České vysoké učení technické v Praze. Fakulta elektrotechnická, SIT.

Testování messengerů Semestrální práce Předmět: Software Testing

Autor: Arina Momot

Datum: 25.05.2020

Obsah

[1. Návrh testovací strategie 2](#_Toc41964933)

[1.1 Popis funkcionality aplikace 2](#_Toc41964934)

[1.2 Přehled částí aplikace 2](#_Toc41964935)

[1.3. Prioritizace částí aplikace 3](#_Toc41964936)

[1.4 Test Levels 4](#_Toc41964937)

[Podrobnější analýza 5](#_Toc41964938)

[1.1 Procesy a požadavky 5](#_Toc41964939)

[1.2 Overall test goals 5](#_Toc41964940)

[1.4 Sjednocené test goals. 6](#_Toc41964941)

[1.6 Možnosti selhání. 6](#_Toc41964942)

[1.7 Určení rizik. 7](#_Toc41964943)

[1.8 Test levels. 9](#_Toc41964944)

[2. Testovací scénáře. 10](#_Toc41964945)

[2.1 Testovací situace pro vstupy (testy vstupů - třídy ekvivalence). 10](#_Toc41964946)

[**Formulář registrace.** 10](#_Toc41964947)

[2.1.1 Vstup „First name“ 11](#_Toc41964948)

[Typy tříd ekvivalence – podle validity dat: 11](#_Toc41964949)

[2.1.2 Vstup „Last name“ 11](#_Toc41964950)

[Typy tříd ekvivalence – podle validity dat: 12](#_Toc41964951)

[2.1.3 Vstup „E-mail “ 12](#_Toc41964952)

[Typy tříd ekvivalence – podle validity dat: 13](#_Toc41964953)

[2.1.4 Vstup „Password “ 13](#_Toc41964954)

[Typy tříd ekvivalence – podle validity dat: 13](#_Toc41964955)

[1. Mezní podmínky 14](#_Toc41964956)

[2.1.5 Vstup „Repeat password “ 14](#_Toc41964957)

[Typy tříd ekvivalence – podle validity dat: 15](#_Toc41964958)

[2. Mezní podmínky 16](#_Toc41964959)

[2.1.6 Vstup „Country “ 16](#_Toc41964960)

[Typy tříd ekvivalence – podle validity dat: 16](#_Toc41964961)

[2.1.7 Vstup „Date of birth “ 17](#_Toc41964962)

[Typy tříd ekvivalence – podle validity dat: 17](#_Toc41964963)

[3. Mezní podmínky 18](#_Toc41964964)

[2.1.8 Vstup „Gender “ 18](#_Toc41964965)

[**Formulář vytvoření skupiny.** 19](#_Toc41964966)

[2.1.1 Vstup „Title “ 19](#_Toc41964967)

[2.1.2 Vstup „Description“ 19](#_Toc41964968)

[1. Mezní podmínky pro Popis 20](#_Toc41964969)

[2.1.3 Vstup „Type “ 20](#_Toc41964970)

[Typy tříd ekvivalence – podle validity dat: 21](#_Toc41964971)

[2.2 Testovací situace pro vstupy (testy vstupů - kombinace dat). 21](#_Toc41964972)

[**Pairwise testing.** 21](#_Toc41964973)

[2.3 Detailní testovací scénáře. 22](#_Toc41964974)

[1) Pozitivní průchod 22](#_Toc41964975)

[2) Negativní průchod: 24](#_Toc41964976)

[2.4 Testy průchodů (path based). 27](#_Toc41964977)

# **Návrh testovací strategie**

V této části dokumentu je popsán testovaný systém, očekávání jeho funkčnosti a jsou prioritizovány kritičtější komponenty k otestování.

## 1.1 Popis funkcionality aplikace

Tato chatová aplikace umožňuje uživatelům se zaregistrovat, vytvářet skupiny pro komunikaci pro dve a více osob, vzájemně komunikovat odesíláním zpráv a nálepek a také prohlížet informace o uživatelích, které on poskytl.

Použité technologie:

* Aplikace se provádí pomocí **GUI**. Lidé jednoduše používají myš k přímé manipulaci s rozhraním.
* Projekt vytvořen pomocí **Maven**.
* Data o uživatelích, skupinách a zprávách uložena do **databáze**.
* Chatová aplikace je implementována pomocí architektury **klient-server**. Tento typ architektury má jeden nebo více klientských počítačů připojených k centrálnímu serveru.
* Vlastní protokol založený na **JSON**.
* Tato chatová aplikace je vícevláknová:
* Umožňují zpracovat požadavky klienta paralelně nezávisle na sobě.
* Server poslouchá požadavky na připojení od klientů v síti. Klienti vědí, jak se připojit k serveru pomocí čísla portu.

## 1.2 Přehled částí aplikace

|  |  |
| --- | --- |
| Část aplikace | Funkce |
| Server aplikace | Jádro programu, poslouchá klienty na portu. |
| Client aplikace | Odpovědný za připojení k serveru, komunikaci s ním, zpracování dat a poslání odpovědi v Connection. |
| Connection | Posilá request a přijímá odpověď ze serveru, zpracuje data a posílá data do front-end. |
| DB Servises | Přístup k databazí. |
| Requests | Komunikaci serveru s klientem. |
| Model s entitami | Representaci dat a objektu s databazí. |
| Controllers | Manipulace s FXML soubory a rozhraním. |
| Blok s FXML soubory | Zobrazení uživatelského rozhrání od logiky aplikace. |
| View | Upozornění uživatele o chybě nebo úspěchů operaci. |

## 1.3. Prioritizace částí aplikace

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Část aplikace | Požadavek | Možnost poškození | Vysvětlení možného poškození | Pravdě  podobnost selhání | Vysvětlení  pravděpodobnost  selhání | Třída rizika |
| Server | Poslouchá klienty na správném portu. | High | Nemožnost používat program | High | Stálá práce této části aplikace. | A |
| Client | Připojení k serveru, komunikaci s ním a s databázi, zpracování dat z databáze a poslání odpovědi v Connection. | High | Nemožnost používat program | High | Stálá práce této části aplikace. | A |
| Connection | Posílání requestů a přijaty odpovědí ze serveru, zpracování dát a posílání dat do front-endu. | High | Nemožnost používat program | High | Stálá práce této části aplikace. | A |
| DB Servises | Možnost neustalé úpravy obsahu databáze (CREATE, READ, UPDATE, DELETE)  Operace pro User, Group, Membership a Message: CREATE, READ, UPDATE, DELETE . | High | Ztráta dát a hlavní funkce programu | High | Částé použití této části aplikace. | A |
| Requests | Komunikaci serveru s klientem. | High | Nemožnost komunikace Serveru a klientu(uzivatelu) | High | Částé použití této části aplikace. | A |
| Model s entity | Kompalitibilní a validní representace datových objektu (data type, mapování, anotace) | High | Možné chyby a poruchy s přístupem k DB, v důsledku ztrata funkcionality | High | Částé použití této části aplikace. | A |
| Controllers | Validace na straně klienta, odchycení chyb serveru | Medium | Pokusy vložení do databáze nevalidních dat, což může vést k jejích ztrátě | Low | Odstranění chyb databazí snižuje riziko. | C |
| Správný přechod mezi okny aplikace a representace dat z databáze | High | Přerušené pořádí může vést ke ztratě dat, nespolehlivá informace vede k zátěži pro dálší zpracování a manipulaci s datami | Medium | Částé použití této části aplikace. | B |
| FXML soubory | Uživatelsky přívětivé rozhrání  Snadné při použití | Medium | Možná ztráta reputace a spokojenosti uživatelů | Low | Funkcionalita je řešena pomocí stabilního systému. | C |
| View | Možnost informovat klienta o úspěchu/možných důvodech chyby při používání aplikaci prostředníctvím informačních/chybových hlášek | Medium | Nepochopení uživatelem zamítnutí transakce | Medium | Vložení neočekávaných vstupu, které nebyly odstráněny při vytvoření validace na straně klienta, chyba odstraněná databazí (neočekávamé zamitnutí transakce) | B |

## 1.4 Test Levels

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Část systému / funkce | Třída rizika | Vývojářské testy | Systémové testy | UAT | Test v produkci |
| Poslech klienty na portu. | A | Vysoká | Vysoká | Střední | Ano |
| Připojení k serveru, komunikaci s ním a s databáze , zpracování dat z databáze a poslání odpovědi v Connection. | A | Vysoká | Vysoká | Střední | Ano |
| Posílání requestů a přijaty odpovědí ze serveru, zpracování dát a posílání dat do front-endu. | A | Vysoká | Vysoká | Střední | Ano |
| DB Serveses | A | Vysoká | Vysoká | Střední | Ano |
| Requests | A | Vysoká | Vysoká | Střední | Ano |
| Controller - validace dat | C |  | Střední | Střední |  |
| View (Informační/chybové hlášky) | B |  | Střední |  |  |
| Controller - přechod mezi okny aplikace | B | Nízká | Nízká | Nízká |  |
| Uživatelsky přívětivé rozhrání | C |  | Střední | Střední | Ano |
| Model s entity (representace datových objektu) | A | Výsoká | Vysoká |  |  |

## Podrobnější analýza

### 1.1 Procesy a požadavky

Jednotlivé požadavky na funkčnost jsou následující:

|  |  |  |
| --- | --- | --- |
| Proces | Požadavek | Část systému |
| Registrace | Registrace pomocí křestního jména, příjmení, mailu, hesla, země, data narození a pohlaví. | Registrace |
| Přihlášení | Přihlášení pomocí kombinace mailu a hesla | Přihlášení |
| Komunikace | Odesílání a přijímání zpráv ve skupině. | Uživatelský modul |
|  |  |  |
| Zobrazení | Zobrazení seznamu skupin uživatele. | Uživatelský modul |
| Zobrazení | Zobrazení informace o skupině. | Uživatelský modul |
| Zobrazení | Zobrazení nastavení profilu uživatele. | Uživatelský modul |
| Zobrazení | Zobrazení zpráv ve vybrané skupině, ve které se uživatel skládá. | Uživatelský modul |
| Zobrazení | Zobrazení seznamu uživatelů při vyhledávání pro přidání do skupiny. | Uživatelský modul |
|  |  |  |
| Spojení s databází | Zpracovaná data musí být ošetřena vůči útokům do databáze | Zabezpečení databáze |
| Registrace | Heslo musí obsahovat 8 a více znaků. | Registrace |
| Registrace | Registrace možna pouze s unikátním mailem. | Registrace |
| Registrace | Při registraci musí být všechna pole vyplněna. | Registrace |
|  |  |  |
| Spojení s databází | Zobrazení stavu aplikace při stahování dat (progress bar) a zobrazení okna s chybou. | Uživatelský modul |
| Spojení s databází | Zobrazení stavu aplikace pomocí zobrazení okna s chybou. | Uživatelský modul |
|  |  |  |
| Správa databáze | Vytvoření nové skupiny pomocí kombinace názvu, popisu a typu skupiny. | Uživatelský modul |
| Správa databáze | Úprava informací o skupině (název a popis). | Uživatelský modul |
| Správa databáze | Přidání nových uživatelů do skupiny pomocí vyhledávání uživatelů podle jména nebo příjmení v Databáze. | Uživatelský modul |
| Správa databáze | Odstranění uživatelů ze skupiny. | Uživatelský modul |
| Správa databáze | Změna práv konkrétního uživatele ve skupině. | Uživatelský modul |
| Správa databáze | Smazání skupiny (pouze tvůrce skupiny). | Uživatelský modul |
| Správa databáze | Úprava informací o uživateli(o sobě). | Uživatelský modul |
| Správa databáze | Smazání uživatele (svého profilu). | Uživatelský modul |

### 1.2 Overall test goals

Celkově se od programu očekává stabilní fungování, bezpečnost dat a uživatelská optimalizace.

|  |  |
| --- | --- |
| ID | Overall test goals |
| 1 | Stabilní fungování systému. |
| 2 | Systém bude uživatelsky optimalizován. |
| 3 | Bude zaručena bezpečnost databáze. |

### 1.4 Sjednocené test goals.

|  |  |
| --- | --- |
| ID | Sjednocené test goals |
| 1 | Stabilní fungování systému. |
| 1.1 | Systém bude umožňovat registraci komukoliv kdo má mail. |
| 1.2 | Systém umožní příhlášení registrovaných uživatelů. |
| 1.3 | Systém umožní uživateli psát a přijímat zprávy ve skupině. |
| 1.4 | Systém umožní zobrazení seznamu skupin uživatele. |
| 2 | Systém bude uživatelsky optimalizován. |
| 2.1 | Při nahrávání dat z databáze bude systém zobrazovat stav procesu (status bar). |
| 2.2 | Při chybách se systém zobrazí okno s touto chybou. |
| 3 | Bude zaručena bezpečnost databáze. |
| 3.1 | Systém bude zabezpečen vůči SQL injection. |
| 3.2 | Heslo při registraci musí být 8 nebo více znaků. |
| 3.3 | Systém neumožní návštěvníkovi se zaregistrovat s mailem, který již bude v databázi. |
| 4 | Ostatní |
| 4.1 | Systém umožní uživateli přidávat nové skupiny. |
| 4.2 | Systém umožní uživateli přidat nové uživatele do skupiny. |
| 4.3 | Systém umožní uživateli upravovat informace o sobě v databázi. |
| 4.4 | Systém umožní uživateli upravovat informace o skupině v databázi. |
| 4.5 | Systém umožní tvůrce skupiny upravovat práva jiného uživatele v databázi. |
| 4.6 | Systém umožní tvůrce skupiny odstranit členy jeho skupiny i z databáze. |
| 4.7 | Systém umožní tvůrce skupiny smazat skupinu. |
| 4.8 | Systém umožní uživateli smazat ho účet. |

### 1.6 Možnosti selhání.

Analýza možností selhání měla za cíl odhlalit možné dopady při selhání jednotlivých komponent, aby bylo možné určit prioritu při testování.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Quality characteristic: Bezchybná funkcionalita | | |
| Proces | Požadavek | | Dopad | Vysvětlení |
| Registrace | Registrace pomocí křestního jména, příjmení, mailu, hesla, země, data narození a pohlaví. | | High | Nemožnost používat program. |
| Přihlášeni | Přihlášení pomocí kombinace mailu a hesla | | High | Nemožnost používat program. |
| Komunikace | Odesílání a přijímání zpráv ve skupině. | | High | Ztráta hlavní funkce programu. |
| Zobrazení | Zobrazení seznamu skupin uživatele. | | High | Ztráta hlavní funkce programu. |
| Zobrazení | Zobrazení informace o skupině. | | High | Ztráta hlavní funkce programu. |
| Zobrazení | Zobrazení nastavení profilu uživatele. | | High | Ztráta hlavní funkce programu. |
| Zobrazení | Zobrazení zpráv ve vybrané skupině, ve které se uživatel skládá. | | High | Ztráta hlavní funkce programu. |
| Zobrazení | Zobrazení seznamu uživatelů při vyhledávání pro přidání do skupiny. | | High | Ztráta hlavní funkce programu. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Quality characteristic: Optimalizace | | | |
| Proces | Požadavek | | Dopad | Vysvětlení |
| Spojení s databází | Zobrazení stavu aplikace při stahování dat (progress bar) a zobrazení okna s chybou. | | High | Klíčový atribut uživatelské optimalizace. |
| Spojení s databází | Zobrazení stavu aplikace pomocí zobrazení okna s chybou. | | High | Klíčový atribut uživatelské optimalizace. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Quality characteristic: Zabezpečení | | | |
| Proces | Požadavek | | Dopad | Vysvětlení |
| Spojení s databází | Zpracovaná data musí být ošetřena vůči útokům do databáze | | High | Ztráta reputace, která způsobí odliv uživatelů |
| Registrace | Heslo musí obsahovat 8 a více znaků. | | Low | Nemá negativní vliv na funkce programu |
| Registrace | Registrace možna pouze s unikátním mailem. | | High | Při selhání dojde k nemožnosti identifikovat uživatele |
| Registrace | Při registraci musí být všechna pole vyplněna. | | Low | Nemá negativní vliv na funkce programu |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Quality characteristic: Ostatní | | | |
| Proces | Požadavek | | Dopad | Vysvětlení |
| Správa databáze | Vytvoření nové skupiny pomocí kombinace názvu, popisu a typu skupiny. | | High | Ztráta hlavní funkce programu. |
| Správa databáze | Úprava informací o skupině (název a popis). | | Low | Sekundární funkce aplikace. |
| Správa databáze | Přidání nových uživatelů do skupiny pomocí vyhledávání uživatelů podle jména nebo příjmení v Databáze. | | High | Ztráta hlavní funkce programu. |
| Správa databáze | Odstranění uživatelů ze skupiny. | | Low | Sekundární funkce aplikace. |
| Správa databáze | Změna práv konkrétního uživatele ve skupině. | | Low | Sekundární funkce aplikace. |
| Správa databáze | Smazání skupiny (pouze tvůrce skupiny). | | Low | Sekundární funkce aplikace. |
| Správa databáze | Úprava informací o uživateli(o sobě). | | Low | Sekundární funkce aplikace. |
| Správa databáze | Smazání uživatele (svého profilu). | | Low | Sekundární funkce aplikace. |

### 1.7 Určení rizik.

Nakonec analýzy byla sestavena celková tabulka rizik a jejich dopadů, jakožto z nich i vyvozena priorita využitá při následném testování systému.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | Quality characteristic: Bezchybná funkcionalita | | |  |
| Část systému | Pravděpodobnost selhání | Proces | Požadavek | | Dopad | Vysvětlení | Třída |
| Registrace | Low | Registrace | Registrace pomocí křestního jména, příjmení, mailu, hesla, země, data narození a pohlaví. | | High | Nemožnost používat program. | B |
| Přihlášeni | Low | Přihlášeni | Přihlášení pomocí kombinace mailu a hesla | | High | Nemožnost používat program. | B |
| Uživatelský modul | Medium | Komunikace | Odesílání a přijímání zpráv ve skupině. | | High | Ztráta hlavní funkce programu. | B |
| Uživatelský modul | Medium | Zobrazení | Zobrazení seznamu skupin uživatele. | | High | Ztráta hlavní funkce programu. | B |
| Uživatelský modul | Medium | Zobrazení | Zobrazení informace o skupině. | | High | Ztráta hlavní funkce programu. | B |
| Uživatelský modul | Medium | Zobrazení | Zobrazení nastavení profilu uživatele. | | High | Ztráta hlavní funkce programu. | B |
| Uživatelský modul | Medium | Zobrazení | Zobrazení zpráv ve vybrané skupině, ve které se uživatel skládá. | | High | Ztráta hlavní funkce programu. | B |
| Uživatelský modul | Medium | Zobrazení | Zobrazení seznamu uživatelů při vyhledávání pro přidání do skupiny. | | High | Ztráta hlavní funkce programu. | B |
| Zabezpečení databáze | Není relevantní |  |  | |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | Quality characteristic: Optimalizace | | |  |
| Část systému | Pravděpodobnost selhání | Proces | Požadavek | | Dopad | Vysvětlení | Třída |
| Registrace | Není relevantní |  |  | |  |  |  |
| Přihlášeni | Není relevantní |  |  | |  |  |  |
| Uživatelský modul | High | Spojení s databází | Zobrazení stavu aplikace při stahování dat (progress bar) a zobrazení okna s chybou. | | High | Klíčový atribut uživatelské optimalizace. | A |
| Uživatelský modul | High | Spojení s databází | Zobrazení stavu aplikace pomocí zobrazení okna s chybou. | | High | Klíčový atribut uživatelské optimalizace. | A |
| Zabezpečení databáze | Není relevantní |  |  | |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | Quality characteristic: Bezpečnost | | |  |
| Část systému | Pravděpodobnost selhání | Proces | Požadavek | | Dopad | Vysvětlení | Třída |
| Registrace | Low | Registrace | Heslo musí obsahovat 8 a více znaků. | | Low | Nemá negativní vliv na funkce program. | C |
| Registrace | Low | Registrace | Registrace možna pouze s unikátním mailem. | | High | Při selhání dojde k nemožnosti identifikovat uživatele. | B |
| Registrace | Low | Registrace | Při registraci musí být všechna pole vyplněna. | | Low | Nemá negativní vliv na funkce program. | C |
| Přihlášeni | Není relevantní |  |  | |  |  |  |
| Uživatelský modul | Není relevantní |  |  | |  |  |  |
| Zabezpečení databáze | Medium | Spojení s databází | Zpracovaná data musí být ošetřena vůči útokům do databáze. | | High | Ztráta reputace, která způsobí odliv uživatelů | B |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | Quality characteristic: Ostatni | | |  |
| Část systému | Pravděpodobnost selhání | Proces | Požadavek | | Dopad | Vysvětlení | Třída |
| Registrace | Není relevantní |  |  | |  |  |  |
| Přihlášeni | Není relevantní |  |  | |  |  |  |
| Uživatelský modul | High | Správa databáze | Vytvoření nové skupiny pomocí kombinace názvu, popisu a typu skupiny. | | High | Ztráta hlavní funkce programu. | A |
| Uživatelský modul | High | Správa databáze | Úprava informací o skupině (název a popis). | | Low | Sekundární funkce aplikace. | C |
| Uživatelský modul | High | Správa databáze | Přidání nových uživatelů do skupiny pomocí vyhledávání uživatelů podle jména nebo příjmení v Databáze. | | High | Ztráta hlavní funkce programu. | A |
| Uživatelský modul | High | Správa databáze | Odstranění uživatelů ze skupiny. | | Low | Sekundární funkce aplikace. | C |
| Uživatelský modul | High | Správa databáze | Změna práv konkrétního uživatele ve skupině. | | Low | Sekundární funkce aplikace. | C |
| Uživatelský modul | High | Správa databáze | Smazání skupiny (pouze tvůrce skupiny). | | Low | Sekundární funkce aplikace. | C |
| Uživatelský modul | High | Správa databáze | Úprava informací o uživateli(o sobě). | | Low | Sekundární funkce aplikace. | C |
| Uživatelský modul | High | Správa databáze | Smazání uživatele (svého profilu). | | Low | Sekundární funkce aplikace. | C |
| Zabezpečení databáze | Není relevantní |  |  | |  |  |  |

### 1.8 Test levels.

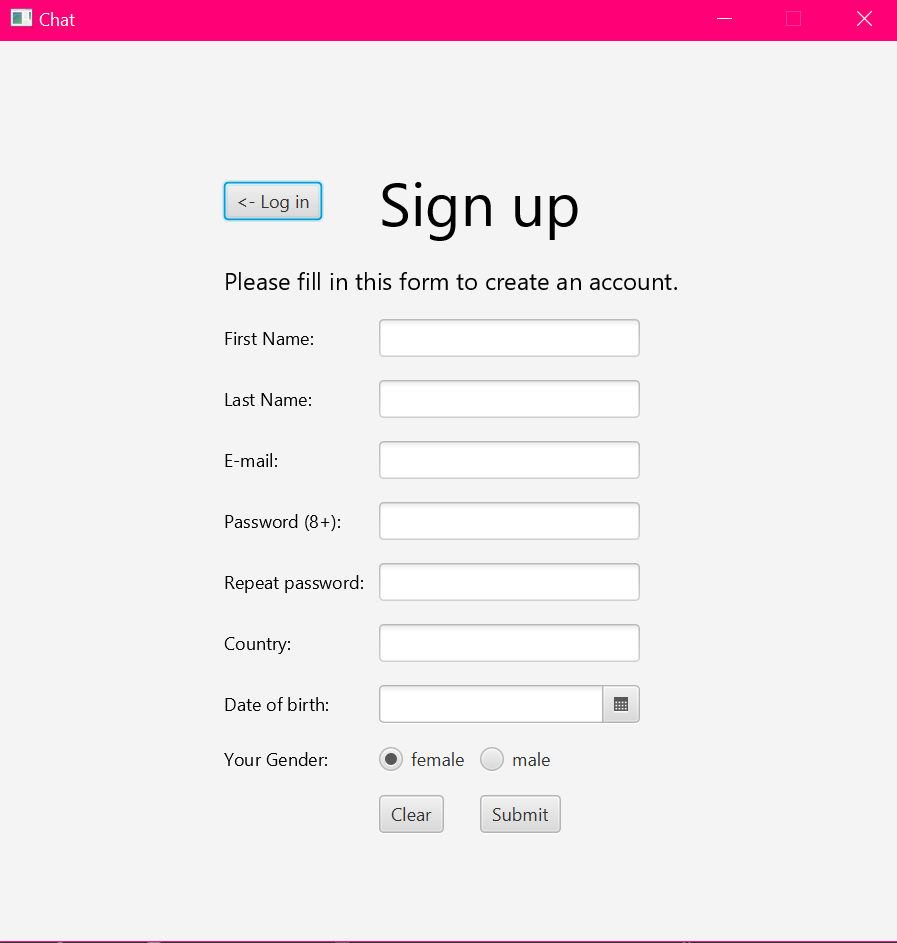
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quality characteristic | | | Test levels | | | | | |
| Část systému | **Funkce** | **Třída rizika** | **Revize** | **Vývojářské testy** | **Systémové testy** | **UAT** | **Test v produkci** | **Typy testů** |
| Bezchybná funkcionalita | | | | | | | | |
| Registrace | Registrace | B | Ano | Vysoká | Nízká |  | Ano | Procesní testy |
| Přihlášení | Přihlášení | B | Ano | Vysoká | Nízká |  | Ano | Procesní testy |
| Uživatelský modul | Komunikace | B | Ano | Střední | Střední | Vysoká | Ano | Unit testy |
| Uživatelský modul | Zobrazení | B |  |  | Nízká |  |  | Procesní testy |
| Optimalizace | | | | | | | | |
| Uživatelský modul | Progress bar | A | Vysoká | Nízká |  |  |  |  |
| Uživatelský modul | Okno s chybou | A | Vysoká | Nízká |  |  |  |  |
| Bezpečnost | | | | | | | | |
| Registrace | Heslo musí obsahovat 8 a více znaků. | C |  |  | Nízká |  |  | Unit testy |
| Registrace | Nutnost unikátního mailu. | B |  |  | Střední |  | Ano | Unit testy |
| Registrace | Při registraci musí být všechna pole vyplněna. | C |  |  | Nízká |  |  | Unit testy |
| Zabezpečení databáze | Ochrana proti SQL injection | B |  |  | Vysoká |  |  | Unit testy |
| Ostatní | | | | | | | | |
| Uživatelský modul | Create group | A | Ano |  | Vysoká | Vysoká | Ano | Integrační testy |
| Uživatelský modul | Change group info | C |  |  | Nízká |  |  | Integrační testy |
| Uživatelský modul | Add new member | A | Ano |  | Vysoká | Vysoká | Ano | Integrační testy |
| Uživatelský modul | Remove member from group | C |  |  | Nízká |  |  | Integrační testy |
| Uživatelský modul | Change member’s rights | C |  |  | Nízká |  |  | Integrační testy |
| Uživatelský modul | Delete group | C |  |  | Nízká |  |  | Integrační testy |
| Uživatelský modul | Change user’s info | C |  |  | Nízká |  |  | Integrační testy |
| Uživatelský modul | Delete user | C |  |  | Nízká |  |  | Integrační testy |

# **Testovací scénáře.**

V této části dokumentu je analyzováno procesní fungování programu a připraveny vstupy pro testy.

## 2.1 Testovací situace pro vstupy (testy vstupů - třídy ekvivalence).

### **Formulář registrace.**



### 2.1.1 Vstup „First name“

Pokud zadané jméno není prázdné a obsahuje pouze písmena, potom bude vyhodnoceno jako validní.

* IF (zadané jméno není prázdné) AND (zadané jméno obsahuje pouze písmena) THEN je validní.

R = (A AND B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Možné kombinace | | | |
| 1 | 2 | 3 | 4 |
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 1 |

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro jméno |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Zvláštní znaky |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Jméno ve validním formátu,ale takové jméno neexistuje |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující jméno ve validním formátu |

### 2.1.2 Vstup „Last name“

Pokud zadané příjmení není prázdné a obsahuje pouze písmena, potom bude vyhodnoceno jako validní.

* IF (zadané příjmení není prázdné) AND (zadané jméno obsahuje pouze písmena) THEN je validní.

R = (A AND B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Možné kombinace | | | |
| 1 | 2 | 3 | 4 |
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 1 |

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro příjmení |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Zvláštní znaky |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Příjmení ve validním formátu, ale takové jméno neexistuje |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující příjmení ve validním formátu |

### 2.1.3 Vstup „E-mail “

Pokud zadaný mail není prázdný, obsahuje pouze platné znaky i @,. a zároveň není v databázi, potom bude vyhodnoceno jako validní.

* IF (zadaný mail není prázdný) AND (zadaný mail obsahuje pouze platné znaky i @,.) AND (zadaný mail ještě není v databázi) THEN je validní.

R = (A AND B AND C)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Možné kombinace | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro e-mail |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Zvláštní znaky a není tam @ a . * Mail už je v databázi |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Mail ve validním formátu, ale takové mail neexistuje |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující mail ve validním formátu a zároveň ještě není v databázi |

### 2.1.4 Vstup „Password “

Pokud zadané heslo není prázdné, obsahuje pouze platné znaky a je delší než sedm znaků a kratší než 40, potom bude vyhodnoceno jako validní.

* IF (zadané heslo není prázdné) AND (zadané heslo obsahuje pouze platen znaky) AND (je delší než sedm znaků,kratší než 40) THEN je validní.

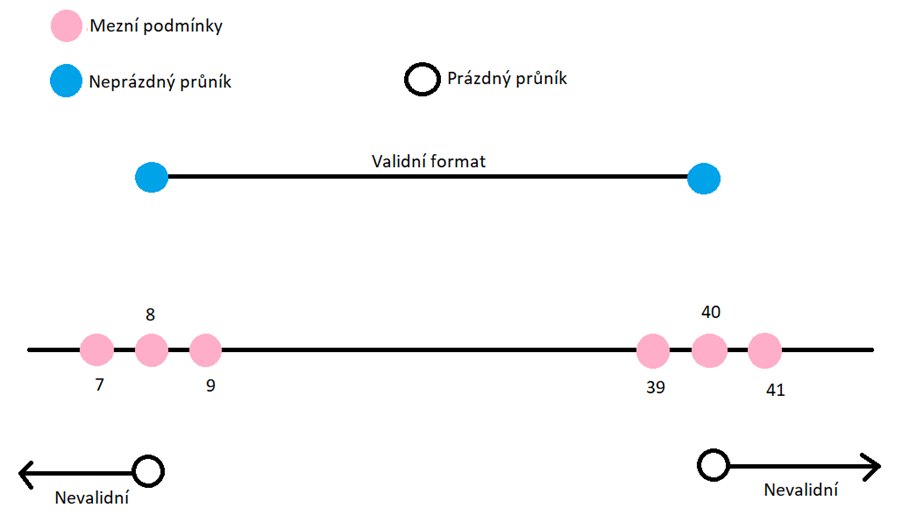
R = (A AND B AND C)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Možné kombinace | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro password |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Zvláštní znaky * Méně než 8 znaků * Více než 40 znaků |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Password ve validním formátu, ale t toto heslo je velmi běžné |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující heslo ve validním formátu a zároveň déle než 7 znaků |

### Mezní podmínky



2.1.5 Vstup „Repeat password “

Pokud zadané ověřovací heslo není prázdné, obsahuje pouze platné znaky, odpovídá zadanému heslu a je delší než sedm znaků a kratší než 40, potom bude vyhodnoceno jako validní.

* IF (zadané heslo není prázdné) AND (zadané heslo obsahuje pouze platen znaky) AND (zadané heslo odpovídá předtím zadanému heslu) AND (je delší než sedm znaků,kratší než 40) THEN je validní.

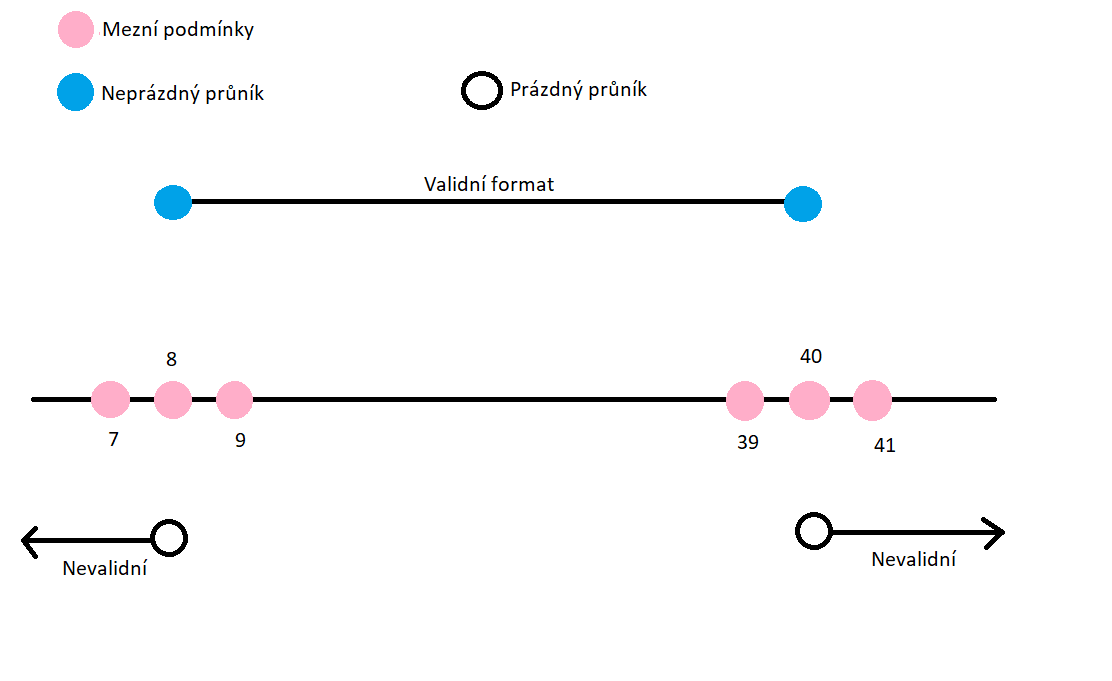
R = (A AND B AND C AND D)

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

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro password |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Zvláštní znaky * Méně než 8 znaků * Více než 40 znaků * Neodpovídá předtím zadanému heslu |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Password ve validním formátu, ale t toto heslo je velmi běžné |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující heslo ve validním formátu, déle než 7 znaků a zároveň odpovídá předtím zadanému heslu |

### Mezní podmínky



### 2.1.6 Vstup „Country “

Pokud zadané zemi není prázdné a obsahuje pouze písmena, potom bude vyhodnoceno jako validní.

* IF (zadané zemi není prázdné) AND (zadané zemi obsahuje pouze písmena) THEN je validní.

R = (A AND B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Možné kombinace | | | |
| 1 | 2 | 3 | 4 |
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 1 |

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro country |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Zvláštní znaky |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Country ve validním formátu, ale takové jméno neexistuje |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující země ve validním formátu |

### 2.1.7 Vstup „Date of birth “

Pokud zadané datum narození není prázdné a dříve než aktuální datum, potom bude vyhodnoceno jako validní.

* IF (zadané datum narození není prázdné) AND (zadané datum narození dříve než aktuální datum, ale od 1900) THEN je validní.

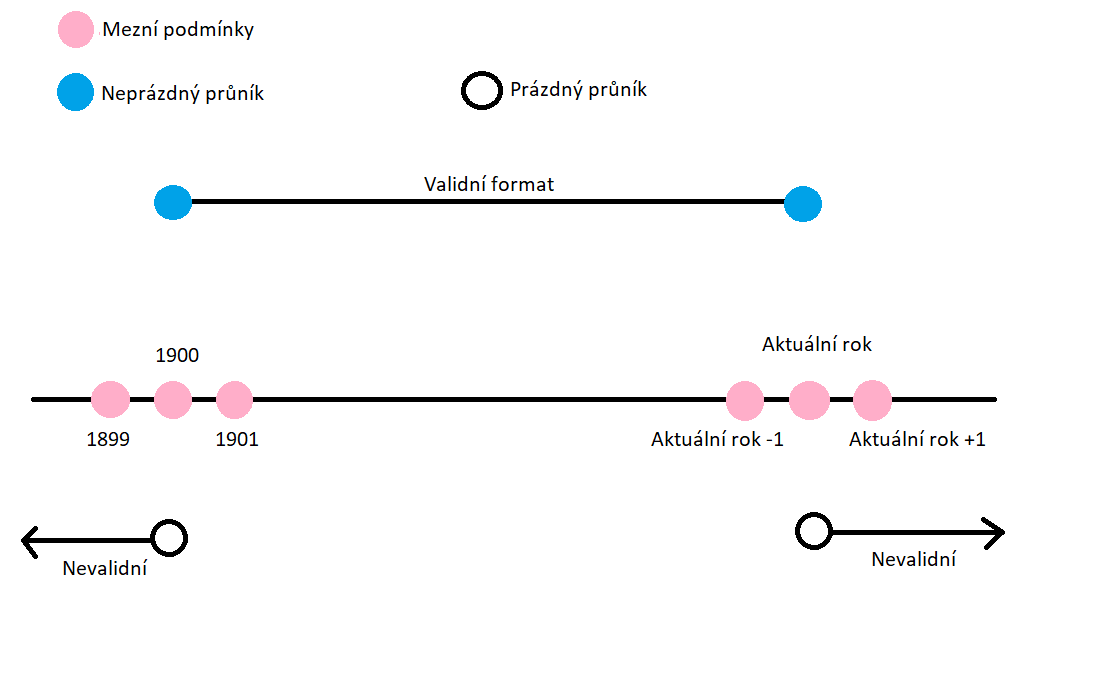
R = (A AND B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Možné kombinace | | | |
| 1 | 2 | 3 | 4 |
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 1 |

### Typy tříd ekvivalence – podle validity dat:

|  |  |  |
| --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Příklad: políčko pro birthday |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | * Prázdné políčko * Datum narození později než aktuální datum a dříve než 1900 rok |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | * Datum narození ve validním formátu, ale uživatel je buď příliš mladý nebo příliš starý |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | * Existující datum narození ve validním formátu a zároveň dříve než aktuální datum |

### Mezní podmínky



### 2.1.8 Vstup „Gender “

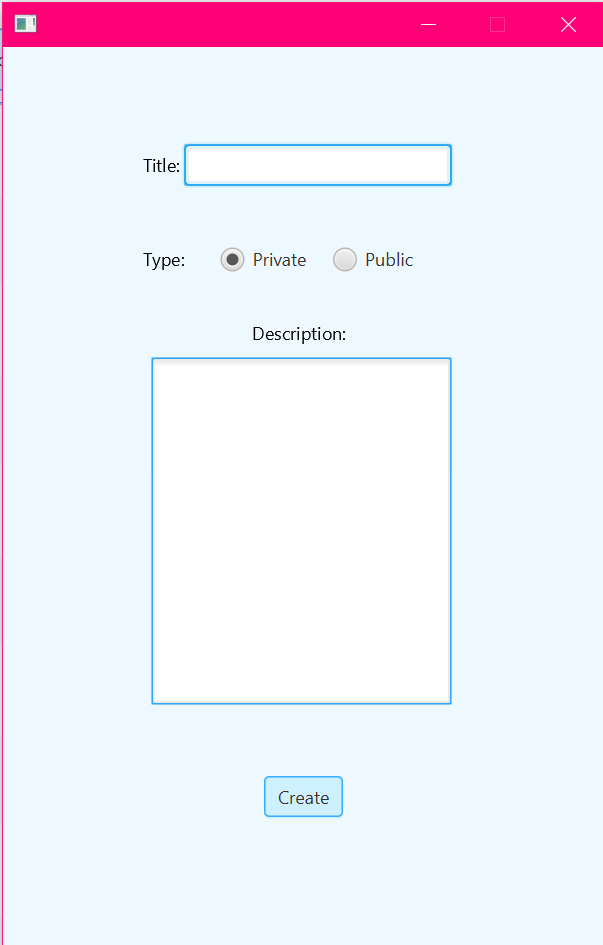
Pokud je pole vyplněné, je validní .

* IF (je pole vyplněné) THEN je validní.

R = (A)

|  |  |  |
| --- | --- | --- |
|  | Možné kombinace | |
| 1 | 2 |
| A | 0 | 1 |
| R | 0 | 1 |

### **Formulář vytvoření skupiny.**



### 2.1.1 Vstup „Title “

Pokud zadaný název není prázdný, obsahuje pouze písmena a zároveň není v databázi, potom bude vyhodnoceno jako validní.

* IF (zadaný název není prázdný) AND (zadaný název obsahuje pouze písmena) AND (zadaný název ještě není v databázi) THEN je validní.

R = (A AND B AND C)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Možné kombinace | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

### 2.1.2 Vstup „Description“

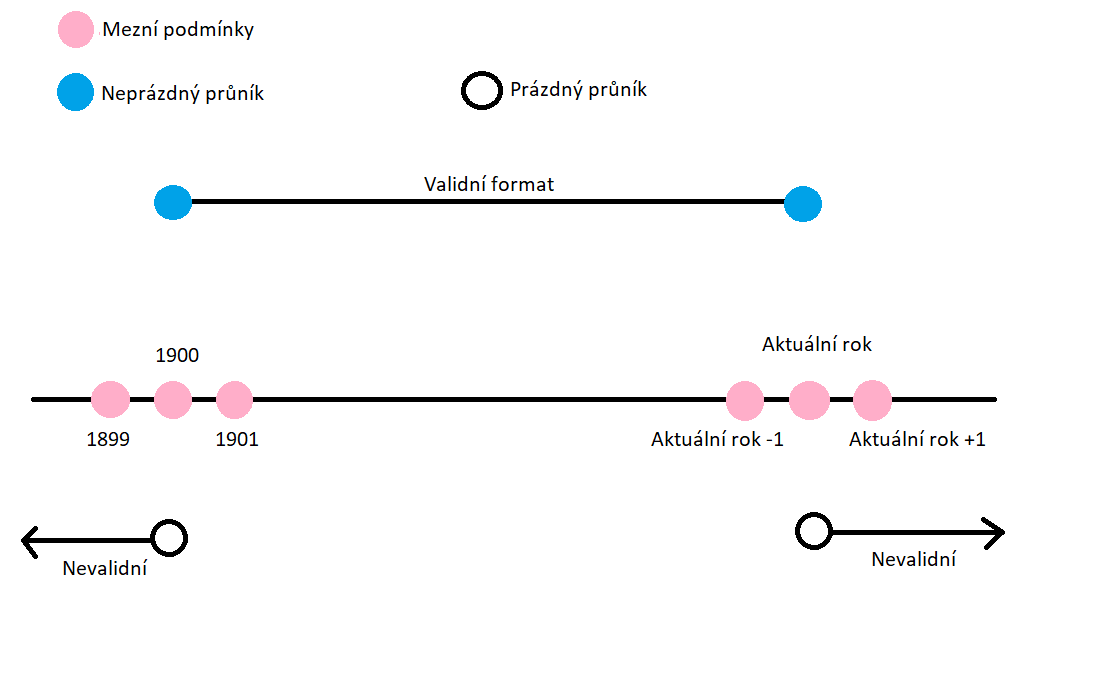
Pokud zadaný popis není prázdný a obsahuje méně než 1000 znaků, potom bude vyhodnoceno jako validní.

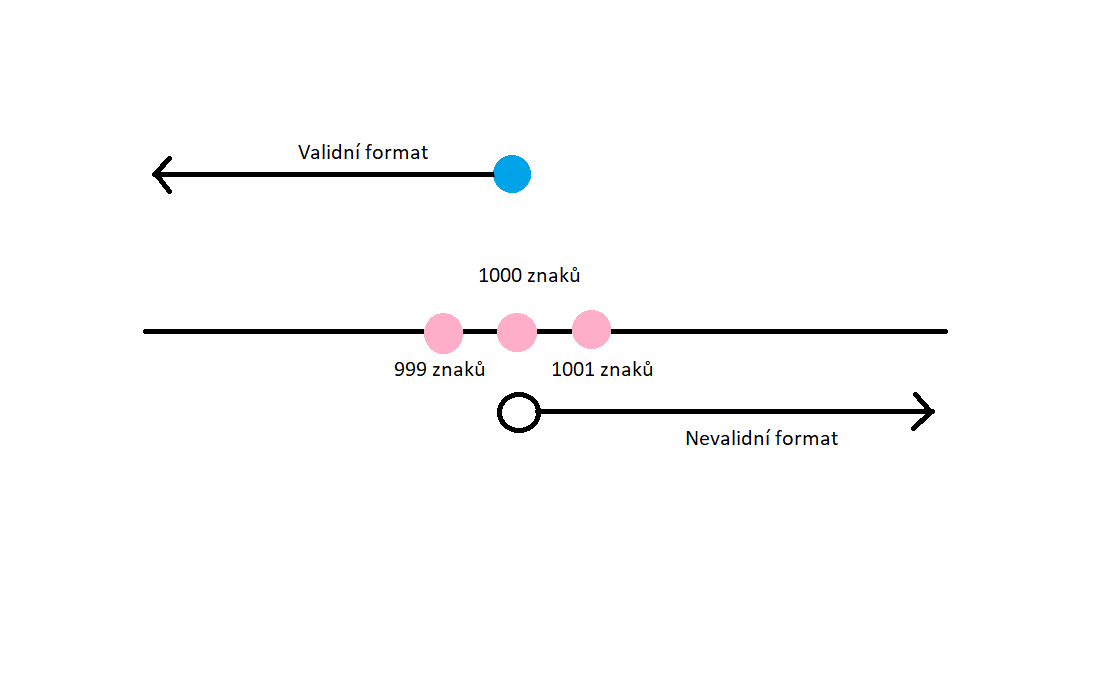
* IF (zadaný popis není prázdný) AND (zadaný popis obsahuje méně než 1000 znaků) THEN je validní.

R = (A AND B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Možné kombinace | | | |
| 1 | 2 | 3 | 4 |
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 1 |

### Mezní podmínky pro Popis





### 2.1.3 Vstup „Type “

Pokud je pole vyplněné, je validní .

* IF (je pole vyplněné) THEN je validní.

R = (A)

|  |  |  |
| --- | --- | --- |
|  | Možné kombinace | |
| 1 | 2 |
| A | 0 | 1 |
| R | 0 | 1 |

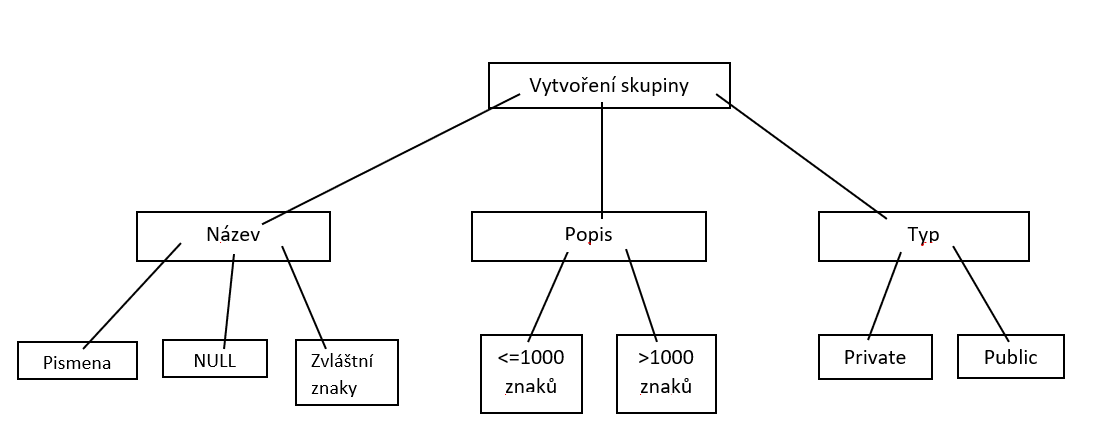
### Typy tříd ekvivalence – podle validity dat:

|  |  |  |  |
| --- | --- | --- | --- |
| Typ třídy ekvivalence | Co znamená pro aplikaci | Název parametru | Příklad: políčko pro Create group |
| Nevalidní třída ekvivalence z  technického pohledu | Data, která neodpovídají datovému typu vstupu =  neplatná data, která aplikace musí ošetřit, aby  nezpůsobila pád | Název | * Prázdné políčko * Zvláštní znaky |
| Popis | * >1000 znaků |
| Nevalidní třída ekvivalence z  business pohledu | Data, která sice odpovídají datovému typu vstupu,  ale z pohledu specifikace business procesu jsou  nevalidní | Název | * Název ve validním formátu,ale takové slovo nebo kombinace slov neexistuje |
| Popis | * Popis ve validním formátu,ale taková kombinace slov neexistuje |
| Typ | * Typ byl vybrán nesprávně |
| Validní třída ekvivalence | Platná data, která mají být zpracovávána podle  business specifikace, vyvolávají korektní průběhy  procesů v aplikaci | Název | * Název ve validním formátu a má smysl |
| Popis | * Má méně než 1000 znaků |
| Typ | * Typ byl vybrán správně |

## 2.2 Testovací situace pro vstupy (testy vstupů - kombinace dat).

### **Pairwise testing.**

Klasifikační strom pro vstupy:



|  |  |  |
| --- | --- | --- |
| Název parametru | Validní třídy ekvivalence | Zkrácené příklady hodnot |
| Název skupiny | Pismena, prázdné pole, zvláštní znaky | Group, NULL, g“1up |
| Popis skupiny | Platný počet znaků, neplatný počet znaků | 1000, 1001 |
| Typ skupiny | Private, public | Private, Public |

Kompletní pokrytí všemi možnými kombinacemi – 12 variant k otestování:

|  |  |  |  |
| --- | --- | --- | --- |
| Pořadí | Název (3) | Popis (2) | Typ (2) |
| 1 | Group | 1000 | Private |
| 2 | Group | 1000 | Public |
| 3 | Group | 1001 | Private |
| 4 | Group | 1001 | Public |
| 5 | NULL | 1000 | Private |
| 6 | NULL | 1000 | Public |
| 7 | NULL | 1001 | Private |
| 8 | NULL | 1001 | Public |
| 9 | g“1up | 1000 | Private |
| 10 | g“1up | 1000 | Public |
| 11 | g“1up | 1001 | Private |
| 12 | g“1up | 1001 | Public |

## 2.3 Detailní testovací scénáře.

### Pozitivní průchod

|  |  |
| --- | --- |
| Parametr | Obsah |
| ID testu | 1 |
| Název testu | Komunikace přes aplikace. |
| Krátký popis | Vytvoření účtu -> vytvoření skupiny -> přidání uživatelů do skupiny -> komunikace -> logout. |
| Hloubka detailu | Vysoká |
| Shrnutí testu | Odeslaní zprávy, pozitivní průchod |
| Popis testu | 1. Uživatel spustí aplikaci. 2. Uživatel klikne na tlačítko “Sig Up”. 3. Uživatel vyplní všechna povinná pole pro vytvoření nového účtu. 4. Uživatel klikne na tlačítko “Submit”. 5. Uživatel se dostane do okna Log In. 6. Uživatel vyplní všechna povinná pole pro přihlášení ke svému účtu. 7. Uživatel klikne na tlačítko “Submit”. 8. Uživatel se dostane do hlavního okna aplikace a mu budou zobrazeny všechny jeho skupiny. 9. Uživatel klikne na ikonku nabídky (menu). 10. Uživatel zaklikne “Create group” z nabídky. 11. Uživatel vyplní všechna povinná pole pro vytvoření nové skupiny. 12. Uživatel klikne na tlačítko “Create”. 13. Uživatel se dostane do hlavního okna aplikace a mu budou zobrazeny všechny jeho skupiny. 14. Uživatel vybere právě vytvořenou jim skupinu ze seznamu všech jeho skupin a klikne na ni. 15. Mu budou zobrazeny nové název skupiny, zprávy a nová ikonka pro úpravu skupinu. 16. Uživatel klikne na ikonku pro úpravu skupiny. 17. Uživatel klikne na “Group settings”. 18. V novém okně uživatel klikne na tlačítko “Add new member”. 19. V novém okně uživatel 20. Uživatel ve vyhledávacím řádku zadá jméno nebo příjmení uživatele, kterého chce přidat do skupiny. 21. Uživatel klikne na tlačítko “Search”. 22. Mu budou zobrazeny uživatelé a tlačítka “Invite“. 23. Uživatel vybere správného uživatele a klikne na tlačítko “Invite“. 24. V hlavním okně uživatel v řádku pro psaní zpráv napíše svou zprávu. 25. Uživatel klikne na tlačítko “Send”. 26. Uživatel si přečte odpověď svého partnera. 27. Uživatel klikne na ikonku nabídky (menu). 28. Uživatel zaklikne “Logout” z nabídky. 29. Uživatel se dostane do okna Log In. 30. Uživatel zavře aplikaci. |
| Očekávaný výsledek | Nový účet uživatele bude úspěšně vytvořen, nová skupina bude také úspěšně vytvořena, uživatel bude přidán ve skupinu, odeslána a přijata zpráva. Dojde k úspěšnému odhlášení z účtu. |
| Vstupní podmínky | Všechny údaje byly vyplněné a byly ve správném formatu. |
| Testovací data | * Registrace:   First Name: Arina  Last Name: Momot  E-mail: [arinamomot@gmail.com](mailto:arinamomot@gmail.com)  Password: 12345678  Repeat password: 12345678  Country: RU  Date of birth: 06.10.2000  Gender: F   * Log In:   Mail: [arinamomot@gmail.com](mailto:arinamomot@gmail.com)  Password: 12345678   * Create Group:   Title: Group  Description: The best group  Type: Public   * Add member:   Jack/Novak   * Message:   Hi, how are you? |

### Negativní průchod:

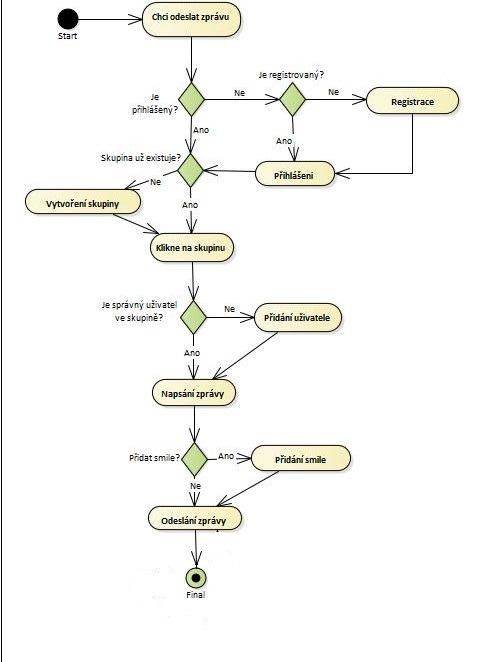
|  |  |
| --- | --- |
| Parametr | Obsah |
| ID testu | 1 |
| Název testu | Komunikace přes aplikace. |
| Krátký popis | Vytvoření účtu -> vytvoření skupiny -> přidání uživatelů do skupiny -> komunikace -> logout. |
| Hloubka detailu | Vysoká |
| Shrnutí testu | Zpráva není odeslaná , negativní průchod |
| Popis testu | 1. Uživatel spustí aplikaci. 2. Uživatel klikne na tlačítko “Sig Up”. 3. Uživatel vyplní všechna povinná pole pro vytvoření nového účtu. 4. Uživatel klikne na tlačítko “Submit”. 5. Uživatel se dostane do okna Log In. 6. Uživatel vyplní všechna povinná pole pro přihlášení ke svému účtu. 7. Uživatel klikne na tlačítko “Submit”. 8. Uživatel se dostane do hlavního okna aplikace a mu budou zobrazeny všechny jeho skupiny. 9. Uživatel klikne na ikonku nabídky (menu). 10. Uživatel zaklikne “Create group” z nabídky. 11. Uživatel vyplní všechna povinná pole pro vytvoření nové skupiny. 12. Uživatel klikne na tlačítko “Create”. 13. Uživatel se dostane do hlavního okna aplikace a mu budou zobrazeny všechny jeho skupiny. 14. Uživatel vybere právě vytvořenou jim skupinu ze seznamu všech jeho skupin a klikne na ni. 15. Mu budou zobrazeny nové název skupiny, zprávy a nová ikonka pro úpravu skupinu. 16. Uživatel klikne na ikonku pro úpravu skupiny. 17. Uživatel klikne na “Group settings”. 18. V novém okně uživatel klikne na tlačítko “Add new member”. 19. V novém okně uživatel 20. Uživatel ve vyhledávacím řádku zadá jméno nebo příjmení uživatele, kterého chce přidat do skupiny. 21. Uživatel klikne na tlačítko “Search”. 22. Mu budou zobrazeny uživatelé a tlačítka “Invite“. 23. Uživatel vybere správného uživatele a klikne na tlačítko “Invite“. 24. V hlavním okně uživatel v řádku pro psaní zpráv napíše svou zprávu. 25. Uživatel klikne na tlačítko “Send”. 26. Uživatel si přečte odpověď svého partnera. 27. Uživatel klikne na ikonku nabídky (menu). 28. Uživatel zaklikne “Logout” z nabídky. 29. Uživatel se dostane do okna Log In. 30. Uživatel zavře aplikaci. |
| Očekávaný výsledek | Zamitnutí transakci, zobrázení chybové hlášky |
| Vstupní podmínky | * Registrace: * Údaje nebyly vyplněné. * Všechny údaje byly vyplněný, ale ne v správném fornatu. * Všechny údaje byly vyplněný v správném fornatu, ale mail už existuje v DB. * Všechny údaje byly vyplněný v správném fornatu, ale hesla se neshoduje. * Log In: * Údaje nebyly vyplněné. * Všechny údaje byly vyplněný, ale ne v správném fornatu. * Všechny údaje byly vyplněný v správném fornatu, ale mail neexistuje v DB. * Všechny údaje byly vyplněný v správném fornatu, ale heslo není platné. * Create Group: * Údaje nebyly vyplněné. * Všechny údaje byly vyplněný, ale ne v správném fornatu. * Všechny údaje byly vyplněný v správném fornatu, ale title už existuje v DB. * Add member: * Údaje nebyly vyplněné. * Všechny údaje byly vyplněný, ale ne v správném fornatu. * Všechny údaje byly vyplněný v správném fornatu, ale uživatel neexistuje. * Message: * Údaje nebyly vyplněné. |
| Testovací data | * Registrace (příklad) :   First Name: Arina  Last Name: \_\_\_\_\_  E-mail: [arinamomot@gmail.com](mailto:arinamomot@gmail.com)  Password: 12345678  Repeat password: 123456789  Country: RU  Date of birth: 06.10.2000  Gender: F   * Log In:   Mail: \_\_\_\_\_\_  Password: 12345678   * Create Group:   Title: Group  Description: The best group  Type: Public   * Add member:   J3ač   * Message:   \_\_\_\_\_\_\_\_\_\_ |

## 2.4 Testy průchodů (path based).

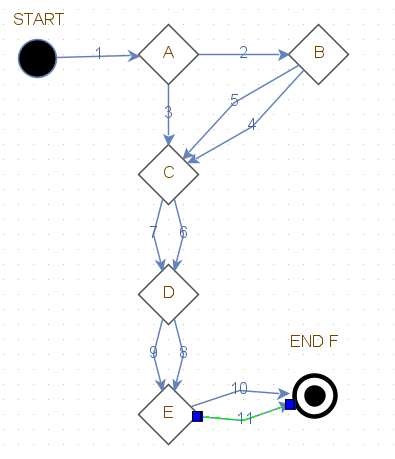
Zde jsou popsány možné průchody programem a jejich testované kombinace v rámci druhé úrovně pokrytí.

|  |  |
| --- | --- |
| Parametr | Obsah |
| ID testu | 1 |
| Název testu | Odeslaní zprávy. |
| Krátký popis | Uživatel se přihlásí do svého účtu, pokud to je nutné vytvoří novou skupinu a přidá tam uzivatele, pak odešle jím zravu. |
| Hloubka detailu | Střední |
| Očekávaný výsledek | Zpráva bude úspěšné odeslaná. |

* **Popis testu (Proces napsání zprávy) :**

****

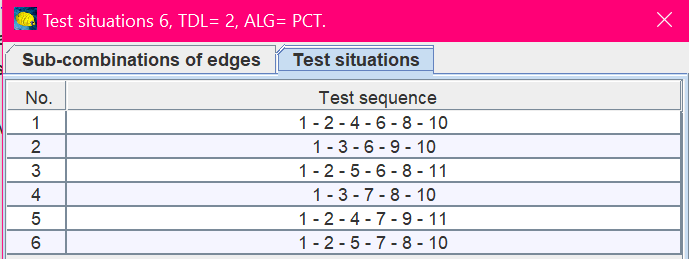
* **Zjednodušení struktury:**



1. Je přihlášený?
2. Je registrovaný?
3. Skupina už existuje?
4. Je správný uživatel ve skupině?
5. Přidat smile?

* **Vytvoření procesních testů (TDL=2) :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Větvící body | Vstupní akce | Výstupní akce | Kombinace pro test | Testovací scénáře |
| A | 1 | 2  3 | 1-2, 1-3 | 1-2-4-6-8-10 |
| B | 2 | 4  5 | 2-4, 2-5 | 1-3-6-9-10 |
| C | 3  4  5 | 6  7 | 3-6, 4-6, 5-6  3-7, 4-7, 5-7 | 1-2-5-6-8-11 |
| D | 6  7 | 8  9 | 6-8, 6-9  7-8, 7-9 | 1-3-7-8-10 |
| E | 8  9 | 10  11 | 8-10, 8-11  9-10, 9-11 | 1-2-4-7-9-11 |
| F (Konec) | 10  11 |  |  | 1-2-5-7-8-10 |



1. **Scénář:** Po spuštění aplikace uživatel udělá nový účet(vyplní formulář a stiskne tlačitko „Sign Up“), pak se do něho přihlásí, vytvoří novou skupinu, přidá tam nového uživatele, napíše zprávu se smilem a odešle zprávu .
2. **Scénář:** Uživatel již používá aplikaci, najde již existující skupinu, napíše zprávu se smilem a odešle zprávu .
3. **Scénář:** Po spuštění aplikace uživatel se přihlásí do svého účtu, najde již existující skupinu, přidá tam nového uživatele, napíše zprávu bez smilu a odešle zprávu .
4. **Scénář:** Uživatel již používá aplikaci, vytvoří novou skupinu, přidá tam nového uyivatele napíše zprávu se smilem a odešle zprávu .
5. **Scénář:** Po spuštění aplikace uživatel udělá nový účet(vyplní formulář a stiskne tlačitko „Sign Up“), pak se do něho přihlásí, najde již existující skupinu, přidá tam nového uyivatele napíše zprávu bez smilu a odešle zprávu .
6. **Scénář:** Po spuštění aplikace uživatel se přihlásí do svého účtu, vytvoří novou skupinu, přidá tam nového uyivatele napíše zprávu se smilem a odešle zprávu .