

Dokumentace projektu – Půjčovna aut

1. Základní informace

Název projektu:

Půjčovna aut (Car Rental Database Application)

Autor:

Václav Křivka

E-mail: *krivka@spsejecna.cz*

Škola:

SPŠE Ječná

Datum vypracování:

leden 2026

Typ projektu:

Školní projekt – databázová aplikace

2. Cíl projektu

Cílem projektu je vytvořit **konzolovou databázovou aplikaci v jazyce Python**, která slouží k evidenci půjčovny aut.

Aplikace využívá **relační databázi Microsoft SQL Server** a je navržena s důrazem na **správnou architekturu, návrhové vzory a oddělení logiky od dat**.

Projekt demonstruje:

- práci s relační databází,
 - použití návrhových vzorů **DAO + Mapper (D1 + D3)**,
 - vazby **1:N a M:N**,
 - import dat z CSV,
 - konfiguraci aplikace pomocí konfiguračního souboru.
-

3. Specifikace požadavků (Use Case – slovní popis)

Uživatel může:

- zobrazit seznam zákazníků
 - přidat / upravit / smazat zákazníka
 - zobrazit seznam aut
 - přidat / upravit / smazat auto
 - evidovat značky aut
 - vytvořit smlouvu o zapůjčení
 - přiřadit více aut k jedné smlouvě (M:N)
 - načíst data z CSV souborů
 - pracovat s aplikací bez přímé znalosti SQL
-

4. Architektura aplikace (návrhové vzory)

Aplikace je rozdělena do logických vrstev:

Použité návrhové vzory

DAO (Data Access Object)

Každá databázová tabulka má vlastní DAO třídu:

- CustomerDAO
- CarDAO
- BrandDAO
- ContractDAO
- Contract_carDAO

DAO zajišťují:

- CRUD operace
- izolaci SQL dotazů

- jednoduché rozhraní pro práci s daty

Mapper (Object–Relational Mapping)

Třída Mapper:

- mapuje Python objekty na databázové tabulky
- automaticky určuje název tabulky a primárního klíče
- provádí obecné operace (insert, update, delete, get_by_id, get_all)

Databázové záznamy jsou reprezentovány pomocí **Python dataclass** objektů.

5. Behaviorální popis aplikace

Tok aplikace (Activity – slovní popis)

1. Aplikace se spustí
2. Načte konfigurační soubor rental_config.json
3. Připojí se k databázi
4. Uživatel volí akce (CRUD operace)
5. DAO vrstva komunikuje s databází
6. Výsledky jsou vráceny uživateli

Stavové chování (State – slovní popis)

- Objekt je vytvořen (bez ID)
 - Objekt je uložen do DB (má ID)
 - Objekt může být aktualizován
 - Objekt může být smazán
-

6. E-R model databáze

Tabulky:

brand

- id_brand (PK, INT)
- name (VARCHAR)
- category (VARCHAR, CHECK)

car

- id_car (PK)
- license_plate (UNIQUE)
- model
- price_per_day
- is_available
- id_brand (FK → brand)

customer

- id_customer (PK)
- name
- surname
- email

contract

- id_contract (PK)
- id_customer (FK → customer)
- date_from
- date_to
- total_price

contract_car (M:N)

- id_contract (FK)
 - id_car (FK)
-

7. Import a export dat

Import

Aplikace podporuje import dat z CSV:

- zákazníci
- auta
- značky

Povinné položky:

- musí odpovídat názvům atributů datových tříd
- soubor musí být ve formátu UTF-8

Export

Export dat není implementován (není požadováno zadáním).

8. Konfigurace aplikace

Konfigurace je řešena pomocí souboru:

`rental_config.json`

Povolené volby:

- driver – ODBC driver
- server – SQL Server instance
- database – název DB
- uid – SQL uživatel
- pwd – heslo

Soubor není součástí repozitáře (bezpečnost).

9. Instalace a spuštění

Požadavky:

- Python 3.10+
- MS SQL Server / LocalDB
- ODBC Driver 17 for SQL Server

Postup:

1. Vytvořit databázi pomocí SQL skriptu
2. Vytvořit SQL uživatele
3. Vytvořit rental_config.json
4. Spustit aplikaci:

```
python main.py
```

10. Chybové stavy

Chyba	Popis	Řešení
Chybí config	Soubor neexistuje	Vytvořit podle vzoru
SQL chyba	Porušení constraintu	Opravit vstupní data
Nenalezen záznam	Neexistující ID	Ověřit vstup

Chyby jsou zachytávány pomocí výjimek.

11. Použité knihovny

- pyodbc – komunikace s MS SQL
 - dataclasses – reprezentace entit
 - csv – import dat
 - json – konfigurace
 - pathlib, os – práce se soubory
-

12. Závěrečné shrnutí

Projekt splňuje všechny požadavky zadání.

Byla vytvořena funkční databázová aplikace s čistou architekturou, využívající návrhové vzory DAO a Mapper.

Projekt je připraven k dalšímu rozšiřování (UI, export, validace).