MilWare – Military database systém 1.0

Autor: David Čadek

Datum: 9.1.2026

TESTOVACÍ SCÉNÁŘ č. 2: Ověření funkčnosti a manipulace s daty (B)

Obsah

1. Cíl testu

Cílem tohoto scénáře je ověřit stabilitu systému při administrativních úkonech. Test simuluje práci správce databáze, který provádí hromadné importy, manuální korekce dat (CRUD), přidělování složitých úkolů (Mise) a ověřuje, zda systém správně reaguje na nestandardní vstupy a výpadky infrastruktury.

2. Prerekvizity (Co musíte mít připraveno)

- Aplikace je nainstalována a spuštěna (python src/main.py).
- Databáze milware_db je aktivní a obsahuje základní strukturu (init_db.sql).
- V databázi jsou aktivní pohledy (views_db).
- Máte přístup k editaci souborů ve složce data.

3. Postup testu (Krok za krokem)

Část A: Validace externích dat (JSON Import)

Ověříme, zda systém akceptuje změny v datových souborech provedené "zvenčí".

1. **Příprava dat:**
    - Nechte aplikaci běžet na pozadí.
    - Ve složce data otevřete soubor new_vehicles.json.
    - **Akce:** Najděte libovolný záznam a změňte položku full_name na **"Admin Testovací"**.
    - Soubor uložte a zavřete.
2. **Import:**
    - V aplikaci zvolte menu **8** (IMPORT DATA).
    - Zvolte **1** (Importovat VOJÁKY).
    - **Očekávaný výsledek:** Program vypíše [ÚSPĚCH].
3. **Ověření:**
    - V menu zvolte **1** (VÝPIS) -> **1** (Vojáci).
    - Zkontrolujte, zda tabulka obsahuje vojáka "Admin Testovací".
    - *Kontrola:* Data se načítají dynamicky, import funguje.

Část B: Správa personálu (CRUD operace)

Simulace příjmu nového specialisty a jeho následná úprava.

1. **Vytvoření (CREATE):**
    - Menu **3** (Rekrutovat).
    - **Jméno:** Chuck Norris
    - **Callsign:** Ranger
    - **Hodnost:** Private
    - **ID Základny:** 1
    - **Výsledek:** Program potvrdí přidělení ID.
2. **Úprava (UPDATE):**
    - Menu **4** (Povýšit/Upravit).
    - Zadejte ID nového vojáka (Chucka Norrise).
    - **Nové jméno:** (Enter - beze změny).
    - **Nová hodnost:** General (okamžité povýšení).
    - **Výsledek:** Program potvrdí aktualizaci.
3. **Kontrola:**
    - Menu **2** (Najít vojáka) -> zadejte ID. Ověřte, že hodnost je General.

Část C: Nasazení do akce (Vazba M:N)

1. **Přiřazení:**
    - Menu **6** (POSLAT NA MISI).
    - Vyberte ID vojáka (Chuck Norris) a ID mise.
    - **Role:** Hlavní vyjednavač.
    - **Výsledek:** Program vypíše [ÚSPĚCH] Rozkaz potvrzen.
2. **Kontrola vazby:**
    - Menu **1** -> **5** (Rozkazy). Záznam musí existovat.

Část D: Přehledy a SQL Views

1. Menu **7** (GENERÁLNÍ REPORT).
2. Zkontrolujte, zda se v tabulce správně sčítají počty vojáků na základně č. 1 (měl by tam být započítán i Chuck Norris).
3. Ověřte, že report nevypisuje chyby SQL dotazu.

Část E: TEST TRANSAKCE (Konzistence dat)

Klíčový test: Změna stavu ve dvou tabulkách najednou.

1. **Výchozí stav:**
    - Vyberte vojáka s čistým jménem (např. "Admin Testovací" z importu).
    - Ověřte, že v jeho Callsign **není** hvězdička.
2. **Transakce:**
    - Menu **6** (POSLAT NA MISI).
    - Vyberte tohoto vojáka a libovolnou misi.
    - **Role:** Pozorovatel.
    - **Výsledek:** [ÚSPĚCH] Transakce provedena.
3. **Křížová kontrola (Důkaz transakce):**
    - **Tabulka 1 (Mise):** Menu **1**->**5**. Záznam s rolí "Pozorovatel" existuje.
    - **Tabulka 2 (Vojáci):** Menu **1**->**1**. Voják má nyní Callsign **Admin Testovací*** (s hvězdičkou).
    - *Závěr:* Systém úspěšně provedl atomickou operaci nad více tabulkami.

4. Postup testu – Testování chyb a výjimek (Exceptions)

Zátěžový test odolnosti aplikace.

Test 1: Ošetření datových typů

1. V menu zvolte **5** (Propustit vojáka).
2. Program čeká číslo (ID). Zadejte text: **chyba**.
3. **Reakce:** Aplikace **nesmí spadnout**. Musí vypsat *"Neplatná hodnota, zadejte číslo"* a vrátit se do menu.

Test 2: Neexistující volby menu

1. V hlavním menu zadejte **007**.
2. **Reakce:** Aplikace vypíše *"Neplatná volba"* a znovu vykreslí menu.

Test 3: Integrita dat (Neexistující záznam)

1. Menu **4** (Upravit vojáka).
2. Zadejte ID **999999**.
3. **Reakce:** Aplikace vypíše *"Voják nenalezen"* (nikoliv databázovou chybu).

Test 4: Simulace výpadku spojení (Connection Error)

1. Nechte aplikaci běžet.
2. Ve Windows (XAMPP/Workbench) **vypněte službu MySQL**.
3. V aplikaci zvolte **1** -> **1** (Výpis vojáků).
4. **Reakce:**
    - Aplikace **NESMÍ SPADNOUT** (Python Traceback).
    - Musí vypsat: *"Chyba připojení k databázi"* (nebo podobnou hlášku z try-except bloku).
    - Aplikace se korektně ukončí nebo vrátí do menu.

5. Vyhodnocení testu

Test je hodnocen jako ÚSPĚŠNÝ (PASS), pokud:

- [ ] Importovaná data (Admin Testovací) jsou v systému viditelná.
- [ ] CRUD operace (Vytvoření Chucka Norrise) proběhla a data jsou uložena v DB.
- [ ] Transakce proběhla korektně (Voják je na misi A ZÁROVEŇ má hvězdičku u jména).
- [ ] Aplikace ustála zadání textu místo čísla (nespadla).
- [ ] Aplikace ustála výpadek databáze s uživatelsky přívětivou chybovou hláškou.

Test je hodnocen jako NEÚSPĚŠNÝ (FAIL), pokud:

- Došlo k pádu aplikace (Traceback) v kterémkoliv kroku.
- Transakce se provedla jen částečně (např. chybí hvězdička).
- Importovaná data se nezobrazila.