Technická dokumentace projektu

Projekt: Library database **Autor:** Alexandre Basseville C4b **Technologie:** Python 3.10, MSSQL, PyODBC

1. Architektura systému

Aplikace je navržena jako konzolová utilita využívající **návrhový vzor Active Record** (D2).

- Každá tabulka v databázi má svou odpovídající třídu v Pythonu (ve složce `src/models`).
- Třídy dědí ze společného předka `BaseModel`, který zajišťuje základní CRUD operace (Save, Find, All).
- Databázová vrstva je oddělena v `src/database.py` a využívá Singleton pattern pro připojení.

2. Databázové schéma (ER Model)

Databáze se skládá z 5 tabulek a relací:

1. **Books:** Hlavní tabulka knih (Title, Year, Price, IsDamaged).
2. **Authors:** Tabulka autorů.
3. **BookAuthors:** Vazební tabulka pro relaci **M:N** mezi knihami a autory.
4. **Members:** Čtenáři knihovny.
5. **Loans:** Výpůjčky. Obsahuje Cizí klíče na `Books` a `Members`.

Systém také využívá SQL Views (`View_BookDetails`) pro generování reportů.

3. Klíčové algoritmy

Transakce: Vrácení knihy

Pro splnění požadavku na bezpečný zápis do více tabulek je implementována metoda `return_book_transaction` (v `src/services.py`).

- Využívá `conn.autocommit = False`.
- V rámci jedné transakce se provede UPDATE tabulky `Loans` (vrácení) a případný UPDATE tabulky `Books` (poškození).
- Pokud dojde k chybě, provede se `ROLLBACK`.

Import dat

Třída `DataImporter` čte CSV soubor, dynamicky vytváří objekty `Author` a `Book` a automaticky vytváří záznamy ve vazební tabulce.

4. Konfigurace

Připojení k databázi je externalizováno v souboru `config/db_config.ini`, což umožňuje snadné přepínání mezi lokálním vývojem (Docker) a produkčním nasazením (Školní SQL Server).