

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

# Testování webové aplikace Notes

Semestrální práce  
Předmět: Zajištění kvality software

Autor: Arina Momot  
Datum: 01.02.2023

---

<b>1. Návrh testovací strategie</b>	<b>2</b>
1.1. Popis aplikace	2
1.2. Testovací technologií	2
1.2.1. Framework	2
1.2.2. Pomocný testovací technologií	3
<b>2. Testování aplikace</b>	<b>3</b>
2.1. Třídy ekvivalence a mezní podmínky	3
2.1.1. Registrační formulář	3
2.1.2. Formulář pro vytvoření poznámky	12
2.2. Kombinace vstupních dat	16
2.2.1. Formulář pro vytvoření poznámky	16
2.2.2. Registrační formulář	17
2.3. Testování procesů	19
<b>3. Závěr</b>	<b>21</b>

# 1. Návrh testovací strategie

## 1.1. Popis aplikace

Pro testování jsem si vybrala a nasadila webovou aplikaci Notes, kterou jsem vytvořila někdy v minulosti v rámci bakalářského studia. Notes je jednoduchá webová aplikace pro psaní poznámek, která umožňuje uživatelům registrovat se na webu, vytvářet a mazat poznámky, přehrávat hudbu.

Aplikace je dostupná přes odkaz: <https://notes-vue-deploy-zks.vercel.app/>

Základní fungování webu se provádí pomocí skriptovacího jazyka JavaScript. Aplikace je postavena na frameworků Vue.js a Quasar.

Pro ukládání informací o poznámkách se používá LocalStorage. Také na webu používají takové technologie jako: History API (posun tlačítka zpět/vpřed prohlížeče), Media API (pro přehrávání hudby), JS API (pro zjišťování stavu připojení k internetu), JS práce se SVG (umožňuje přesunout objekt po stránce).

## 1.2. Testovací technologií

### 1.2.1. Framework

Pro provedení front-end testování webové aplikace Notes jsem zvolila open-source framework **Serenity BDD**, který poskytuje strukturovaný způsob testování webových aplikací pomocí testů stylu BDD (Behavior Driven Development), což umožňuje psaní testů v přirozenějším jazykovém formátu, který je snazší číst a pochopit.

Serenity se integruje s dalšími testovacími nástroji, jako jsou JUnit, který jsem také použila k testování. Tento framework poskytuje vestavěnou podporu pro Selenium WebDriver, který zjednodušil proces nastavení a spouštění webových testů. Serenity taky umožnil provádění testů paralelně.

Pro strukturování a snadnější čitelnost jsem použila vzory **Screenplay** a **PageObject**.

Screenplay pattern je a high-level návrhový vzor, který se zaměřuje na chování uživatele a interakce s aplikací. V tomto vzoru jsou testy psány v přirozenějším jazykovém formátu, který je snazší číst a pochopit. Vzor scénáře je založen na třech hlavních prvcích: Actor, Task, and Ability. Actor reprezentuje uživatele, Task reprezentuje specifickou akci, kterou může uživatel provést, a Ability reprezentuje uživatelskou schopnost nebo dovednost. Screenplay pattern poskytuje modulárnější a udržitelnější přístup k testování, protože testy jsou organizovány do malých, opakovaně použitelných komponent, které lze snadno upravovat a rozšiřovat.

Page Object pattern je low-level návrhový vzor, který se zaměřuje na UI komponenty uživatelského rozhraní aplikace. V tomto vzoru jsou UI komponenty uživatelského rozhraní zapouzdřeny do objektů, které představují různé stránky nebo sekce aplikace. Objekty stránky obsahují metody a vlastnosti potřebné k interakci s komponentami uživatelského rozhraní na stránce. Tento vzor poskytuje strukturovanější přístup k testování, protože testy jsou organizovány na základě různých stránek nebo částí aplikace.

### 1.2.2. Pomocný testovací technologií

- **ACTS** (Automated Combinatorial Testing for Software) je kombinatorický testovací nástroj, který jsem použila pro efektivně vytváření testovacích případů pro zajištění kvality a spolehlivosti softwaru.
- **Oxygen** umožnil vytváření modelů a automatizované generování testovacích případů založených na cestě pro aplikační procesy a pracovní postupy.
- **Gitlab pipeline** pro vytváření, testování a nasazení kódu konzistentním a automatizovaným způsobem.

## 2. Testování aplikace

### 2.1. Třídy ekvivalence a mezní podmínky

Tato kapitola se zabývá analýzou tříd ekvivalence a mezních podmínek formulářů aplikace.

Aplikace obsahuje 2 vstupní formuláře: registrační formulář a formulář pro vytváření poznámky.

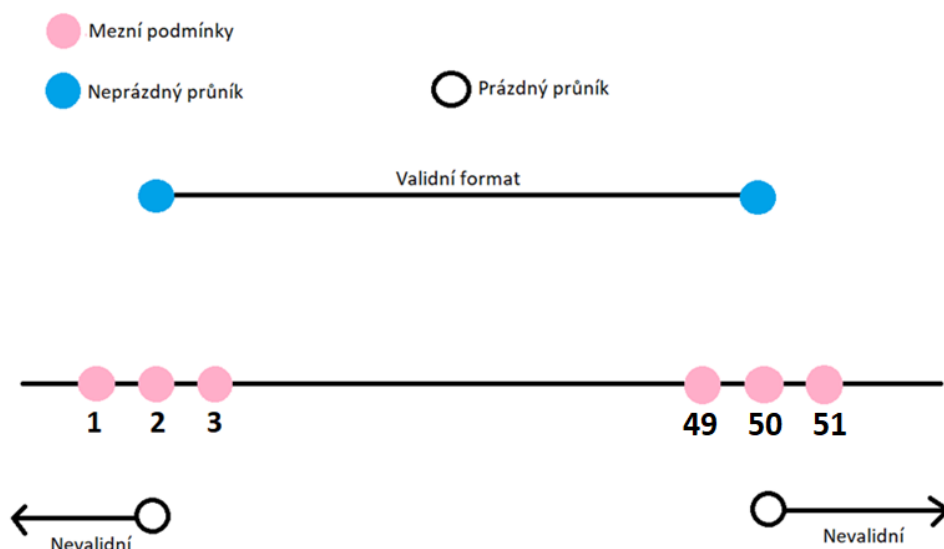
#### 2.1.1. Registrační formulář

The image shows a registration form with the following elements:

- Name \*** and **Surname \*** text input fields.
- Date of birth \*** field with a date mask `mm/dd/yyyy` and a calendar icon.
- City \*** and **University \*** dropdown menus.
- Email \*** text input field.
- Password \*** text input field with a visibility toggle icon.
- A password hint: "Use a combination of letters, numbers, and symbols - a total of at least 8 characters".
- A toggle switch and the text "I accept the license and terms".
- SUBMIT** and **RESET** buttons.

Name

Mezní podmínky podle délky:



Pokud zadané jméno splňuje následující RegEx: `^[a-zA-Z]+(?:['-][a-zA-Z]+)*$` a je delší než 1 znak a kratší než 51, potom bude vyhodnoceno jako validní.

- IF (zadané jméno splňuje RegEx) AND (je delší než 1 znak a kratší než 51) THEN je validní.

$$R = (A \text{ AND } B)$$

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

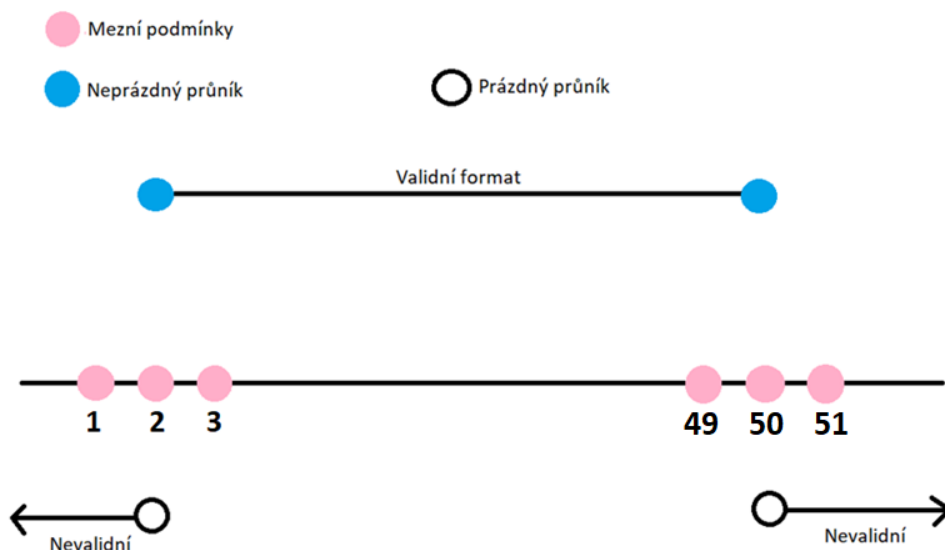
Typy tříd ekvivalence – podle validity dat:

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	<ul style="list-style-type: none"><li>• Prázdné políčko</li><li>• Zvláštní znaky (nesplňují RegEx)</li><li>• Kratší než 1 znak a delší než 50 znaků</li></ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"><li>• Jméno ve validním formátu, ale takové jméno neexistuje</li></ul>

Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>Existující jméno ve validním formátu</li> </ul>
---------------------------	---	--

Surname

Mezní podmínky podle délky:



Pokud zadané příjmení splňují následující RegEx: `^[a-zA-Z]+(?:[ '][a-zA-Z]+)*$` a je delší než 1 znak a kratší než 51, potom bude vyhodnoceno jako validní.

- IF (zadané příjmení splňují RegEx) AND (je delší než 1 znak a kratší než 51) THEN je validní.

$$R = (A \text{ AND } B)$$

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

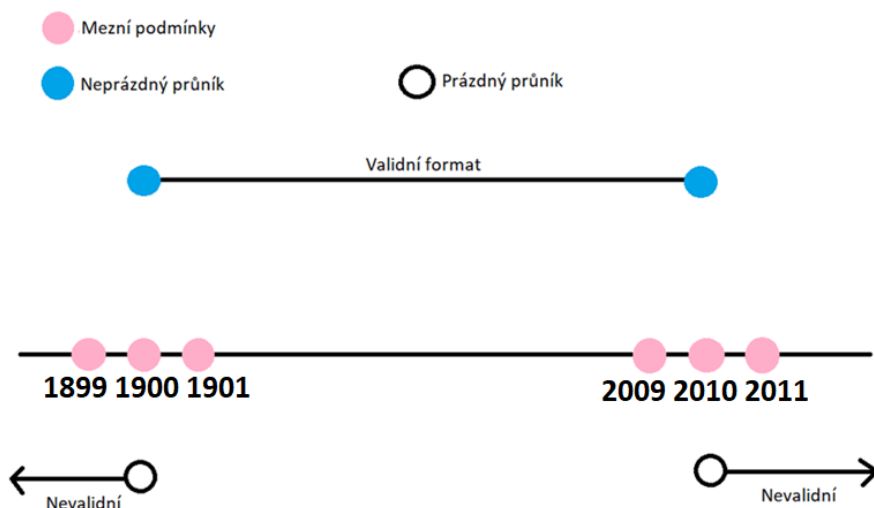
Typy tříd ekvivalence – podle validity dat:

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	<ul style="list-style-type: none"> <li>Prázdné políčko</li> <li>Zvláštní znaky (nesplňují RegEx)</li> <li>Kratší než 1 znak a delší než 50 znaků</li> </ul>

Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>Příjmení ve validním formátu, ale takové příjmení neexistuje</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>Existující příjmení ve validním formátu</li> </ul>

## Date of birth

Mezní podmínky podle roku narození:



Pokud zadané datum narození splňují omezení na rok narození (1900 - 2010) a není prázdné, potom bude vyhodnoceno jako validní.

- IF (zadané město splňují omezení na rok narození (1900 - 2010)) AND (je není prázdné) THEN je validní.

$$R = (A \text{ AND } B)$$

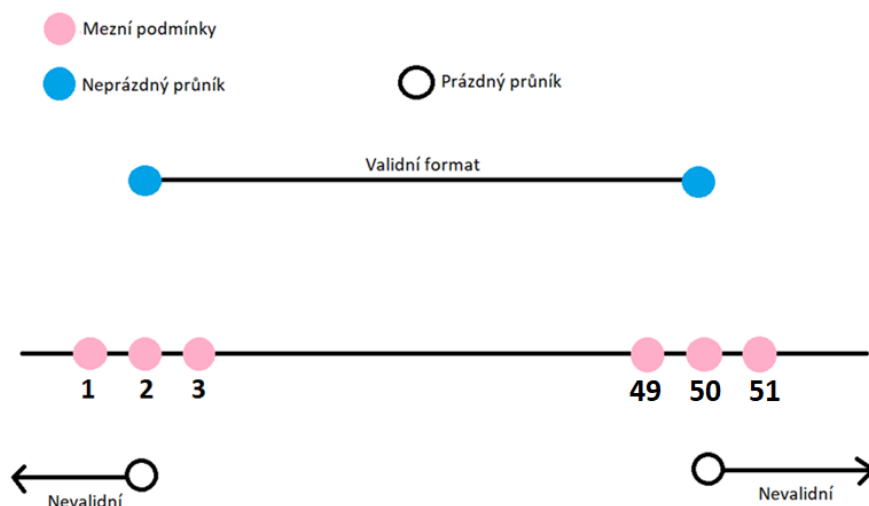
	Možné kombinace			
	1	2	3	4
A	0	0	1	1
B	0	1	0	1
R	0	0	0	1

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	<ul style="list-style-type: none"> <li>Prázdné políčko</li> <li>Nesplňují omezení na rok narození</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>Datum narození ve validním formátu, ale informace není pravdivá</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>Pravdivé datum narození ve validním formátu</li> </ul>

City

Mezní podmínky podle délky:



Pokud zadané město splňuje následující RegEx:  $^[A-Za-z]+(?:[\s-][A-Za-z]+)*\$$  a je delší než 1 znak a kratší než 51, potom bude vyhodnoceno jako validní.

- IF (zadané město splňuje RegEx) AND (je delší než 1 znak a kratší než 51) THEN je validní.

$$R = (A \text{ AND } B)$$

$$R = (A \text{ AND } B)$$

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

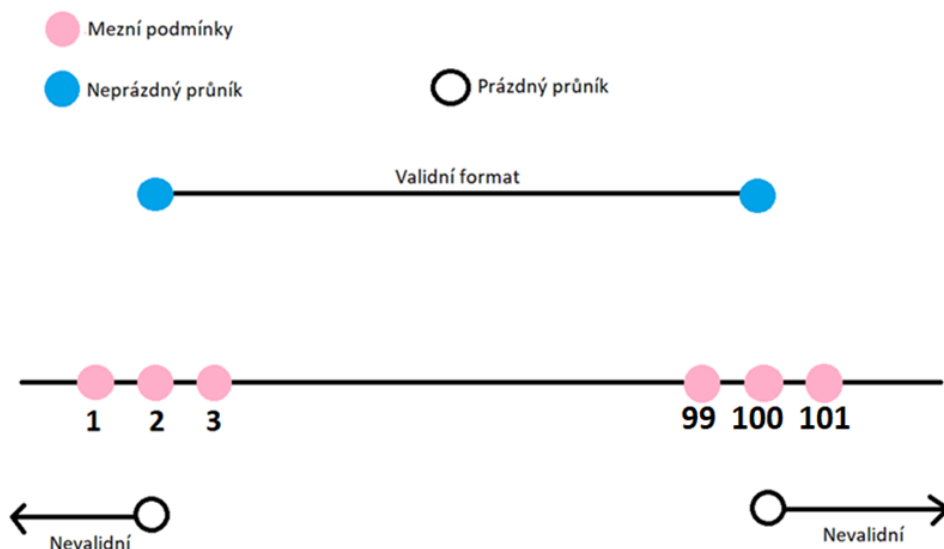


Typy tříd ekvivalence – podle validity dat:

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	<ul style="list-style-type: none"> <li>Prázdné políčko</li> <li>Zvláštní znaky (nesplňují RegEx)</li> <li>Kratší než 1 znak a delší než 50 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>Město ve validním formátu, ale takové město neexistuje</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracována podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>Existující město ve validním formátu</li> </ul>

## University

Mezní podmínky podle délky:



Pokud zadaná vysoká škola splňuje následující RegEx: `^[A-Za-z0-9\s, '-]*$` a je delší než 1 znak a kratší než 101, potom bude vyhodnoceno jako validní.

- IF (zadaná vysoká škola splňuje RegEx) AND (je delší než 1 znak a kratší než 101) THEN je validní.

$$R = (A \text{ AND } B)$$

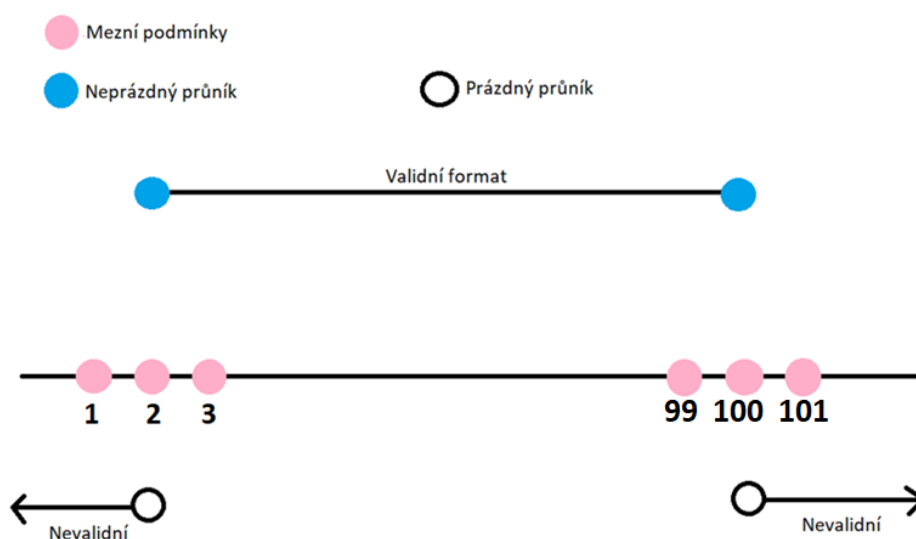
	Možné kombinace			
	1	2	3	4
A	0	0	1	1
B	0	1	0	1
R	0	0	0	1

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	<ul style="list-style-type: none"> <li>Prázdné políčko</li> <li>Zvláštní znaky (nesplňují RegEx)</li> <li>Kratší než 1 znak a delší než 100 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>Vysoká škola ve validním formátu, ale taková vysoká škola neexistuje</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>Existující vysoká škola ve validním formátu</li> </ul>

## Email

Mezní podmínky podle délky:



Pokud zadaný mail splňuje následující RegEx: `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$` a je delší než 1 znak a kratší než 101, potom bude vyhodnoceno jako validní.

- IF (zadaný mail splňuje RegEx) AND (je delší než 1 znak a kratší než 101) THEN je validní.

$$R = (A \text{ AND } B)$$

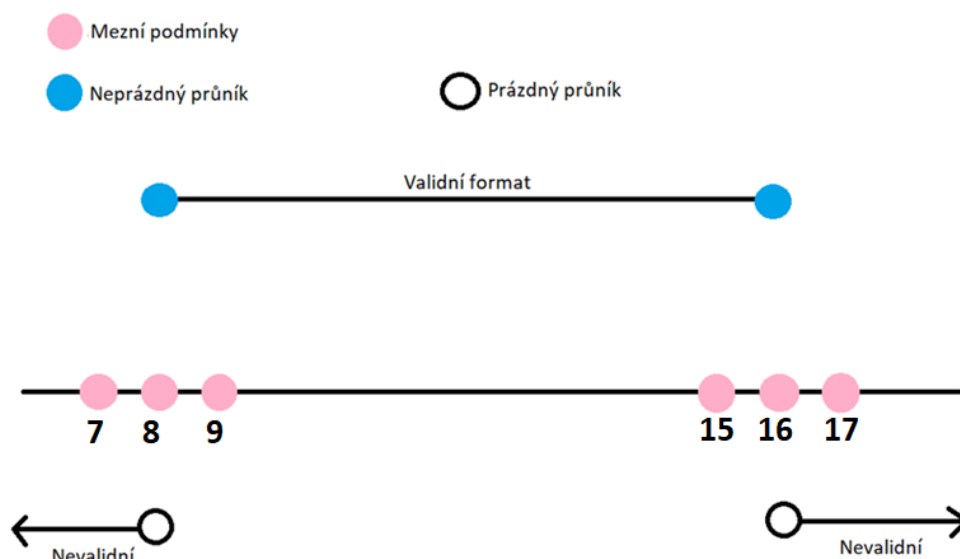
	Možné kombinace			
	1	2	3	4
A	0	0	1	1
B	0	1	0	1
R	0	0	0	1

Typy tříd ekvivalence – podle validity dat:

Typ třídy ekvivalence	Co znamená pro aplikaci	Příklad: políčko pro e-mail
Nevalidní třída ekvivalence z technického pohledu	Data, která neodpovídají datovému typu vstupu = neplatná data, která aplikace musí ošetřit, aby nezpůsobila pád	<ul style="list-style-type: none"> <li>• Prázdné políčko</li> <li>• Zvláštní znaky a není tam @ a . (nesplňují RegEx)</li> <li>• Kratší než 1 znak a delší než 100 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>• Mail ve validním formátu, ale takové mail neexistuje</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>• Existující mail ve validním formátu</li> </ul>

## Password

Mezní podmínky podle délky:



Pokud zadané heslo splňuje následující RegEx:

$^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#\$%^&*()_+.\'])S{8,}$$  a je delší než 7 znaků a kratší než 17, potom bude vyhodnoceno jako validní.

- IF (zadané heslo splňuje RegEx) AND (je delší než 7 znaků a kratší než 17) THEN je validní.

$$R = (A \text{ AND } B)$$

	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 password
Nevalidní třída ekvivalence z technického pohledu	Data, která neodpovídají datovému typu vstupu = neplatná data, která aplikace musí ošetřit, aby nezpůsobila pád	<ul style="list-style-type: none"> <li>• Prázdné políčko</li> <li>• Nesplňují RegEx</li> <li>• Méně než 8 znaků a více než 16 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>• Password ve validním formátu, ale toto heslo je velmi běžné</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>• Existující heslo ve validním formátu a zároveň déle než 7 znaků a kratší než 17 znaků + splňují RegEx</li> </ul>

## 2.1.2. Formulář pro vytvoření poznámky

Title

Title \*

Location

Location

Start date:

End date:

--:-- --

🕒

--:-- --

🕒

Start time

End time

mm/dd/yyyy

📅

mm/dd/yyyy

📅

Start date

End date

Description

Description \*

URL

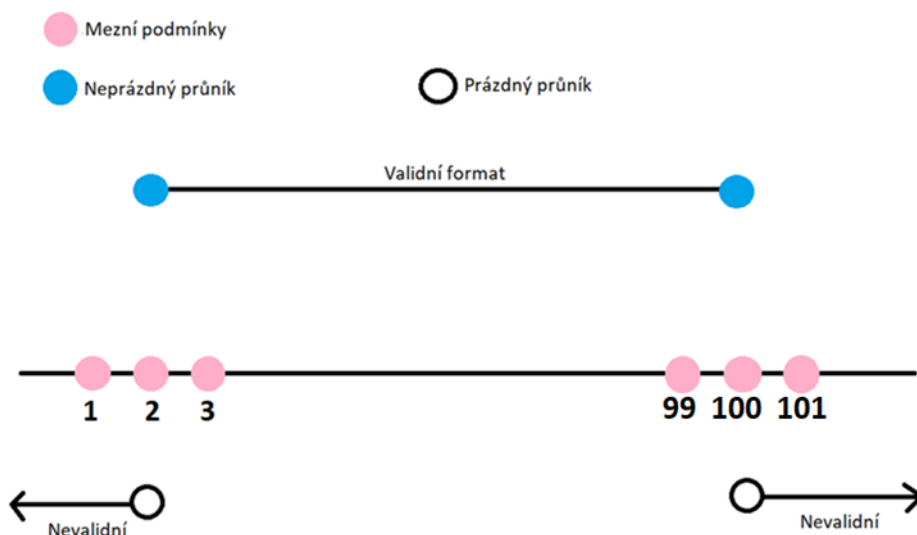
URL

ADD NOTE

RESET

Title

Mezní podmínky podle délky:



Pokud zadaný název není prázdný a je delší než 0 a kratší než 101, potom bude vyhodnoceno jako validní.

- IF (zadaný název není prázdný) AND (je delší než 0 a kratší než 101) THEN je validní.

$$R = (A \text{ AND } B)$$

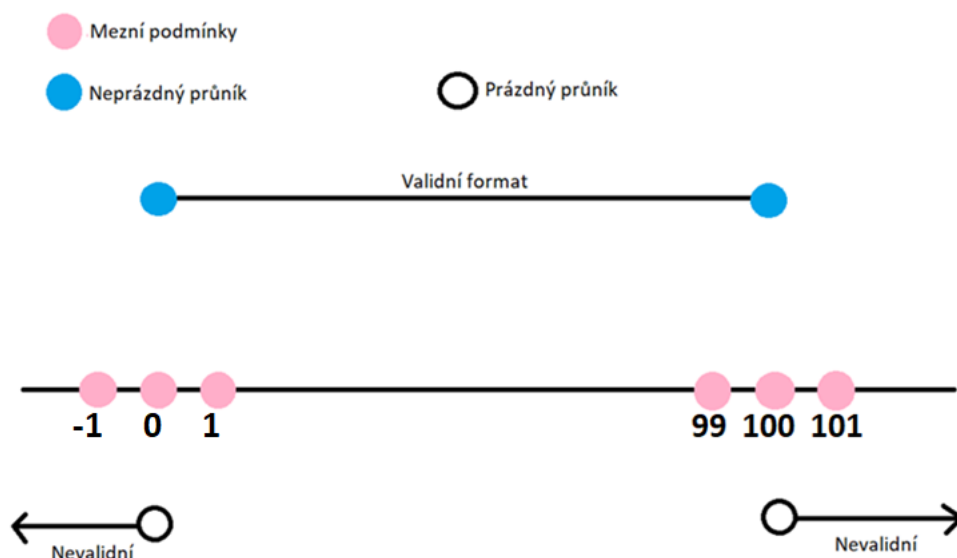
	Možné kombinace			
	1	2	3	4
A	0	0	1	1
B	0	1	0	1
R	0	0	0	1

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	<ul style="list-style-type: none"> <li>Prázdné políčko</li> <li>Delší než 100 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>Název ve validním formátu, ale takové jméno nemá smysl</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>Smysluplný název ve validním formátu</li> </ul>

## Location

Mezní podmínky podle délky:



Pokud zadaná lokaci splňují následující RegEx:  $^[\backslash w\backslash s]^*\$$  a je kratší než 101, potom bude vyhodnoceno jako validní.

- IF (zadaná lokaci splňují RegEx) AND (je kratší než 101) THEN je validní.

$$R = (A \text{ AND } B)$$

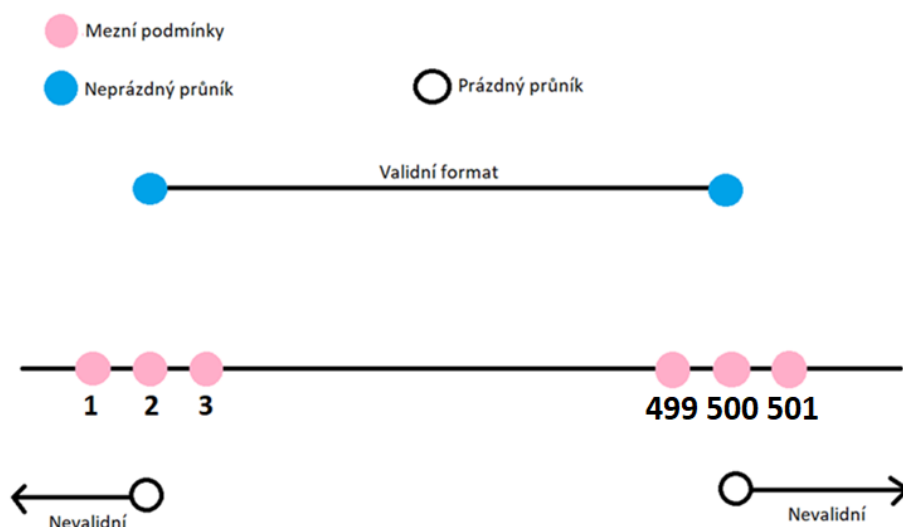
	Možné kombinace			
	1	2	3	4
A	0	0	1	1
B	0	1	0	1
R	0	0	0	1

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	<ul style="list-style-type: none"> <li>• Nesplňují RegEx</li> <li>• Delší než 100 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>• Lokace ve validním formátu, ale taková lokace neexistují</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>• Neexistující lokace ve validním formátu</li> </ul>

## Description

Mezní podmínky podle délky:



Pokud zadaný popis není prázdný a je delší než 0 a kratší než 501, potom bude vyhodnoceno jako validní.

- IF (zadaný popis není prázdný) AND (je delší než 0 a kratší než 501) THEN je validní.

$$R = (A \text{ AND } B)$$

	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	<ul style="list-style-type: none"> <li>• Prázdné políčko</li> <li>• Delší než 500 znaků</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>• Popis ve validním formátu, ale takové popis nemá smysl</li> </ul>
Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>• Smysluplný popis ve validním formátu</li> </ul>

## URL

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	<ul style="list-style-type: none"> <li>• Není ve formátu URL</li> </ul>
Nevalidní třída ekvivalence z business pohledu	Data, která sice odpovídají datovému typu vstupu, ale z pohledu specifikace business procesu jsou nevalidní	<ul style="list-style-type: none"> <li>• URL ve validním formátu, ale takové URL neexistují</li> </ul>



Validní třída ekvivalence	Platná data, která mají být zpracovávána podle business specifikace, vyvolávají korektní průběhy procesů v aplikaci	<ul style="list-style-type: none"> <li>• Smysluplný URL ve validním formátu</li> </ul>
---------------------------	---	--

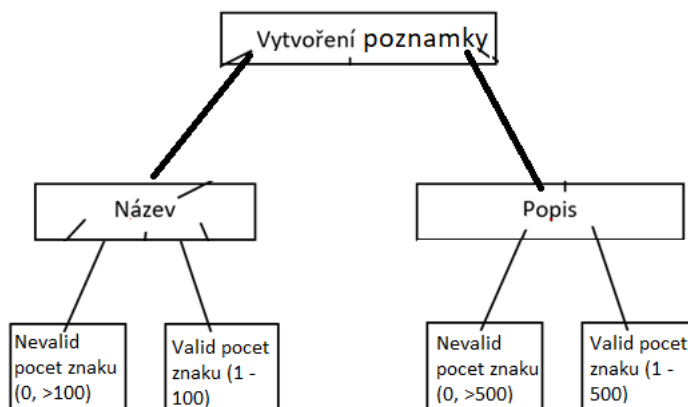
## 2.2. Kombinace vstupních dat

### 2.2.1. Formulář pro vytvoření poznámky

V tomto testu jsem se rozhodla otestovat pouze povinná pole ve formuláři vytvoření poznámky: název a popis podle počtu znaků.

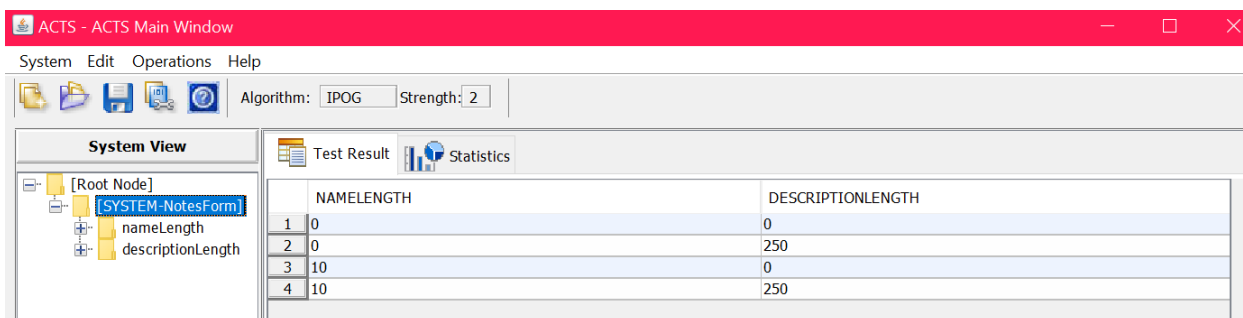
**Pairwise testing.**

**Klasifikační strom pro vstupy pro vytvoření poznámky:**



Název parametru	Třídy ekvivalence	Příklady hodnot	Vybraná hodnota
Název	Nevalidní	<1 , >100	0
	Validní	1-100	10
Popis	Nevalidní	<1 , >500	0
	Validní	1-500	250

## Model SUT pomocí ACTS tool (Strength coverage 2):



### 2.2.2. Registrační formulář

V tomto testu jsem se rozhodla otestovat formulář proregistrace uživatele podle validnosti dat.

#### Pairwise testing.

Název parametru	Třídy ekvivalence	Příklady hodnot	Vybraná hodnota
Jméno	Nevalidní	Zvláštní znaky	Arina1!
	Validní	Správný formát	Arina
Příjmení	Nevalidní	Zvláštní znaky	Momot2!
	Validní	Správný formát	Momot
Datum narození	Nevalidní	Nevalidní rok: <1900, >2010	2023
	Validní	Validní rok	2000
Město	Nevalidní	Zvláštní znaky	P2_@f
	Validní	Správný formát	Prague
Vysoká škola	Nevalidní	Zvláštní znaky	C@>dwe
	Validní	Správný formát	CVUT
Email	Nevalidní	Není ve formátu mailu	arimom@com
	Validní	Ve formátu mail	arimom@fel.cz
Heslo	Nevalidní	Nesplňuje požadavky na heslo (alespoň 8 znaků, alespoň 1 velká písmena, 1 malá písmena, 1 číslici a 1 speciální znak)	Arina123
	Validní	Splňuji požadavky na heslo	Arina123!

## Model SUT pomocí ACTS tool (Strength coverage 3):

ACTS - ACTS Main Window

System Edit Operations Help

Algorithm: IPOG Strength: 3

**System View**

- [Root Node]
  - [SYSTEM-NotesForm]
    - nameLength
    - descriptionLength
  - [SYSTEM-RegistrationForm]
    - Name
    - Surname
    - DateOfBirth
    - City
    - University
    - Email
    - Password

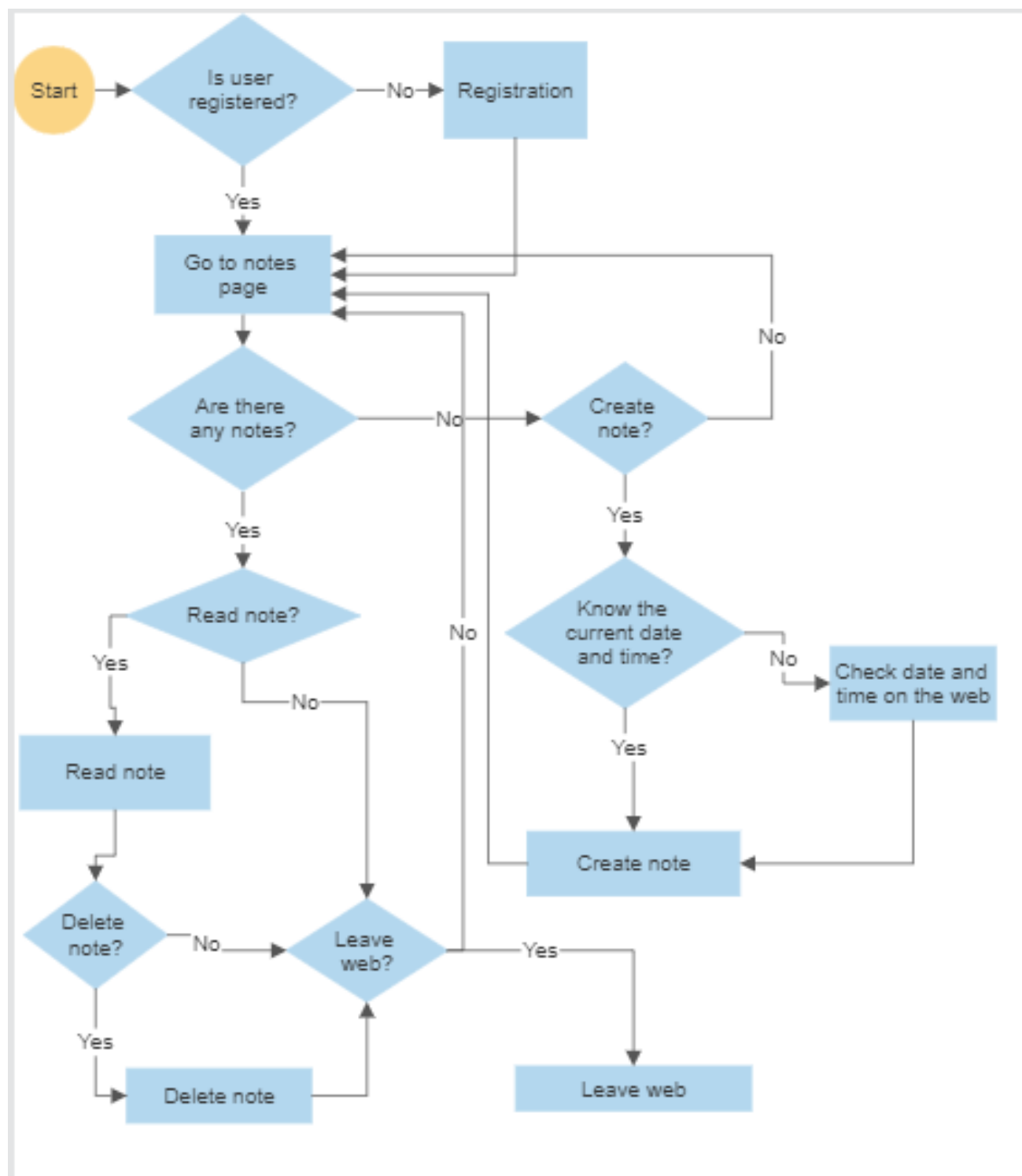
**Test Result** **Statistics**

	NAME	SURNAME	DATEOFBIRTH	CITY	UNIVERSITY	EMAIL	PASSWORD
1	Arina 1!	Momot2!	2023	P2_@f	C@>dwe	arimom@com	Arina123
2	Arina 1!	Momot2!	2000	Prague	CVUT	arimom@fel.cz	Arina123!
3	Arina 1!	Momot	2023	Prague	C@>dwe	arimom@fel.cz	Arina123
4	Arina 1!	Momot	2000	P2_@f	CVUT	arimom@com	Arina123!
5	Arina	Momot2!	2023	Prague	CVUT	arimom@com	Arina123
6	Arina	Momot2!	2000	P2_@f	C@>dwe	arimom@fel.cz	Arina123!
7	Arina	Momot	2023	P2_@f	CVUT	arimom@fel.cz	Arina123
8	Arina	Momot	2000	Prague	C@>dwe	arimom@com	Arina123!
9	Arina	Momot	2023	P2_@f	C@>dwe	arimom@com	Arina123!
10	Arina 1!	Momot2!	2000	Prague	C@>dwe	arimom@com	Arina123
11	Arina 1!	Momot2!	2023	P2_@f	CVUT	arimom@fel.cz	Arina123!
12	Arina	Momot	2000	Prague	CVUT	arimom@fel.cz	Arina123
13	Arina	Momot	2000	P2_@f	C@>dwe	arimom@com	Arina123
14	Arina 1!	Momot2!	2023	Prague	C@>dwe	arimom@com	Arina123!
15	Arina 1!	Momot2!	2000	Prague	CVUT	arimom@fel.cz	Arina123
16	Arina	Momot	2023	Prague	CVUT	arimom@fel.cz	Arina123!

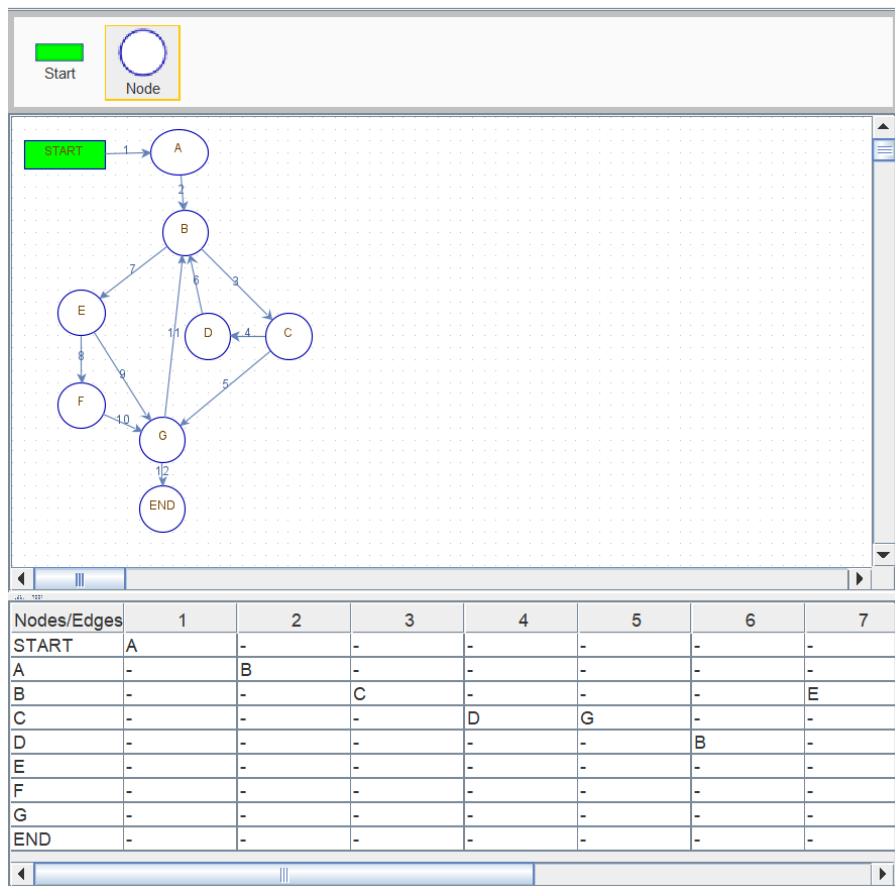
## 2.3. Testování procesů

Pro procesní testování jsem zvolila proces vytvoření a odstranění poznámky.

**Diagram průchodu:**



## Zjednodušení struktury:



- A - Is user registered?
- B - Are there any notes?
- C - Create note?
- D - Know the current date and time?
- E - Read note?
- F - Delete note?
- G - Leave web?

## Vygenerování průchodu (TDL=1) pomocí nástroje Oxygen :

Test situations 7, TDL= 1, ALG= PCT.	
Sub-combinations of edges	
Test situations	
No.	Test sequence
1	1 - 2 - 3 - 4 - 6 - 7 - 9 - 11 - 3 - 5 - 12
2	1 - 2 - 7 - 8 - 10 - 11 - 3 - 4 - 6 - 3 - 5 - 12

## Vygenerování průchodu (TDL=2) pomocí nástroje Oxygen :

Test situations 8, TDL= 2, ALG= PCT.	
Sub-combinations of edges	
Test situations	
No.	Test sequence
1	1 - 2 - 3 - 4 - 6 - 3 - 5 - 11 - 3 - 4 - 6 - 7 - 8 - 10 - 11 - 7 - 9 - 11 - 3 - 5 - 12
2	1 - 2 - 7 - 8 - 10 - 12
3	1 - 2 - 7 - 9 - 12

## Vygenerování průchodu (TDL=3) pomocí nástroje Oxygen :

Test situations 9, TDL= 3, ALG= PCT.	
Sub-combinations of edges	
Test situations	
No.	Test sequence
1	1 - 2 - 3 - 4 - 6 - 3 - 4 - 6 - 7 - 8 - 10 - 11 - 3 - 4 - 6 - 3 - 5 - 11 - 3 - 5 - 12
2	1 - 2 - 7 - 8 - 10 - 12
3	1 - 2 - 3 - 5 - 11 - 7 - 8 - 10 - 11 - 7 - 9 - 11 - 3 - 4 - 6 - 3 - 4 - 6 - 7 - 9 - 12
4	1 - 2 - 7 - 9 - 11 - 7 - 8 - 10 - 11 - 3 - 4 - 6 - 3 - 4 - 6 - 7 - 8 - 10 - 12

### 3. Závěr

V tomto testovacím reportu jsem identifikovala třídy ekvivalence a mezní podmínky pro formuláře webu Notes. Byl proveden test procesu, který zahrnuje hlavní funkci webu.

Testování ukázalo, že web je funkční a účinný. Různé testovací metody identifikovaly některé drobné problémy, které je třeba vyřešit, ale nepředstavují významnou hrozbu pro výkon produktu. Na základě výsledků testování se doporučuje opravit web pro optimalizaci jeho výkonu a uživatelské zkušenosti a znovu nasadit.