

SPŠE Ječná
Informační technologie
Ječná 30, 121 36, Praha 2

Systém pro správu knihovny

Marie Borisová, borisova@spsejecna.cz
2025/2026

Tento projekt je zpracován jako školní projekt v rámci předmětu Programové vybavení na SPŠE Ječná

Obsah

1. Analýza požadavků	3
1.1 Funkční požadavky	3
1.2 Nefunkční požadavky	4
2. Návrh	4
2.1 Návrh architektury aplikace	4
2.2 E-R diagram databáze	5
3. Implementace	5
3.1 Použité rozhraní a protokoly	5
3.2 Seznam externích knihoven a závislostí	5
3.3 Konfigurace programu	6
3.4 Instalace a spuštění aplikace	6
3.5 Validace a ošetření vstupů	6
4. Testování	6

1. Analýza požadavků

Cílem aplikace je poskytnout systém pro spravování knihovny pomocí konzolového uživatelského rozhraní. Systém je určen pro administrátora knihovny. Níže jsou uvedeny veškeré funkční a nefunkční požadavky.

1.1 Funkční požadavky

- FR1: Uživatel může přidat novou knihu do databáze.
- FR2: Uživatel může zobrazit seznam všech knih.
- FR3: Uživatel může smazat knihu podle ID.
- FR4: Uživatel může vytvořit novou výpůjčku
- FR5: Uživatel může vrátit knihu
- FR6: Uživatel může smazat výpůjčku
- FR7: Uživatel může upravit existující výpůjčku
- FR8: Systém musí detekovat knihy po termínu a označit je jako overdue.
- FR9: Uživatel může přidat nového čtenáře.
- FR10: Uživatel může zobrazit seznam všech čtenářů.
- FR11: Uživatel může smazat čtenáře podle ID.
- FR12: Uživatel může přidat nového autora.
- FR13: Uživatel může zobrazit seznam všech autorů.
- FR14: Uživatel může smazat autora podle ID.
- FR15: Uživatel může přidat nový žánr.
- FR16: Uživatel může zobrazit seznam všech žánrů.
- FR17: Uživatel může smazat žánr podle ID.
- FR18: Uživatel může importovat data do tabulek books a authors z CSV souboru.
- FR19: Systém musí validovat formát a typy dat při importu.
- FR20: Uživatel může zobrazit agregované statistiky
- FR21: Uživatel může zobrazit 2 dostupné pohledy
- FR22: Vytvoření výpůjčky a návrat knihy probíhá v transakci nad více tabulkami
- FR23: Systém načítá přihlašovací údaje k databázi z JSON konfiguračního souboru.
- FR24: Konfigurační údaje jsou validovány stejně jako uživatelský vstup

1.2 Nefunkční požadavky

- NFR1: Přístup k databázi je omezený na uživatele lib_admin.
- NFR2: Systém musí zachovat konzistenci databáze při chybě (rollback transakcí).
- NFR3: Při chybě načtení konfiguračního souboru musí systém vypsat srozumitelnou chybu a ukončit se.
- NFR4: Konzolové UI musí být přehledné a umožňovat intuitivní navigaci mezi hlavním menu a submenu.
- NFR5: Uživateli se zobrazují jasné chybové hlášky při špatném zadání vstupů.

2. Návrh

2.1 Návrh architektury aplikace

Níže je popsána struktura vrstev aplikace.

1. UI vrstva

- Zobrazuje uživateli informace v konzolovém rozhraní
- Zpracovává a validuje vstup od uživatele
- Volá DAO metody pro operace s tabulkami

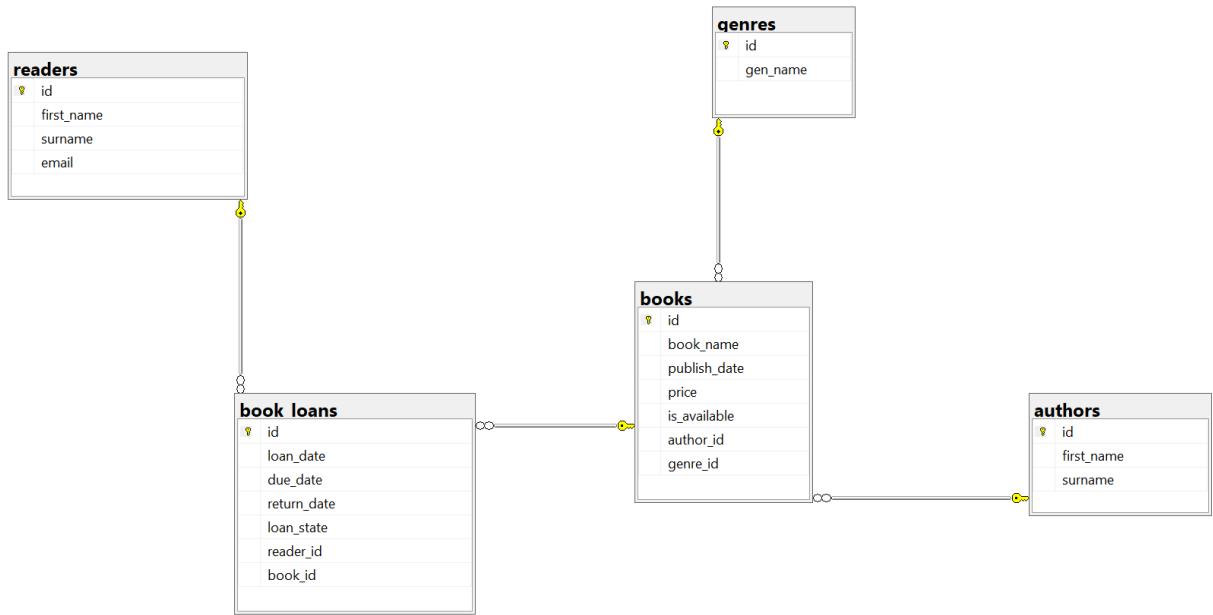
2. Aplikační vrstva

- Využívá návrhový vzor DAO
- Provádí CRUD operace a transakce
- Odděluje přístup k databázi

3. Databázová vrstva

- Využívá návrhový vzor Singleton
- Poskytuje samotné připojení k databázi

2.2 E-R diagram databáze



Obr. 1: E-R diagram databáze

3. Implementace

3.1 Použité rozhraní a protokoly

Pro komunikaci s Microsoft SQL Server je použito rozhraní ODBC. Konfigurační soubor je ve formátu JSON. Soubory pro importování dat do databáze jsou ve formátu CSV.

3.2 Seznam externích knihoven a závislostí

V aplikaci jsou použity níže uvedené Python knihovny a balíčky:

- pyodbc - pro připojení k Microsoft SQL Server
- csv - pro práci s CSV soubory
- datetime - pro práci s daty

- os - pro práci se soubory

3.3 Konfigurace programu

Aplikace je konfigurovatelná přes JSON soubor **db_config.json**, který obsahuje atributy server, database, username, password a driver. Hodnoty v konfiguračním souboru jsou validovány přes **ConfigLoader** třídu, která kontroluje správnost datových typů a existenci všech povinných atributů v souboru. Při chybné konfiguraci aplikace vypíše do konzole error a ukončí se běh programu.

3.4 Instalace a spuštění aplikace

Postup instalace a spuštění aplikace je popsán v README.md souboru projektu.

3.5 Validace a ošetření vstupů

Uživatelský vstup je validován a ošetřen ve třídě **ConsoleUI**, kde jsou metody **input_non_empty_stringy**, **input_int**, **input_float**, **input_date** které kontrolují, zda-li uživatel zadal požadovaný datový typ a formát. Při chybě ve vstupu aplikace vypisuje chybové hlášky, které jsou srozumitelné pro uživatele. Co se týče validace konfiguračního souboru, již bylo zmíněno v předešlé podkapitole, že se validuje přes třídu **ConfigLoader**.

4. Testování

Aplikace je testována prostřednictvím testovacích scénářů. Testovací scénáře a jejich specifické provedení jsou obsažené v projektu v adresáři **test**.