

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
2.3 Behaviorální diagramy	6
3. Implementace	7
3.1 Použité rozhraní a protokoly	7
3.2 Schéma importovaných souborů	7
3.2.1 Importování knih	7
3.2.2 Importování autorů	7
3.3 Seznam externích knihoven a závislostí	8
3.4 Konfigurace programu	8
3.5 Instalace a spuštění aplikace	8
3.6 Chybové stavy aplikace	8
3.6.1 Chyby databázového spojení	8
3.6.2 Chyby konfigurace	8
3.6.3 Chyby uživatelského vstupu	9
3.6.3 Chyby při importování dat	9
3.7 Validace a ošetření vstupů	9
4. Testování	9

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: Uživatelé 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

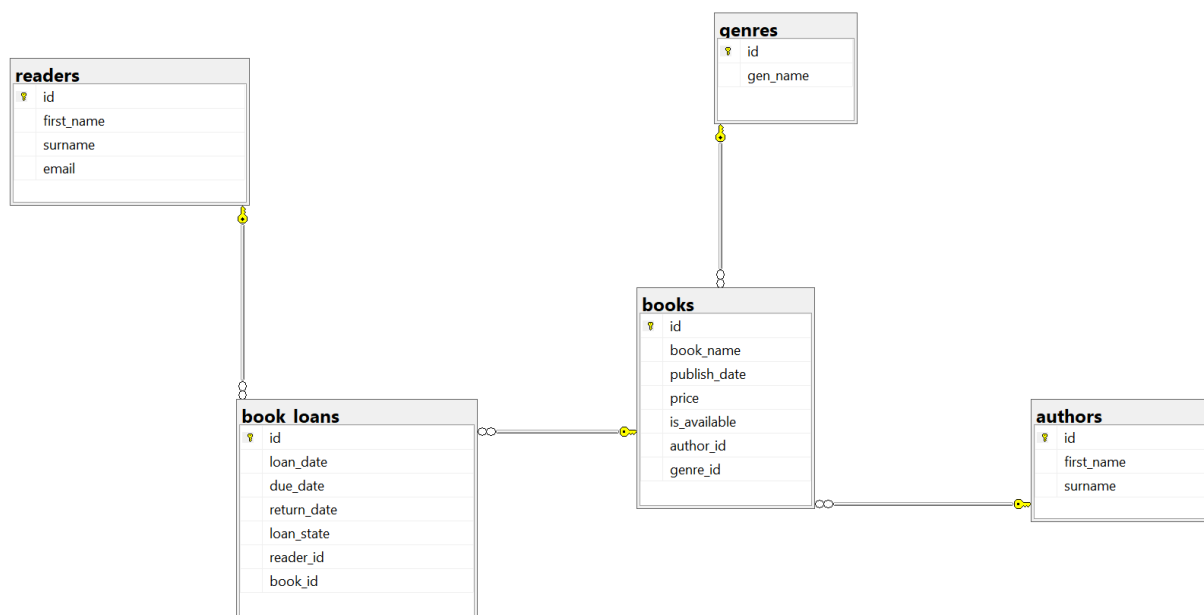
2. Aplikační vrstva

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

3. Databázová vrstva

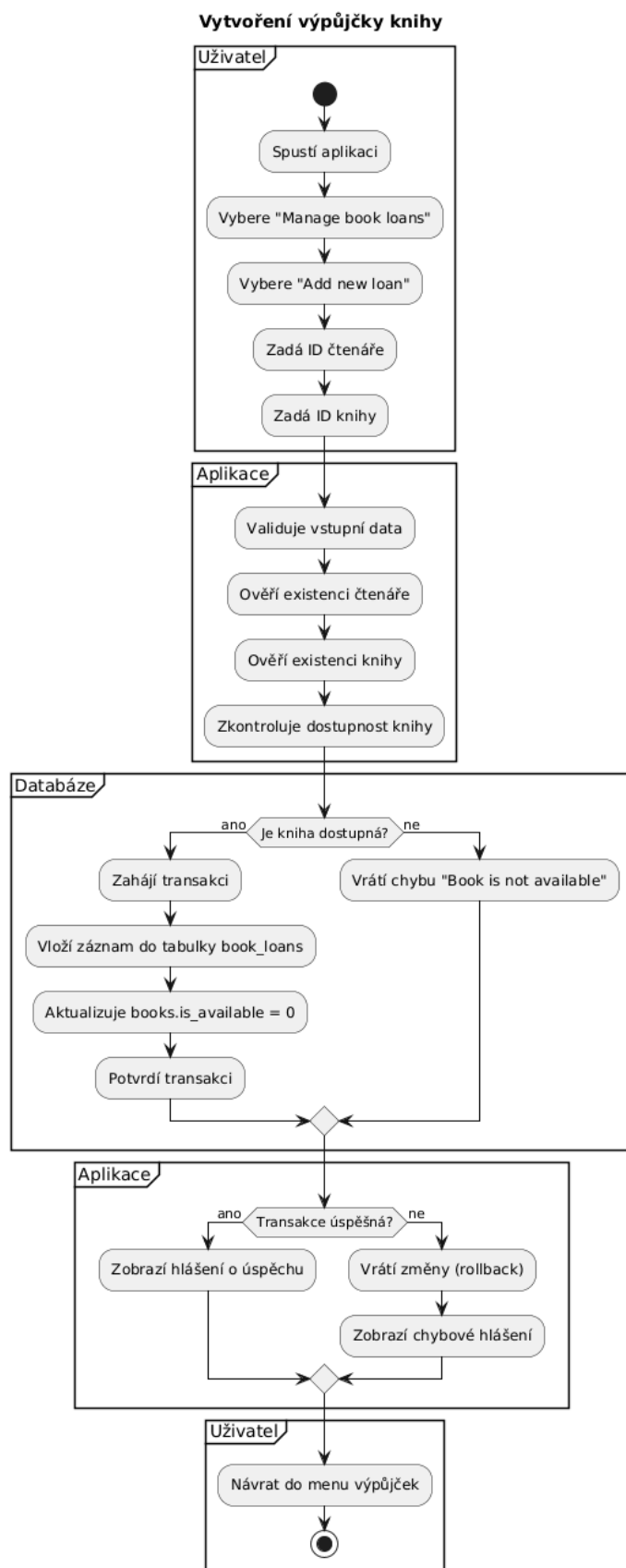
- Využívá návrhový vzor Singleton
- Poskytuje samotné připojení k databázi

2.2 E-R diagram databáze



Obr. 1: E-R diagram databáze

2.3 Behaviorální diagramy



Obr. 2: Activity diagram znázorňující postup vytvoření výpůjčky

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 Schéma importovaných souborů

Aplikace umožňuje import dat do databáze z CSV souborů. Import slouží k hromadnému vkládání dat bez nutnosti ručního zadávání přes uživatelské rozhraní.

3.2.1 Importování knih

Povinné položky pro importování knih z CSV souboru jsou **book_name** (string), **publish_date** (datum), **price** (float), **author_id** (int) a **genre_id** (int). Omezení jsou následující:

- **publish_date** musí být ve formátu YYYY-MM-DD
- **author_id** musí existovat v tabulce **authors**
- **genre_id** musí existovat v tabulce **genres**
- **price** musí být ≥ 0

Příklad obsahu CSV souboru pro importování knih:

```
book_name,publish_date,price,author_id,genre_id
1984,1949-06-08,299.90,1,2
Pýcha a předsudek,1813-01-28,249.50,2,3
```

3.2.2 Importování autorů

Povinné položky pro importování autorů z CSV souboru jsou **first_name** (string) a **surname** (string). Omezení jsou následující:

- **first_name** nesmí být prázdná hodnota
- **surname** nesmí být prázdná hodnota

Příklad obsahu CSV souboru pro importování autorů:

```
first_name,surname
George,Orwell
Jane,Austen
```

3.3 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

3.4 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.

3.5 Instalace a spuštění aplikace

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

3.6 Chybové stavy aplikace

3.6.1 Chyby databázového spojení

Database connection failed: Chyba nastane když databázový server není dostupný nebo jsou chybně nastavené přihlašovací údaje. Aplikace vypíše chybové hlášky obsahující popis chyby z ODBC ovladače a nepokračuje v běhu.

3.6.2 Chyby konfigurace

Config file not found: Chyba nastane když soubor db_config.json při načítání konfiguračního souboru neexistuje nebo není dostupný.

Missing config key: Chyba nastane když soubor neobsahuje všechny povinné atributy.

Invalid type: Chyba nastane když konfigurační soubor obsahuje nevalidní datové typy u atributů.

Config value must not be empty: Chyba nastane když hodnota atributu je prázdná.

3.6.3 Chyby uživatelského vstupu

Given ID does not exist: Chyba nastane když uživatel v aplikaci zadá ID záznamu, které v databázi neexistuje.

Neplatný datový typ vstupu (obecné označení): Chyba nastane když uživatel zadá hodnotu nesprávného datového typu (např. text místo čísla). Uživatel je vyzván k opětovnému zadání hodnoty.

3.6.3 Chyby při importování dat

File not found: Chyba nastane když uživatel zadá název CSV souboru, který se nenachází v projektovém adresáři.

3.7 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`**.