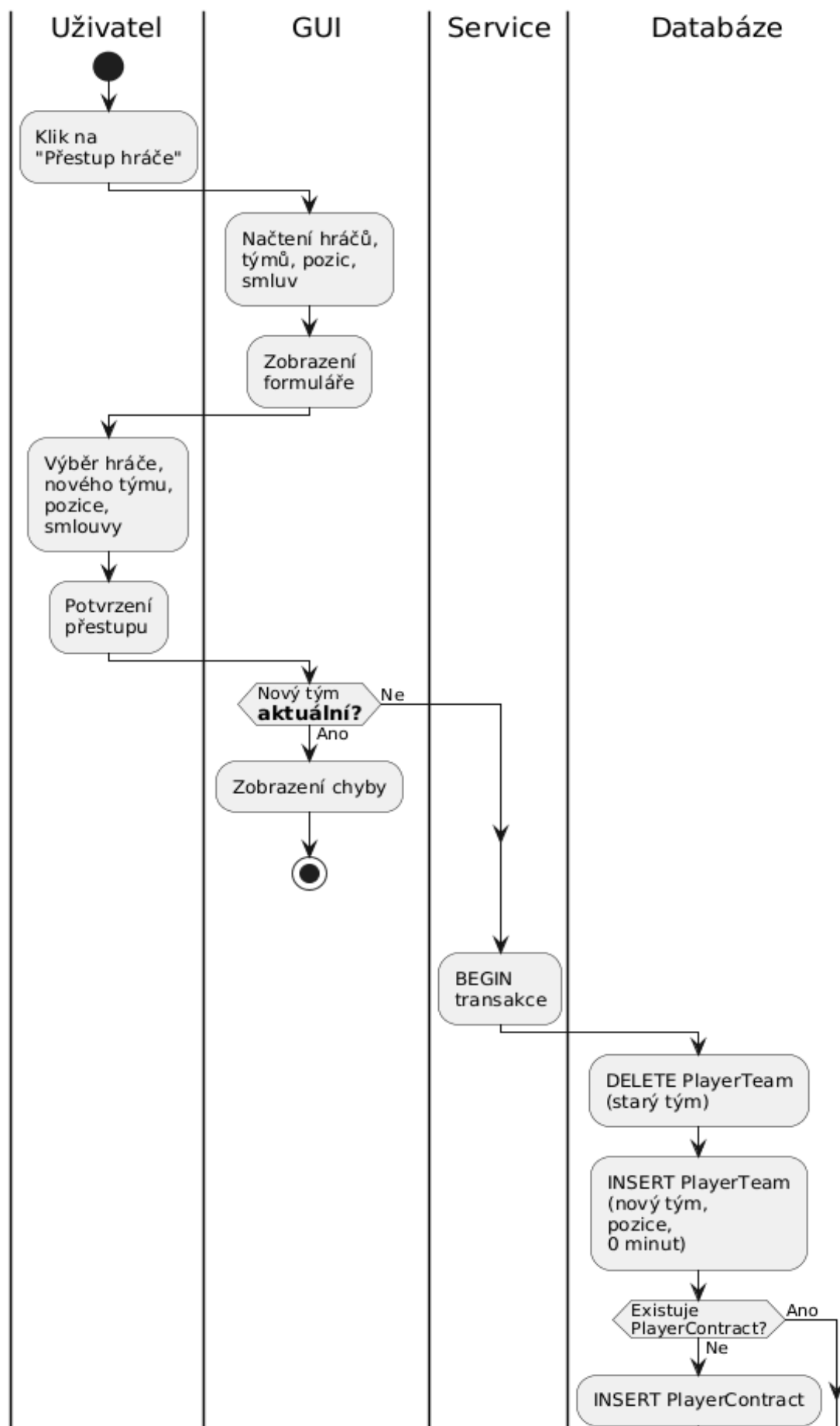
Aplikace využívá třívrstvou architekturu s jasným oddělením odpovědností:

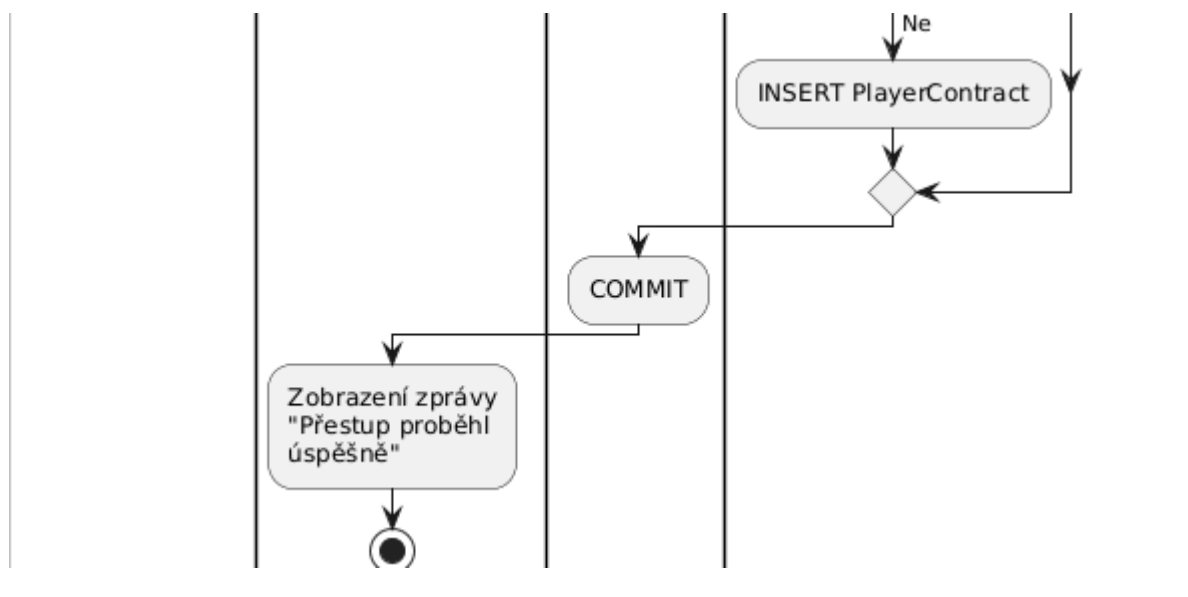


### 3.2 Class Diagram - Hlavní třídy



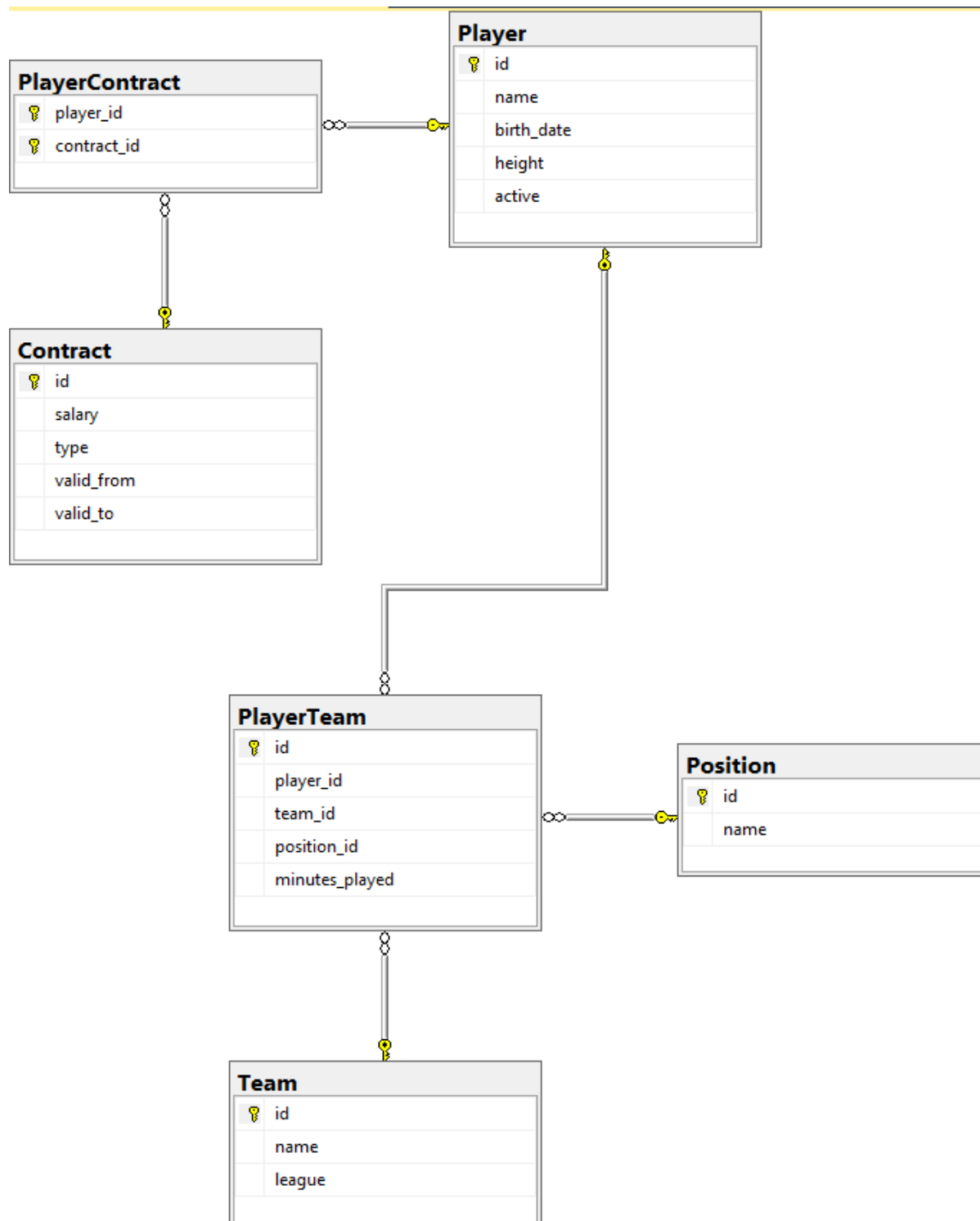
### 3.3 Activity Diagram - Přestup hráče (transakce)





## 4. DATABÁZOVÁ STRUKTURA

### 4.1 E-R Diagram



## 5. FUNKČNÍ POŽADAVKY

## 5.1 Operace nad více tabulkami

**Funkce:** `transfer_player()` v `src/services/transfer_service.py`

**Upravované tabulky:**

1. **PlayerTeam** - odstranění z původního týmu
2. **PlayerTeam** - přidání do nového týmu
3. **PlayerContract** - přiřazení nové smlouvy

## 5.2 Agregovaný report

**VIEW:** `V_TeamStatistics`

**Účel:** Zobrazení souhrnných statistik týmů

**Agregované funkce:**

- `COUNT(pt.player_id)` - počet hráčů v týmu
- `AVG(p.height)` - průměrná výška hráčů
- `SUM(c.salary)` - celkový plat týmu
- `SUM(pt.minutes_played)` - celkové odehrané minuty

**Použité tabulky:**

1. Team
2. PlayerTeam
3. Player
4. PlayerContract
5. Contract

## 5.3 Import dat

**Import CSV - Hráči**

**Soubor:** `src/services/import_service.py`

**Funkce:** `import_players_from_csv()`

**Cílová tabulka:** Player

**Formát CSV:**

csv

name,birth\_date,height,active

Jan Novák,15.05.1995,1.85,1

Petr Svoboda,20.08.1998,1.78,1

**Povinné sloupce:**

- name - jméno hráče (VARCHAR)



- birth\_date - datum narození ve formátu DD.MM.YYYY
- height - výška v metrech (FLOAT)
- active - aktivní (1 = ano, 0 = ne)

### Import JSON - Týmy

**Funkce:** `import_teams_from_json()`

**Cílová tabulka:** Team

### Formát JSON:

```
json
[
  {
    "name": "Slavia Praha",
    "league": "1. LIGA"
  },
  {
    "name": "Sparta Praha",
    "league": "1. LIGA"
  }
]
```

### Povinné položky:

- name - název týmu (VARCHAR)

## 6. KONFIGURACE, INSTALACE A SPUŠTĚNÍ

vše o konfiguraci, instalaci a spuštění naleznete v README které je u projektu:

[https://github.com/Jakubiou/Football\\_Roster\\_Finder.git](https://github.com/Jakubiou/Football_Roster_Finder.git)

## 7. Chybové stavy

### 7.1 Databázové chyby

Kód: **DB\_001**

Chyba: „Config soubor nenalezen“

Příčina: config.json není ve stejné složce jako aplikace

Řešení: Zkopírovat soubor config.json do stejné složky jako .exe soubor

Kód: **DB\_002**

Chyba: „Login failed for user“

Příčina: Špatné přihlašovací údaje

Řešení: Zkontrolovat UID a PWD v souboru config.json

Kód: **DB\_003**

Chyba: „ODBC Driver not found“

Příčina: Chybí ODBC Driver 17

Řešení: Nainstalovat ODBC Driver 17 z oficiálních stránek Microsoft

Kód: **DB\_004**

Chyba: „Database does not exist“

Příčina: Databáze neexistuje

Řešení: Vytvořit databázi pomocí příkazu CREATE DATABASE <DBNAME>

Kód: **DB\_005**

Chyba: „Connection timeout“

Příčina: Server neodpovídá

Řešení: Zkontrolovat adresu serveru a nastavení firewallu

## 7.2 Validační chyby

Kód: **VAL\_001**

Chyba: „Jméno musí mít alespoň 3 znaky“

Příčina: Zadané jméno je příliš krátké

Řešení: Zadat jméno o minimální délce 3 znaky

Kód: **VAL\_002**

Chyba: „Jméno může obsahovat pouze písmena a mezery“

Příčina: Jméno obsahuje čísla nebo speciální znaky

Řešení: Zadat pouze písmena české abecedy a mezery

Kód: **VAL\_003**

Chyba: „Datum narození nemůže být v budoucnosti“

Příčina: Zadané datum je v budoucnosti

Řešení: Zadat datum z minulosti

Kód: **VAL\_004**

Chyba: „Výška musí být mezi 1.0 a 2.5 m“

Příčina: Výška je mimo povolený rozsah

Řešení: Zadat výšku v rozmezí 1.0–2.5 metru

Kód: **VAL\_005**

Chyba: „Liga musí být číslo 1 nebo 2“

Příčina: Neplatné číslo ligy

Řešení: Zadat pouze číslo 1 nebo 2

Kód: **VAL\_006**

Chyba: „Plat musí být kladný“

Příčina: Záporná nebo nulová hodnota platu

Řešení: Zadat kladné číslo

Kód: **VAL\_007**

Chyba: „Datum ,do‘ musí být po datu ,od““

Příčina: Špatné pořadí datumů

Řešení: Datum ukončení smlouvy musí být pozdější než datum začátku

Kód: **VAL\_008**

Chyba: „Minuty musí být kladné číslo“

Příčina: Záporné nebo nulové minuty

Řešení: Zadat kladné celé číslo

### **7.3 Chyby business logiky**

Kód: **BUS\_001**

Chyba: „Hráč již v tomto týmu je“

Příčina: Pokus o přestup do stejného týmu

Řešení: Vybrat jiný tým

Kód: **BUS\_002**

Chyba: „Žádní hráči v databázi“

Příčina: Databáze neobsahuje žádné hráče

Řešení: Přidat hráče ručně nebo importovat data

Kód: **BUS\_003**

Chyba: „Žádné týmy v databázi“

Příčina: Databáze neobsahuje žádné týmy

Řešení: Přidat týmy ručně nebo provést import

Kód: **BUS\_004**

Chyba: „Nejdříve vytvořte smlouvu“

Příčina: Pokus o přestup bez existující smlouvy

Řešení: Vytvořit smlouvu v sekci Smlouvy

## 7.4 Importní chyby

Kód: **IMP\_001**

Chyba: „Import selhal: Neplatný formát“

Příčina: Nesprávná struktura CSV nebo JSON souboru

Řešení: Zkontrolovat formát a strukturu souboru

Kód: **IMP\_002**

Chyba: „Soubor nenalezen“

Příčina: Zadaná cesta k souboru neexistuje

Řešení: Vybrat existující soubor

Kód: **IMP\_003**

Chyba: „Neplatné datum“

Příčina: Nesprávný formát data v CSV souboru

Řešení: Použít formát DD.MM.RRRR

Kód: **IMP\_004**

Chyba: „Neplatné číslo“

Příčina: Textová hodnota místo čísla

Řešení: Zkontrolovat číselné sloupce v importovaném souboru

## 7.5 Řešení běžných problémů

Problém: **Aplikace se nespustí**

1. Zkontrolovat přítomnost souboru config.json
2. Ověřit, zda běží SQL Server

3. Zkontrolovat instalaci ODBC Driveru
4. Spustit aplikaci z příkazové řádky pro zobrazení chyb

Problém: „**Připojení selhalo**“

1. Ověřit dostupnost serveru pomocí příkazu ping
2. Ověřit otevřený port pomocí telnet na port 1433
3. Zkontrolovat firewall
4. Ověřit povolení TCP/IP v SQL Configuration Manageru

Problém: „**Transakce selhala**“

1. Zkontrolovat integritu databáze
2. Ověřit FOREIGN KEY vztahy
3. Zkontrolovat oprávnění databázového uživatele
4. Resetovat databázi pomocí funkce Import/Reset

## 8. KNIHOVNY TŘETÍCH STRAN

### 8.1 Python standardní knihovny

Knihovna	Účel	Licence
tkinter	GUI framework	PSF
json	Práce s JSON	PSF

csv	Import CSV	PSF
datetime	Práce s datумы	PSF
os	Práce se soubory	PSF
sys	Systémové funkce	PSF

## 8.2 Externí závislosti

### pyodbc

- **Verze:** 5.2.0+
- **Autor:** Michael Kleehammer
- **Účel:** ODBC databázové připojení pro Python
- **Licence:** MIT License
- **URL:** <https://github.com/mkleehammer/pyodbc>
- **Instalace:** `pip install pyodbc`
- **Použití:** Připojení k Microsoft SQL Server

## 9. ZÁVĚR

### 9.1 Shrnutí projektu

Football Roster Manager je plně funkční databázová aplikace pro správu fotbalových týmů a hráčů. Projekt úspěšně implementuje všechny požadavky zadání D1 s využitím DAO design patternu a Three-Tier architektury.

#### Klíčové vlastnosti:

- Kompletní CRUD operace pro hráče
- Transakční přestupy hráčů mezi týmy
- Agregované statistické reporty
- Import dat z CSV a JSON
- Validace vstupů
- Ošetření chyb

### 9.2 Splnění požadavků

Požadavek		Implementace
DAO Pattern	->	PlayerDAO, TeamDAO, ContractDAO, PositionDAO
5+ tabulek	->	6 tabulek (Team, Player, Position, Contract, PlayerTeam, PlayerContract)

2+ VIEW	->	4 pohledy (V_TeamRoster, V_TeamStatistics, V_CurrentRoster, V_PlayerContracts)
M:N vazba	->	PlayerContract (Player $\longleftrightarrow$ Contract)
Všechny datové typy	->	FLOAT, BIT, ENUM, VARCHAR, DATE
Operace nad 2+ tabulkami	->	transfer_player() (PlayerTeam + PlayerContract)
Transakce	->	BEGIN/COMMIT/ROLLBACK v transfer_player()
Report ze 3+ tabulek	->	V_TeamStatistics (5 tabulek, agregace)
Import CSV	->	import_players_from_csv()
Import JSON	->	import_teams_from_json()
Konfigurační soubor	->	config.json
Ošetření chyb	->	Try-catch všude, validace vstupů

### 9.3 Možná rozšíření

#### Budoucí vylepšení:

- Export do PDF/Excel
- Grafické vizualizace statistik
- Historie přestupů
- REST API pro mobilní aplikace
- Automatické generování reportů
- Automatické plnění dat z externího API

*Verze: 1.0*

*Datum: 3.1. 2025*

*Autor: Jakub Šrámek*

*Škola: Střední průmyslová škola elektrotechnická Ječná, Praha 2, Ječná 30*