Dokumentace projektu:

Knihovní Systém (LibraryDB)

Název projektu: Library Database System

Autor: Martin Pop

Kontakt: pop@spsejecna.cz

Datum vypracování: 9.1 2026

Škola: SPŠE Ječná

Typ projektu: Školní databázový projekt – D1

1. Specifikace požadavků a účel aplikace

Cílem projektu je vytvoření desktopové/webové aplikace pro správu malé až středně velké knihovny. Systém umožňuje knihovníkovi spravovat knihy, evidovat čtenáře a vytvářet vypůjčení.

Funkční požadavky (Use Cases)

Aplikace pokrývá následující klíčové případy užití:

- **Správa katalogu:** Přidávání, editace a mazání autorů a knižních titulů.
- **Evidence výtisků:** Každý titul může mít více fyzických kopií (skladová evidence).
- **Správa čtenářů:** Registrace nových čtenářů a správa jejich údajů.
- **Výpůjční proces:**
    - Vytvoření nové výpůjčky (kontrola dostupnosti výtisku).
    - Vrácení knihy (změna statusu výtisku zpět na "Dostupný").
- **Přehledy a statistiky:** Zobrazení aktuálního stavu knihovny (počet výpůjček, hodnota knihovny) na Dashboardu.

2. Architektura aplikace

Aplikace je navržena podle vrstvené architektury, která separuje logiku zobrazení, byznys logiku a přístup k datům. Implementace je realizovaná v prostředí frameworku Flask.

Hlavní komponenty systému:

1. **Prezentační vrstva (View/Templates):**
    - Zajišťuje interakci s uživatelem pomocí HTML5 šablon (Jinja2) a CSS stylů.
2. **Aplikační vrstva (Controllers/Blueprints):**
    - Zpracovává HTTP požadavky, vezme vstupy od uživatele a volá příslušné služby.
    - Neobsahuje žádnou SQL logiku.
3. **Servisní vrstva (Services):**
    - Obsahuje byznys logiku aplikace (např. "Je tato kniha dostupná?", "Může si tento uživatel půjčit další knihu?").
    - Validuje data (např. "Má jméno alespoň 2 znaky?").
    - Tvoří most mezi Controllerem a DAO.
4. **Datová vrstva (DAO - Data Access Objects):**
    - Zajišťuje přímou komunikaci s databází MS SQL Server.
    - Provádí CRUD operace a mapuje databázové řádky na Python objekty/slovníky.

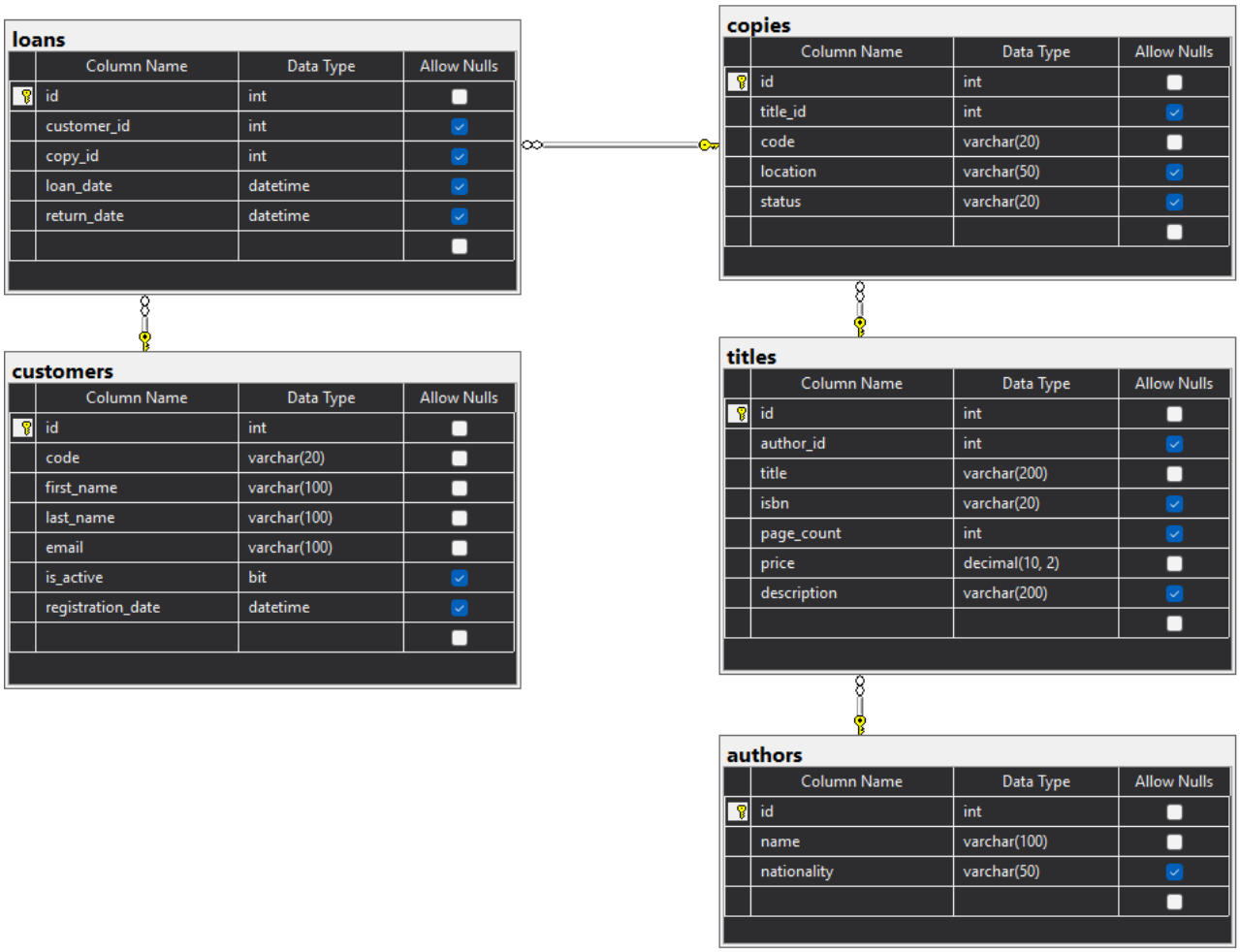
3. Popis chování aplikace (Behaviorální model)

Aplikace funguje na principu "Request-Response". Níže je popsán průběh klíčového procesu – **Výpůjčka knihy**.

1. **Požadavek:** Uživatel ve formuláři vybere čtenáře a konkrétní kopii knihy.
2. **Validace (Controller):** Systém ověří, zda jsou vstupní data kompletní.
3. **Logická kontrola (Service):**
    - Servisa LoanService se dotáže databáze, zda má daná kopie status Available.
    - Pokud je status On Loan nebo Lost, proces je zastaven a vyvolána výjimka.
4. **Transakce (DAO):**
    - Vytvoří se záznam v tabulce loans (datum půjčení = dnes).
    - Aktualizuje se záznam v tabulce copies (změna statusu na On Loan).
5. **Odpověď:** Uživatel je přesměrován na seznam výpůjček a zobrazí se zpráva o úspěchu.

4. Datový model (E-R Diagram)

Data jsou ukládána v relační databázi **Microsoft SQL Server**. Struktura je normalizována a obsahuje následující entity:



5. Konfigurace aplikace

Konfigurace je oddělena od zdrojového kódu a nachází se ve složce config/. Aplikace využívá formát JSON.

Soubor webserver_config.json

Slouží k nastavení sítě a zabezpečení webového serveru.

JSON

```
{
  "host": "127.0.0.1",              // IP adresa serveru (např. localhost)
  "port": 8080,                     // Port (rozsah 1024-65535)
  "secret_key": "tajne_heslo"       // Klíč pro šifrování
}
```

Soubor db_config.json

Slouží k nastavení připojení k MS SQL Serveru.

JSON

```
{
  "driver": "ODBC Driver 17 for SQL Server",
  "server": "localhost",                          // Adresa SQL serveru
  "database": "LibraryDB",                        // Název databáze
  "uid": "library_user",                          // Login
  "pwd": "pass123"                                // Heslo loginu
}
```

6. Instalace a spuštění

Detailní postup instalace a požadavky jsou popsán v souboru **README.md** v kořenovém adresáři projektu.

7. Chybové stavy a jejich řešení

Aplikace implementuje ošetření chyb a logování.

Typické chybové stavy:

1. **Chyba připojení k DB:**
    - *Příznak:* Aplikace po spuštění se načte ale nenačte seznam knih.
    - *Chyba:* Místo knih se načte stránka s varováním a konkrétní chybou.
    - *Řešení:* Zkontrolujte db_config.json a zda běží služba SQL Server.
2. **Nevalidní konfigurace portu:**
    - *Příznak:* Aplikace se nespustí.
    - *Log:* V server_errors.log bude chybová hláška.
    - *Řešení:* Port v webserver_config.json musí být číslo mezi 0 a 65535.
3. **Logické chyby (např. půjčení již půjčené knihy):**
    - *Reakce:* Aplikace zobrazí uživateli chybovou hlášku (Flash message) přímo v prohlížeči a nedovolí operaci dokončit.

Logy jsou ukládány do souboru server_errors.log v kořenovém adresáři (vytvoří se až po spuštění).

8. Použité technologie a knihovny třetích stran

Projekt je postaven na jazyce **Python** a využívá následující externí knihovny:

- **Flask:** Framework pro webové rozhraní a routing.
- **pyodbc:** Konektor pro komunikaci s databází MS SQL Server.
- **pyinstaller:** Kompiluje projekt do samostatně spustitelného souboru.

9. Závěr

Tento projekt demonstruje vytvoření funkčního informačního systému s důrazem architekturu, oddělení vrstev a bezpečný přístup k datům. Aplikace splňuje všechny stanovené požadavky, umožňuje správu knihovních procesů a je připravena k nasazení v lokální síti. Kód umožňuje snadné budoucí rozšíření o nové funkce.