

Testovací scénář 3: Testování chyb a výjimečných stavů

Projekt: Library DB System (D1)

Verze: 1.0

Datum: 2025-01-11

Předpoklady: Aplikace je nainstalována a běží (viz Scénář 1)

Test 1: Chyby databázového připojení

Krok 1.1: Odpojení databáze

1. Zastavte SQL Server (nebo odpojte síť k databázi)
2. V aplikaci zkuste kliknout na "View Books"

Očekávaný výsledek: - Chybová hláška: "Database error: ..." - Aplikace se nezhroutí - Lze se vrátit na hlavní stránku

Krok 1.2: Obnovení připojení

1. Spusťte SQL Server zpět
2. Klikněte na "View Books"

Očekávaný výsledek: - Aplikace funguje normálně

Test 2: Porušení integrity databáze

Krok 2.1: Pokus o smazání používaného autora

Příprava: - Ujistěte se, že autor "Isaac Asimov" má alespoň 1 knihu

Test: 1. Klikněte na "View Authors" 2. Klikněte "Delete" u autora "Isaac Asimov"

Očekávaný výsledek: - Chybová hláška obsahuje informaci o cizím klíči - Autor NENÍ smazán - Jeho knihy zůstávají v databázi

Krok 2.2: Pokus o půjčení neexistující knihy

1. Klikněte na "Loan Book"
2. Vyplňte:
 - Book ID: 9999 (neexistující)
 - User ID: 1
3. Klikněte "Loan Book"

Očekávaný výsledek: - Chybová hláška: "Error: Book not found" - Půjčka NENÍ vytvořena

Krok 2.3: Pokus o půjčení již vypůjčené knihy

Příprava: - Vypůjčte knihu s ID 1 uživateli s ID 1

Test: 1. Klikněte na “Loan Book” 2. Pokuste se půjčit stejnou knihu znovu: - Book ID: 1 - User ID: 1 3. Klikněte “Loan Book”

Očekávaný výsledek: - Chybová hláška: “Error: Book is currently not available” - Další půjčka NENÍ vytvořena

Test 3: Neplatné vstupy

Krok 3.1: Záporná cena

1. Klikněte na “Add Book with Categories”
2. Vyplňte:
 - Title: Test Book
 - Author: Test Author
 - Price: -50
 - Published Date: 2020-01-01
 - Zaškrtněte 1 kategorii
3. Klikněte “Add Book”

Očekávaný výsledek: - Chybová hláška: “Price must be positive” - Kniha NENÍ vytvořena

Krok 3.2: Neplatné datum

1. Klikněte na “Add Book with Categories”
2. Vyplňte:
 - Title: Test Book
 - Author: Test Author
 - Price: 100
 - Published Date: invalid-date (pokud to HTML dovolí)
 - Zaškrtněte 1 kategorii
3. Klikněte “Add Book”

Očekávaný výsledek: - HTML validace nebo chybová hláška: “Invalid date format” - Kniha NENÍ vytvořena

Krok 3.3: Kniha bez kategorie

1. Klikněte na “Add Book with Categories”
2. Vyplňte všechna pole KROMĚ kategorií (žádnou nezaškrtněte)
3. Klikněte “Add Book”

Očekávaný výsledek: - Chybová hláška: “At least one category must be selected” - Kniha NENÍ vytvořena

Krok 3.4: Email bez zavináče

1. Klikněte na “Add User”
2. Vyplňte:
 - Full Name: Jane Doe
 - Email: invalidemail.com (bez @)
3. Klikněte “Add User”

Očekávaný výsledek: - Chybová hláška: “Invalid email format” - Uživatel NENÍ vytvořen

Test 4: Chybný JSON import

Krok 4.1: Prázdný soubor

1. Vytvořte prázdný soubor `empty.json`
2. Klikněte na “Import JSON”
3. Vyberte `empty.json`
4. Klikněte “Import”

Očekávaný výsledek: - Chybová hláška: “Import error: ...” - Žádná data nejsou importována

Krok 4.2: Neplatný JSON formát

Vytvořte soubor `invalid.json`:

```
{  
  "title": "Test",  
  "missing_comma"  
  "author": "Test"  
}
```

Test: 1. Klikněte na “Import JSON” 2. Vyberte `invalid.json` 3. Klikněte “Import”

Očekávaný výsledek: - Chybová hláška: “Import error: ...” nebo “Failed to load JSON file: ...” - Žádná data nejsou importována

Krok 4.3: JSON s chybějícími povinnými poli

Vytvořte soubor `incomplete.json`:

```
[  
  {  
    "title": "Test Book",  
    "author": "Test Author"  
  }  
]
```

Test: 1. Klikněte na “Import JSON” 2. Vyberte `incomplete.json` 3. Klikněte “Import”

Očekávaný výsledek: - Chybová hláška: “Import error: ...” (chybí `price` nebo `published_date`) - Data NEJSOU importována

Krok 4.4: Nenahrání žádného souboru

1. Klikněte na “Import JSON”
2. NEVYBÍREJTE žádný soubor
3. Klikněte “Import”

Očekávaný výsledek: - Chybová hláška: “No file selected!” - Žádná akce není provedena

Krok 4.5: Nahrání non-JSON souboru

1. Vytvořte textový soubor `test.txt`
2. Klikněte na “Import JSON”
3. Vyberte `test.txt`
4. Klikněte “Import”

Očekávaný výsledek: - Chybová hláška: “File must be JSON format” - Žádná data nejsou importována

Test 5: Konfigurace - různé scénáře

Krok 5.1: Prázdné pole v `config.json`

1. Zastavte aplikaci
2. V `config/config.json` nastavte:

```
{
  "server": "",
  "database": "library_db",
  "username": "library_app",
  "password": "password",
  "driver": "ODBC Driver 17 for SQL Server"
}
```

3. Spusťte aplikaci

Očekávaný výsledek: - Chyba při startu: “FATAL ERROR: ...” - Jasná hláška, že konfigurace je neplatná

Krok 5.2: Nesprávný driver

1. V `config/config.json` změňte driver:

```
{  
  "driver": "Nonexistent Driver"  
}
```

2. Spustte aplikaci

Očekávaný výsledek: - Chyba: “FATAL ERROR: Database connection failed: ...” - Informace o tom, že driver není nalezen

Krok 5.3: Obnovení správné konfigurace

1. Opravte config/config.json na správné hodnoty
2. Spustte aplikaci

Očekávaný výsledek: - Aplikace běží normálně

Test 6: Edge cases

Krok 6.1: Velmi dlouhý text

1. Klikněte na “Add Book with Categories”
2. Vyplňte Title s 300 znaky (překročí VARCHAR(200))
3. Klikněte “Add Book”

Očekávaný výsledek: - Chybová hláška o překročení délky - Kniha NENÍ vytvořena

Krok 6.2: Speciální znaky

1. Klikněte na “Add Author”
2. Vyplňte Name: O'Brien's Books & Co.
3. Klikněte “Add Author”

Očekávaný výsledek: - Autor je vytvořen - Speciální znaky jsou správně uloženy (SQL injection prevence)

Krok 6.3: Duplicitní email

1. Vytvořte uživatele s emailem test@example.com
2. Pokuste se vytvořit dalšího uživatele se STEJNÝM emailem

Očekávaný výsledek: - Oba uživatelé jsou vytvořeni (v DB schema není UNIQUE constraint na email) - Nebo Chyba, pokud byl constraint přidán

Test 7: Transakční integrita

Krok 7.1: Současná půjčka téže knihy

Tento test vyžaduje 2 okna prohlížeče současně:

Okno 1: 1. Otevřete “Loan Book” 2. Vyplňte Book ID: 2, User ID: 1

Okno 2: 1. Otevřete “Loan Book” 2. Vyplňte Book ID: 2, User ID: 2

Současně: 3. Klikněte “Loan Book” v obou oknech téměř současně

Očekávaný výsledek: - Pouze JEDNA půjčka je vytvořena - Druhá dostane chybu: “Book is currently not available” - Transakční integrita je zachována

Test 8: Vrácení již vrácené knihy

Příprava: - Půjčte knihu a vraťte ji

Test: 1. V databázi zjistěte ID vrácené půjčky:

```
SELECT id FROM loan WHERE returned = 1 ORDER BY id DESC;
```

2. Ručně pokuste se zavolat return endpoint (např. pomocí curl nebo browser dev tools)

Očekávaný výsledek: - Chyba: “Error: Book already returned” - Stav knihy se nezmění

Test 9: Chybějící datové typy

Ověření všech datových typů v DB:

```
-- FLOAT
```

```
SELECT price FROM book;
```

```
-- BIT
```

```
SELECT available FROM book;
```

```
SELECT active FROM library_user;
```

```
SELECT returned FROM loan;
```

```
-- ENUM (CHECK constraint)
```

```
SELECT category_type FROM category WHERE category_type NOT IN ('fiction', 'nonfiction', 'story');
```

```
-- Mělo by vrátit 0 řádků
```

```
-- VARCHAR
```

```
SELECT title FROM book;
```

```
SELECT email FROM library_user;
```

```
-- DATE/DATETIME
```

```
SELECT published_date FROM book;
```

```
SELECT created_at FROM library_user;
```

```
SELECT loan_date FROM loan;
```

Očekávaný výsledek: - Všechny datové typy jsou používány - ENUM
(CHECK constraint) funguje správně

Závěr testování chyb

Shrnutí testů:

Test	Scénář	Výsledek
1	Chyby DB připojení	PASS / FAIL
2	Porušení integrity	PASS / FAIL
3	Neplatné vstupy	PASS / FAIL
4	Chybný JSON import	PASS / FAIL
5	Konfigurace	PASS / FAIL
6	Edge cases	PASS / FAIL
7	Transakční integrita	PASS / FAIL
8	Duplicitní operace	PASS / FAIL
9	Datové typy	PASS / FAIL

Poznámky k chybám:

- Všechny chyby musí být ošetřeny s user-friendly hláškami
- Aplikace se nesmí zhroutit (Python traceback)
- Uživatel musí mít možnost pokračovat po chybě
- Databázová integrita musí být zachována

Tester: _____

Datum: _____

Celkový výsledek: PASS / FAIL

Poznámky: