

DATABÁZE BANK A JEJICH KLIENTŮ

(B6B36TS1) Testování software – Semestrální práce

24. května 2018

David Bubeník

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Obsah

[2 Návrh testovací strategie a scénářů 2](#_Toc514689153)

[2.1 Popis aplikace 2](#_Toc514689154)

[2.2 Přehled částí aplikace 3](#_Toc514689155)

[2.2.1 Hlavní okno 3](#_Toc514689156)

[2.2.2 Okno přidání a editace záznamu 3](#_Toc514689157)

[2.2.3 Okno detailu záznamu 4](#_Toc514689158)

[2.3 Prioritizace částí aplikace 4](#_Toc514689159)

[2.3.1 Třídy rizik 4](#_Toc514689160)

[2.3.2 Odhady rizik 5](#_Toc514689161)

[2.4 Test levels 7](#_Toc514689162)

# Návrh testovací strategie a scénářů

## Popis aplikace

Aplikace, kterou jsem programoval jako poslední domácí úkol pro předmět *(B0B36DBS) Databázové systémy*.

Zadání domácího úkolu: [http://www.ksi.mff.cuni.cz/~svoboda/courses/172-B0B36DBS/#assignment-07-JPA](http://www.ksi.mff.cuni.cz/~svoboda/courses/172-B0B36DBS/%23assignment-07-JPA)

Aplikace je napsána v jazyce Java 8. Mezi další použité nástroje a knihovny patří například Maven, JavaFX, FXML, JavaFX Scene Builder 2.0, a další…

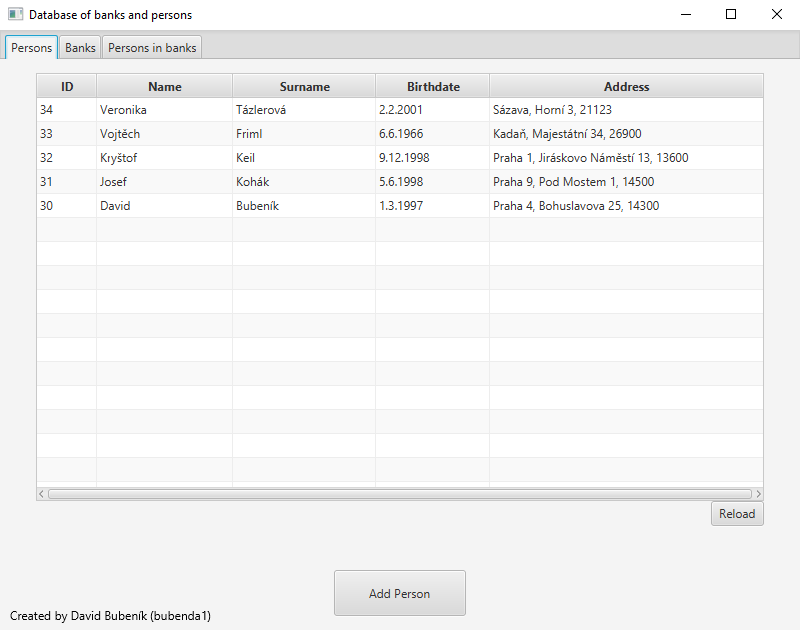
Aplikace je vizuální nástroj pro správu PostgreSQL databáze bank a klientů těchto bank. Díky aplikaci se dají provádět jednoduché úkony s databází (CRUD).

Databáze obsahuje: 2 entitní tabulky – Banks, Persons

1 vztahovou tabulku typu *@ManyToMany* – BanksPersons

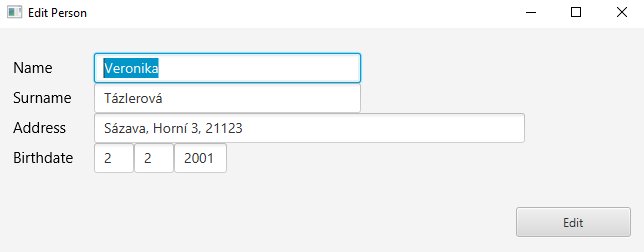
## Přehled částí aplikace

### Hlavní okno

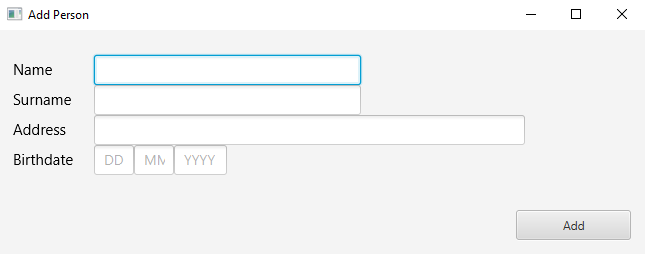


Obrázek Hlavní okno které obsahuje záložky s výpisem všech dat z tabulek

### Okno přidání a editace záznamu

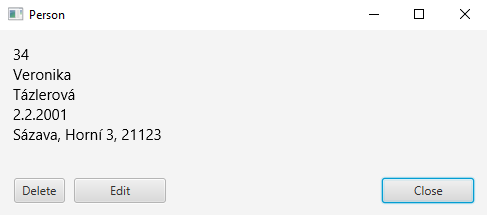


Obrázek Editace záznamu v databáze



Obrázek Přidání nového záznamu do databáze

### Okno detailu záznamu



Obrázek Detail záznamu v databázi s tlačítky pro smazání, nebo editaci záznamu

## Prioritizace částí aplikace

### Třídy rizik

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Pravděpodobnost selhání | | |
| Vysoké (H) | Střední (M) | Malá (L) |
| Možné poškození v případě selhání | Vysoké (H) | **A** | **B** | **B** |
| Střední (M) | **B** | **B** | **C** |
| Malé (L) | **C** | **C** | **C** |

Tabulka Tabulka tříd rizik (Převzata z BDTM)

|  |  |
| --- | --- |
| Třída rizika | Priorita |
| A | **Vysoká (H)** |
| B | **Střední (M)** |
| C | **Nízká (L)** |

Tabulka Vysvětlení značení tříd rizik

### Odhady rizik

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proces | Podproces | Požadavek | Možné poškození | Vysvětlení možného poškození | Část systému | Pravděpodobnost selhání | Vysvětlení pravděpodobnosti selhání | Třída rizika |
| Práce na projektu | Vývoj | Založení | **H** | Pokud se projekt nezaloží nebude nikdy funkční | Projektová struktura | **L** | Pokud se již rozhodneme projekt začít je skoro malá pravděpodobnost, že nebude založen | **B** |
| Práce na projektu | Vývoj | Uložení | **H** | Neuložení může znamenat spoustu ztraceného času | Projektová struktura | **H** | Často se stává, že selže HW, SW, nebo programátor a projekt nebude uložen | **A** |
| Práce s projektem | - | Spuštění | **H** | Projekt se nespustí, a proto nebude možné v něm cokoliv dělat | Klientská aplikace | **M** | I malé chybě v kódu se nemusí projekt spustit | **B** |
| Práce s projektem | JDBC | Připojení k databázovému serveru | **H** | Nepřipojení k databázovému serveru znamená, že nebude možné v aplikaci cokoliv dělat | Databázový ovladač | **L** | Server běží na školních serverech. Malá pravděpodobnost výpadku | **B** |
| Práce s projektem | JPA | Správné databázové schéma | **H** | Špatné schéma zapříčiní nemožnost v aplikaci cokoliv dělat | Databázový ovladač | **M** | Není implementován systém na automatické vytvoření schématu a proto je pravděpodobnost větší | **B** |
| Funkčnost GUI | - | Tlačítka funkční | **M** | Nefunkční tlačítka | FXML schéma | **L** | Nestává se, že by tlačítka nebyla správně propojena s kódem | **C** |
| Funkčnost GUI | - | Tabulky funkční | **M** | Nefunkční tabulky | FXML schéma | **L** | Nestává se, že by tabulky nebyli správně propojeny s kódem | **C** |
| Přidávání dat | XSS | Nestandartní znak/soubor znaků | **H** | Nestandartní znak, nebo soubor znaků může poškodit databázi | Validátor | **H** | Velice často se zapomene na nějakou kombinaci, co by mohla poškodit databázi | **A** |
| Přidávání dat | Validace | Validní vstupní data | **M** | Data, která nejsou očekávána mohou zapříčinit negativní výsledek | Validátor | **M** | Jednoduchá validace, malá pravděpodobnost chyby | **B** |
| Přidávání dat | Opakování | Zamezit opakování klíčů | **H** | Duplikáty mohou vést k nestandartnímu chování | Validátor | **L** | Dobře navržené databázové schéma samo hlídá duplikáty | **B** |
| Přidávání dat | Uložení | Správně vložit do databáze | **M** | Data se nemusí vložit, ale nepoškodí to databázi | CRUD Kontrolér | **M** | Malá šance, že se data neuloží ale je to základní funkce | **B** |
| Editace dat | XSS | Nestandartní znak/soubor znaků | **H** | Nestandartní znak, nebo soubor znaků může poškodit databázi | Validátor | **H** | Velice často se zapomene na nějakou kombinaci, co by mohla poškodit databázi | **A** |
| Editace dat | Validace | Validní vstupní data | **M** | Data která nejsou očekávána mohou zapříčinit negativní výsledek | Validátor | **M** | Jednoduchá validace, malá pravděpodobnost chyby | **B** |
| Editace dat | Opakování | Zamezit opakování klíčů | **H** | Duplikáty mohou vést k nestandartnímu chování | Validátor | **L** | Dobře navržené databázové schéma samo hlídá duplikáty | **B** |
| Editace dat | Příjem dat | Správné předvyplnění | **L** | Špatné předvyplnění může vést ke zmatení klienta, ale nepoškodí databázi | CRUD Kontrolér | **M** | Z důvodu malé složitosti tohoto úkonu je malá pravděpodobnost chyby | **C** |
| Editace dat | Uložení | Správně vložit do databáze | **M** | Data se nemusí vložit, ale nepoškodí to databázi | CRUD Kontrolér | **M** | Malá šance, že se data neuloží, ale je to základní funkce | **B** |
| Smazání dat | - | Správně smazat položku | **M** | Data se nemusí smazat, ale nepoškodí to databázi | CRUD Kontrolér | **M** | Malá šance, že se data neuloží, ale je to základní funkce | **B** |
| Čtení dat | - | Správně přečíst data z databáze | **M** | Data se nemusí přečíst, ale nepoškodí nic | CRUD Kontrolér | **M** | Malá šance, že se data nepřečtou, ale je to základní funkce |  |

Tabulka Odhady rizik

## Test levels

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Charakteristika kvality | Třída rizika | Test levels | | | | | |
| Část systému / funkce | Revize | Vývojářské testy | Systémové testy | UAT | Test v produkci | |
| **Bezchybná funkcionalita** | | | | | | |
| Vytvoření a ukládání aplikace | A | ANO |  |  | Střední |  | |
| Spuštění aplikace | A |  |  |  | Střední | ANO | |
| Bezchybnost funkce GUI | B |  |  | Střední | Střední |  | |
| Přidávání dat | B |  | Vysoká | Vysoká | Nízká |  | |
| Editace dat | B |  | Vysoká | Vysoká | Nízká |  | |
| Mazání dat | B |  | Střední | Střední | Nízká |  | |
| Čtení dat | B |  | Střední | Střední | Nízká |  | |
| **Výkonnost** | | | | | | |
| Čtení dat | C |  | Nízká | Nízká | Nízká | ANO | |
| **Bezpečnost** | | | | | | |
| Zápisy do databáze | A | ANO | Nízká | Vysoká | Nízká |  | |

Tabulka Test levels

## Testovací scénáře

|  |  |
| --- | --- |
| Část systému / funkce | Testy |
| Bezchybnost funkce GUI | UAT |
| Přidávání dat | Unit, Coverage techniky |
| Editace dat | Unit, Coverage techniky |
| Mazání dat | Unit |
| Čtení dat | Unit |
| Průchod aplikací | UI Testy, Test Cases |
| Výkonost – Čtení dat | UAT |

Tabulka Testovací scénáře

## Testovací situace

### Formulář přidání osoby do databáze

Česky:

POKUD (**DELKA\_JMENA** < 1 NEBO **DELKA\_JMENA** > 255)

NEBO (**DELKA\_PRIJMENI** < 1 NEBO **DELKA\_PRIJMENI** > 255)

NEBO (**DELKA\_ADRESY** < 1 NEBO **DELKA\_ADRESY** > 255)

NEBO (**DEN\_NAROZENI** < 1 NEBO **DEN\_NAROZENI** > 31)

NEBO (**MESIC\_NAROZENI** < 1 NEBO **MESIC\_NAROZENI** > 12)

NEBO (**ROK\_NAROZENI** < 1900 NEBO **ROK\_NAROZENI** > 2018)

TAK **NEUKLADAT\_DO\_DATABAZE**

Formálně:

IF(**A** OR **B**)

OR (**C** OR **D**)

OR (**E** OR **F**)

OR (**G** OR **H**)

OR (**I** OR **J**)

OR (**K** OR **L**)

THEN R

**Velké množství proměnných. Pro tento formulář jsem nedělal testy.**

### Formulář přidání banky do databáze

Česky:

POKUD (**DELKA\_JMENA** < 1 NEBO **DELKA\_JMENA** > 255)

NEBO (**DELKA\_KODU** < 1 NEBO **DELKA\_KODU** > 5)

NEBO (**DELKA\_ADRESY** < 1 NEBO **DELKA\_ADRESY** > 255)

TAK **NEUKLADAT\_DO\_DATABAZE**

Formálně:

IF(**A** OR **B**)

OR (**C** OR **D**)

OR (**E** OR **F**)

THEN R

#### MCC

**64 Možných kombinací. Použít MCC je časově náročné pokud se testy nezautomatizují (Unit testy?).**

Udělám proto tabulku pouze pro JMÉNO a KÓD:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Možné kombinace** | | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **A** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **B** | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **C** | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| **D** | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| **R** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Tabulka MCC možné kombinace vyplnění jména a kódu

#### MC/DC

Udělám proto tabulku pouze pro JMÉNO a KÓD:

|  |  |  |
| --- | --- | --- |
| R = (A OR B ) OR ( C OR D ) | 1 | 0 |
| A | 1000 | ~~0000~~ |
| B | 0100 | ~~0000~~ |
| C | 0010  1010  0110 | ~~0000~~ |
| D | 1001  0101  0011 | 0000 |

Tabulka MC/DC

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Kombinace vstupů** | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **16** |
| **A** | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| **B** | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **C** | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| **D** | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| **R** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Tabulka MC/DC Kombinace výstupů

Omezili jsme počet kombinací, ale tím jsme i zvýšili riziko že neotestujeme kombinaci, která může vyjít špatně. Protože je toto ale relativně triviální, je jasné, že tento počet testů (i méně) bude stačit.

#### DC

Testujeme takovým způsobem, abychom získali alespoň jednou všechny výsledky logického výrazu.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case | A | B | C | D | E | F |
| 1. | 1 | 1 | 1 | 1 | 1 | 1 |
| 2. | 0 | 0 | 0 | 0 | 0 | 0 |

Tabulka Decision Coverage

#### CC

Testujeme takovým způsobem, abychom získali alespoň jednou všechny kombinace každé proměnné.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test case | A | B | C | D | E | F | R |
| 1. | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 2. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3. | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Tabulka Condition Coverage

#### C/DC

Kombinace CC a DC.

Stejné jako **Tabulka 10**.

#### Pairwise

Použijeme metodu Pairwise v programu Allpairs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Délka jména menší než 1** | **Délka jména větší než 255** | **Délka kódu menší než 1** | **Délka kódu větší než 5** | **Délka adresy menší než 1** | **Délka adresy větší než 255** |
| Ano | Ano | Ano | Ano | Ano | Ano |
| Ano | Ne | Ne | Ne | Ne | Ne |

Tabulka Pairwise před

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TEST CASES | | | | | | | |
| case | Délka jména menší než 1 | Délka jména větší než 255 | Délka kódu menší než 1 | Délka kódu větší než 5 | Délka adresy menší než 1 | Délka adresy větší než 255 | pairings |
| 1 | Ano | Ano | Ano | Ano | Ano | Ano | 15 |
| 2 | Ano | Ne | Ne | Ne | Ne | Ne | 15 |
| 3 | Ne | Ano | Ne | Ano | Ne | Ano | 11 |
| 4 | Ne | Ne | Ano | Ne | Ano | Ne | 11 |
| 5 | ~Ano | Ano | Ano | Ne | Ne | Ano | 3 |
| 6 | ~Ano | Ne | Ne | Ano | Ano | Ne | 3 |
| 7 | ~Ne | Ano | ~Ne | ~Ne | ~Ano | Ne | 1 |
| 8 | ~Ne | Ne | ~Ano | ~Ano | ~Ne | Ano | 1 |

Tabulka Pairwise po

Dostali jsme 8 testů. Ty by měli stačit. Podle potřeby by šlo ještě snížit na 4 testy.

#### Ruční výběr

Ručně vybereme testovací scénář. Velice nedostačující.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | R |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Tabulka Ruční výběr

### Ostatní formuláře přidání a editace

Pro všechny ostatní formuláře stejný styl jako u předchozího. Nejúčinnější se zdá Pairwise nebo MC/DC technika.

## Analýza mezních podmínek

### Funkčnost databáze

Databázový server musí být spuštěný a přístupný.

### Datum

#### Den

Den musí být celé číslo mezi 1 a 31.

*Pozn.: Program nekontroluje den vzhledem k měsíci přímo, ale pokud uživatel zadá například 31. února, tak program vyhodí výjimku, která se vypíše do chybového pole ve formuláři. Databáze se nepoškodí*.

#### Měsíc

Měsíc musí být celé číslo mezi 1 a 12.

#### Rok

Rok musí být celé číslo mezi 1900 a 2018.

*Pozn.: V databázi musí (a je) být použit datový typ Date, protože datové typy počítající od 1.1.1970 by nedovolili data starší.*

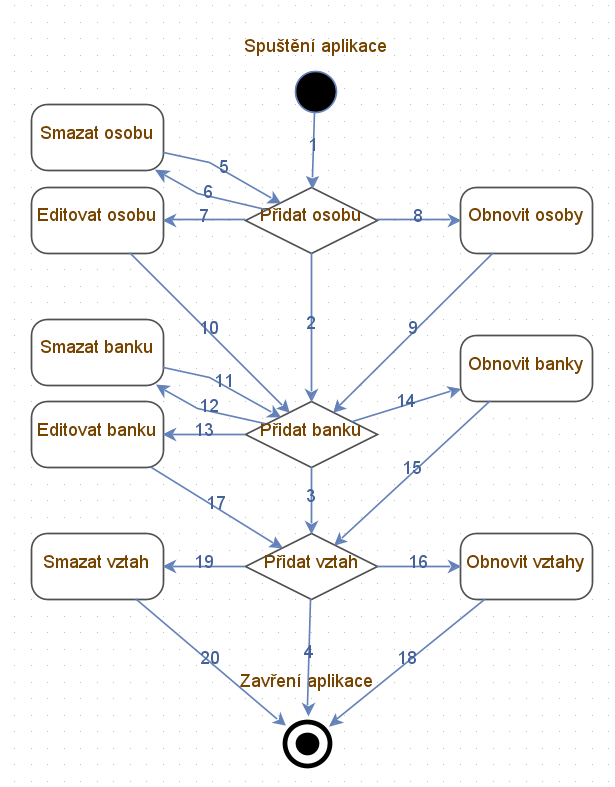
### Přidání vztahu mezi bankou a osobou

Pole osoba musí být validní a existující (V databázi) objekt typu „JavaPersons“.

Pole banka musí být validní a existující (V databázi) objekt typu „JavaBanks“.

## Procesní test

### Graf



Obrázek Procesní test

|  |  |  |
| --- | --- | --- |
| Větvící bod | Vstupní akce | Výstupní akce |
| Spuštění aplikace | - | 1 |
| Přidat osobu | 1 | 2,7,8 |
| Smazat osobu | 6 | 5 |
| Editovat osobu | 7 | 10 |
| Obnovit osoby | 8 | 9 |
| Přidat banku | 10,2,9 | 13,3,14 |
| Smazat banku | 11 | 11 |
| Editovat banku | 13 | 17 |
| Obnovit banky | 14 | 15 |
| Přidat vztah | 17,3,15 | 19,16 |
| Smazat vztah | 19 | 20 |
| Obnovit vztahy | 16 | 18 |
| Zavření aplikace | 20,4,18 | - |

Tabulka Procesní test – Vstupní a výstupní akce pro větvící body

### Hloubka pokrytí 1

|  |  |
| --- | --- |
| Test | Sekvence akcí |
| 1. | 1-2-3-4 |
| 2. | 1-6-5-7-10-12-11-13-17-19-20 |
| 3. | 1-8-9-14-15-16-18 |

Tabulka Procesní test – Hloubka pokrytí 1

### Hloubka pokrytí 2

|  |  |
| --- | --- |
| Test | Sekvence akcí |
| 1. | 1 - 2 – 3 - 4 |
| 2. | 1 - 8 - 9 - 3 - 16 - 18 |
| 3. | 1 - 7 - 10 - 3 - 19 - 20 |
| 4. | 1 - 6 - 5 - 2 - 14 - 15 - 4 |
| 5. | 1 - 6 - 5 - 6 - 5 - 8 - 9 - 14 - 15 - 16 - 18 |
| 6. | 1 - 6 - 5 - 7 - 10 - 14 - 15 - 19 - 20 |
| 7. | 1 - 8 - 9 - 13 - 17 - 4 |
| 8. | 1 - 8 - 9 - 12 - 11 - 3 - 4 |
| 9. | 1 - 2 - 13 - 17 - 16 - 18 |
| 10. | 1 - 2 - 12 - 11 - 14 - 15 - 4 |
| 11. | 1 - 7 - 10 - 13 - 17 - 19 - 20 |
| 12. | 1 - 7 - 10 - 12 - 11 - 13 - 17 - 4 |
| 13. | 1 - 8 - 9 - 12 - 11 - 12 - 11 - 3 - 4 |

Tabulka Procesní test – Hloubka pokrytí 2

### Test datové konzistence

|  |  |  |  |
| --- | --- | --- | --- |
|  | Osoby | Banky | VztahOsobaBanka |
| 1. Přidat osobu | C |  |  |
| 1. Smazat osobu | D |  |  |
| 1. Editovat osobu | R, U |  |  |
| 1. Obnovit osoby | R |  |  |
| 1. Přidat banku |  | C |  |
| 1. Smazat banku |  | D |  |
| 1. Editovat banku |  | R, U |  |
| 1. Obnovit banky |  | R |  |
| 1. Přidat vztah | R | R | C |
| 1. Smazat vztah |  |  | D |
| 1. Obnovit vztahy |  |  | R |

Tabulka Test datové konzistence - CRUD matice

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 4 | 9 | 3 | 4 | 9 | 2 | 4 | 9 |
| C | R | R | RU | R | R | D | R | R |

Tabulka Test datové konzistence – Osoby

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 8 | 9 | 7 | 8 | 9 | 6 | 8 | 9 |
| C | R | R | RU | R | R | D | R | R |

Tabulka Test datové konzistence – Banky

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | 11 | 10 | 11 |
| C | R | D | R |

Tabulka Test datové konzistence – VztahOsobaBanka

## Detailní testovací scénáře

|  |  |  |
| --- | --- | --- |
| ID Testu | 1 | |
| Název testu | Spuštění aplikace a průchod | |
| Shrnutí testu | Spuštění aplikace, načtení dat z databáze a bezchybný průchod oknem, pozitivní průchod | |
| Vstupní podmínky | * Existující aplikace * Aplikace je spustitelná na daném HW/SW * Všechny tři tabulky v databázi aplikace mají alespoň jeden řádek dat | |
| Testovací data | - | |
| Očekávaný výsledek | Aplikace se správně spustí a připojí k databázi (Zobrazí se seznam dat z databáze), lze překlikávat mezi záložkami | |
| Kroky testu | |  |
| 1. | | Spustit spustitelný soubor aplikace (aplikace.jar) |
| Očekávaný výsledek | | Zobrazí se okno aplikace, ve kterém bude vidět tabulka s daty z databáze |
| 2. | | Kliknout na záložku ‚Banks‘ |
| Očekávaný výsledek | | Zobrazí se tabulka s bankami |
| 3. | | Kliknout na záložku ‚Persons in banks‘ |
| Očekávaný výsledek | | Zobrazí se tabulka se vztahy osob a banek |

Tabulka Detailní testovací scénář 1

|  |  |  |
| --- | --- | --- |
| ID Testu | 2 | |
| Název testu | Přidání nové osoby | |
| Shrnutí testu | Přidání nové osoby do databáze, pozitivní průchod | |
| Vstupní podmínky | * Spuštěná aplikace | |
| Testovací data | * Jméno: Karel * Příjmení: Novák * Adresa: Karlovy Vary, Kozí 35, 89600 * Datum narození: 1.9.1995 | |
| Očekávaný výsledek | Okno pro přidání osoby se zavře a náš záznam se zobrazí v tabulce | |
| Kroky testu | |  |
| 1. | | Vybrat záložku „Persons“, Kliknout na „Add Person“ |
| Očekávaný výsledek | | Zobrazí se okno s formulářem |
| 2. | | Vyplnit testovací data a kliknout na tlačítko „Add“ |
| Očekávaný výsledek | | Okno se zavře a naše osoba se zobrazí v tabulce |

Tabulka Detailní testovací scénář 2

|  |  |  |
| --- | --- | --- |
| ID Testu | 3 | |
| Název testu | Editace banky | |
| Shrnutí testu | Editace záznamu banky, změna adresy, pozitivní průchod | |
| Vstupní podmínky | * Spuštěná aplikace * Funkční zobrazení detailu záznamu * Minimálně jedna banka v tabulce bank | |
| Testovací data | * Adresa: Jakákoliv odlišná od stávající | |
| Očekávaný výsledek | Okno editace se zavře a záznam v tabulce se aktualizuje | |
| Kroky testu | |  |
| 1. | | Vybrat záložku „Banks“ a kliknout na záznam banky v tabulce |
| Očekávaný výsledek | | Zobrazí se okno s detailem banky |
| 2. | | Kliknout na tlačítko „Edit“ |
| Očekávaný výsledek | | Otevře se nové okno s formulářem. Formulář má předvyplněná data banky |
| 3. | | Upravit adresu (Změnit na jakoukoliv odlišnou od již vyplněné) |
| Očekávaný výsledek | | Okno se zavře a banka v tabulce bude mít v poli adresa naší vyplněnou adresu. |

Tabulka Detailní testovací scénář 3

|  |  |  |
| --- | --- | --- |
| ID Testu | 4 | |
| Název testu | Obnovení dat tabulky osob | |
| Shrnutí testu | Načtení aktuálních dat tabulky osob z databáze, pozitivní průchod | |
| Vstupní podmínky | * Spuštěná aplikace * Možnost vložit data do tabulky databáze externě (Přes administraci databáze) | |
| Testovací data | * Jméno: Karel * Příjmení: Novák * Adresa: Karlovy Vary, Kozí 35, 89600 * Datum narození: 1.9.1995 | |
| Očekávaný výsledek | Náš externě vložený řádek se zobrazí v tabulce aplikace | |
| Kroky testu | |  |
| 1. | | Vložit nový řádek s testovacími daty do tabulky (Přímo do databáze, externě, ne přes testovanou aplikaci) |
| Očekávaný výsledek | | Řádek se nám zobrazí v databázi |
| 2. | | V aplikaci vybrat záložku „Persons“ a kliknout na tlačítko „Reload“ |
| Očekávaný výsledek | | Náš externě vložený řádek se zobrazí v tabulce testované aplikace |

Tabulka Detailní testovací scénář 4

|  |  |  |
| --- | --- | --- |
| ID Testu | 5 | |
| Název testu | Smazání osoby | |
| Shrnutí testu | Smazání osoby z databáze, pozitivní průchod | |
| Vstupní podmínky | * Spuštěná aplikace * Funkční zobrazení detailu záznamu * Minimálně jedna osoba v tabulce osob | |
| Testovací data | - | |
| Očekávaný výsledek | Osoba zmizí z tabulky | |
| Kroky testu | |  |
| 1. | | Vybrat záložku „Persons“ a kliknout na záznam osoby v tabulce |
| Očekávaný výsledek | | Zobrazí se okno s detailem osoby |
| 2. | | Kliknout na tlačítko „Delete“ |
| Očekávaný výsledek | | Okno se zavře a osoba zmizí z tabulky |
| 3. | | Kliknout na tlačítko ‚Reload‘ |
| Očekávaný výsledek | | Smazaná osoba se nezobrazí v tabulce ani nyní |

# Unit testy

## Základní testy pro metody tříd

V kódu aplikace. Něco přes 10 obyčejných testů a jeden parametrizovaný.

## Procesní testy

V kódu aplikace. Několik procesních testů kontrolujících zápisy a čtení z/do databáze.

# Nalezené chyby

V aplikaci nebyly nalezeny žádné chyby, ale bylo třeba lehce upravit některé třídy, aby bylo testování vůbec proveditelné v požadovaném měřítku. Databáze a FXML také trochu komplikovali testování (Testování GUI v Javě jsme neprobírali)