

# DOKUMENTACE PROJEKTU: TATTOO PLANNER

**Autor:** Zuzana Pospíšilová

**Email:** z.pospisilova06@gmail.com

**Škola:** SPŠE Ječná

**Datum:** 9.1.2026

---

## 1. ZÁKLADNÍ INFORMACE O PROJEKTU

Projekt Tattoo Planner je konzolová databázová aplikace napsaná v jazyce Python. Aplikace slouží ke správě tetovacího studia a umožňuje evidenci zákazníků, tatérů, stylů tetování a především správu rezervací termínů.

### Cíl projektu:

- Práce s relační databází.
- Návrhový vzor DAO (Data Access Object).
- Použití databázových transakcí.
- Import externích dat.
- Tvorbu reportů pomocí databázových pohledů (VIEW).

Databáze je realizována v Microsoft SQL Serveru a aplikace s ní komunikuje pomocí ODBC rozhraní.

---

## 2. SPECIFIKACE POŽADAVKŮ A USE CASES

Správa zákazníků umožňuje evidovat základní kontaktní údaje a následně je využívat při vytváření rezervací. Správa tatérů a stylů tetování slouží k přehledné organizaci nabídky studia. Rezervace termínů představují klíčovou část aplikace, protože propojují zákazníka, tatéra i konkrétní styl tetování do jednoho celku.

Import dat a reporting zvyšují praktičnost aplikace a umožňují rychlé naplnění databáze daty a

získání přehledných informací.

**Funkční požadavky aplikace:**

- Správa zákazníků
  - Správa tatérů
  - Správa stylů tetování
  - Rezervace termínů
  - Import dat
  - Reporting
- 

### 3. ARCHITEKTURA APLIKACE

Prezentační vrstva zajišťuje komunikaci s uživatelem prostřednictvím konzole.

Aplikační vrstva obsahuje hlavní logiku aplikace, kontrolu vstupů a práci s databázovými transakcemi. Datová vrstva je realizována pomocí DAO tříd, které se starají o komunikaci s databází a oddělují SQL dotazy od zbytku aplikace.

Takové řešení odpovídá návrhovému vzoru DAO (D1).

**Prezentační vrstva:**

- src/main.py – konzolové rozhraní

**Aplikační vrstva:**

- business logika, validace, transakce

**Datová vrstva:**

- DAO třídy ve složce src/dao

**Databáze:**

- Microsoft SQL Server
- 

### 4. HLAVNÍ PROCES – VYTVOŘENÍ REZERVACE

Proces vytvoření rezervace je navržen jako transakční operace, aby byla zajištěna konzistence dat v databázi. To znamená, že všechny kroky rezervace musí proběhnout

úspěšně, jinak se žádná data neuloží.

**Rezervace probíhá transakčně:**

- výběr zákazníka
  - výběr tatéra
  - výběr stylu
  - výpočet ceny
  - uložení s COMMIT
- 

## 5. DATABÁZOVÝ MODEL

**Hlavní entity:**

- Customers
  - TattooArtists
  - TattooStyles
  - Appointments
  - Reviews
- 

## 6. IMPORT DAT

Import dat slouží k rychlému naplnění databáze testovacími nebo počátečními daty. Aplikace umožňuje import tatérů ze souboru CSV a zákazníků ze souboru JSON.

- artists.csv – tatéři
  - customers.json – zákazníci
- 

## 7. KONFIGURACE

- config.json – DB připojení a nastavení aplikace
- 

## 8. CHYBOVÉ STAVY

Aplikace je navržena tak, aby dokázala reagovat na běžné chyby, ke kterým může během provozu dojít. Mezi tyto chyby patří například problémy s připojením k databázi, neplatné

vstupy od uživatele.

- chyba DB připojení
  - neplatný vstup
  - porušení integrity dat
- 

## 9. POUŽITÉ KNIHOVNY

Pro komunikaci s databází Microsoft SQL Server je využita knihovna pyodbc, která umožňuje provádět SQL dotazy a pracovat s databázovými transakcemi. Knihovna reportlab je použita pro případné generování výstupů a reportů.

- pyodbc
  - reportlab
- 

## 10. ZÁVĚR

Aplikace demonstruje práci s relační databází, návrhovým vzorem DAO, databázovými transakcemi a importem dat. Výsledkem je funkční a přehledná databázová aplikace, která může sloužit jako praktická ukázka propojení Python aplikace s databází Microsoft SQL Server.