

# Testovací scenar 2: Testovani funkci aplikace

**Projekt:** E-Shop (D1 Repository Pattern) **Autor:** zajicek3 **Datum:** 2026

---

## Predpoklady

- ☐ Aplikace nainstalovana podle scenare 1
  - ☐ Backend bezi na `http://localhost:5000`
  - ☐ Frontend bezi na `http://localhost:3000`
  - ☐ Databaze obsahuje testovací data
- 

## Test 1: Zobrazeni produktu (READ)

### Kroky:

- Otevrit `http://localhost:3000/products`

### Ocekavany vysledek:

- Zobrazí se seznam 6 produktu
- Kazdy produkt má: název, cenu, počet na skladě
- Tlačítka "Detail" a "Přidat do košíku"

### Kontrola v API:

- GET `http://localhost:5000/api/products`
  - Odpověď: JSON pole s 6 produkty
- 

## Test 2: Přidání do košíku

### Kroky:

- Na stránce `/products` kliknout "Přidat do košíku" u produktu "iPhone 15"
- Zobrazila se hláška "Přidáno do košíku!"
- Prejít na `/cart`

### Ocekavany vysledek:

- Košík obsahuje iPhone 15
  - Zobrazena cena a množství
  - Celková cena vypočtena správně
- 

## Test 3: Vytvoření objednávky (CRUD více tabulek)

Tento test overuje požadavek: "CRUD přes více tabulek"

### Kroky:

- V košíku mít alespoň 1 produkt
- Kliknout tlačítko "Objednat"
- Sledovat terminal backendu

### Ocekavany vysledek:

- Hlaska "Objednavka vytvorena! ID: X"
- Kosik se vyprazdni
- V terminalu backendu: `POST /api/orders 201`

### Overeni v databazi (SSMS):

```
-- Nova objednávka
SELECT * FROM orders ORDER BY id DESC;

-- Polozky objednávky
SELECT * FROM order_items ORDER BY id DESC;

-- Zmena skladu (stock se snizil)
SELECT name, stock FROM products WHERE name = 'iPhone 15';
```

### Co se deje na pozadi:

1. INSERT do tabulky `orders`
2. INSERT do tabulky `order_items`
3. UPDATE tabulky `products` (snizeni stock)

### 3 tabulky v jedne transakci!

---

## Test 4: Report (agregace 3+ tabulek)

Tento test overuje pozadavek: "Report z 3+ tabulek"

### Kroky:

1. Prejit na `/admin`
2. Zkontrolovat sekci "Report"

### Ocekavany vysledek:

- Celkove trzby
- Pocet objednavek
- Pocet uzivatelu
- Prumerna hodnota objednavky
- Top produkty

### Kontrola v API:

- GET `http://localhost:5000/api/report`
  - Odpoved obsahuje agregace z tabulek: orders, users, products, order\_items, categories
- 

## Test 5: Import produktu (JSON import)

Tento test overuje pozadavek: "Import z JSON"

### Kroky:

1. Prejit na `/admin`

2. Do textoveho pole vložit:

```
{
  "products": [
    {
      "name": "Test Produkt",
      "description": "Testovací popis",
      "price": 999,
      "stock": 50,
      "category_id": 1,
      "is_featured": false
    }
  ]
}
```

3. Kliknout "Importovat"

#### Očekavany výsledek:

- Hláška "Importováno 1 produktu"
- Nový produkt se objeví v seznamu

#### Overení v databázi:

```
SELECT * FROM products WHERE name = 'Test Produkt';
```

## Test 6: Transakce - převod kreditu

Tento test overuje požadavek: "Transakce nad více tabulkami"

#### Kroky (pres API):

1. Otevřít nástroj pro API (Postman, curl, nebo prohlížeč)
2. Poslat POST request:

```
POST http://localhost:5000/api/users/transfer-credits
Content-Type: application/json

{
  "from_user_id": 1,
  "to_user_id": 2,
  "amount": 100
}
```

#### Očekavany výsledek:

- Odpověď: {"message": "Kredity převedeny"}
- Status: 200

#### Overení v databázi:

```
-- Pred prevodom: Jan ma 500, Petra ma 1000
-- Po prevodu: Jan ma 400, Petra ma 1100
SELECT name, credits FROM users WHERE id IN (1, 2);
```

---

## Test 7: Vytvoreni produktu (CREATE)

### Kroky (pres API):

```
POST http://localhost:5000/api/products
Content-Type: application/json
```

```
{
  "name": "Novy Produkt",
  "description": "Popis",
  "price": 1500,
  "stock": 20,
  "category_id": 1,
  "is_featured": true
}
```

### Ocekavany vysledek:

- Status: 201
- Odpoved: {"message": "Produkt vytvoren"}

---

## Test 8: Uprava produktu (UPDATE)

### Kroky (pres API):

```
PUT http://localhost:5000/api/products/1
Content-Type: application/json
```

```
{
  "name": "iPhone 15 Pro",
  "description": "Upraveny popis",
  "price": 34990,
  "stock": 5,
  "category_id": 1,
  "is_featured": true
}
```

### Ocekavany vysledek:

- Status: 200
- Odpoved: {"message": "Produkt aktualizovan"}

### Overeni:

- Refresh stranky /products
- Produkt ma novy nazev a cenu

---

## Test 9: Smazani produktu (DELETE)

### Kroky:

1. Prejit na `/admin`
2. V tabulce produktu kliknout "Smazat" u testovciho produktu
3. Potvrdit smazani

### Ocekavany vysledek:

- Produkt zmizi ze seznamu
  - V terminalu: `DELETE /api/products/X 200`
- 

## Test 10: Filtrovani podle kategorie

### Kroky (pres API):

```
GET http://localhost:5000/api/products?category_id=1
```

Nebo v repository:

```
ProductRepository.get_by_category(1)
```

### Ocekavany vysledek:

- Pouze produkty z kategorie Elektronika
- 

## Kontrolni seznam

Po dokonceni testu overit:

- ☐ Produkty se zobrazuji
  - ☐ Kosik funguje
  - ☐ Objednavka se vytvori (3 tabulky)
  - ☐ Report zobrazuje spravna data
  - ☐ Import JSON funguje
  - ☐ Prevod kreditu funguje (transakce)
  - ☐ CRUD operace funguji
- 

**Test PASSED** pokud vsechny kroky probehly uspesne.