

Kingdom of development – Testovací dokumentace

Lukáš Vévar, ČVUT FEL, 28. 5. 2020

1. Popis funkcionality výsledné aplikace

Ve zkratce jde o blog vývojářských nástrojů. Kromě běžných informačních příspěvků, návodů se dá jednotlivým nástrojům napsat recenze.

Důležitá součást webové aplikace je otázkový „chatbot“, který se ptá na otázky a na základě odpovědí je schopen doporučit nástroj nebo informaci, která se uživateli hodí při vývoji určité věci.

Web obsahuje registraci a přihlášení. Pro vytváření příspěvků nebo přidávání nástrojů musí být člověk přihlášen.

Aplikace pro zatím nemá uživatelské role, proto každý registrovaný uživatel má stejné přístupy.

Zdrojový kód má dvě části:

- Frontend – Javascriptový kód psaný za pomoci frameworku React, a dalších vedlejších javascriptových knihoven
- Backend – Java REST aplikace, psaná pomocí známého frameworku Spring. Použité jsou i další vedlejší Java knihovny. Databáze je typu PostgreSQL.

Aplikace je ještě ve vývoji!

2. Návrh testovací strategie

Pozn.: Vztahuje se k aktuální verzi projektu.

2.1 Přehled částí aplikace

Část systému	Proces	Požadavek	Quality characteristic
Registrační modul	Uložení dat	Návštěvník, si může vytvořit účet, pomocí emailu, přezdívky a hesla.	Bezpečnost
Přihlašovací modul	Ověření dat	Registrovaný návštěvník se může přihlásit pomocí přezdívky a hesla pro zpřístupnění více možností aplikace	
Seznam příspěvků – blog modul	Získání dat	Návštěvník může prohlížet příspěvky uložené v aplikaci.	Informace
Přidání příspěvku	Uložení dat	Přihlášený návštěvník může přidávat nový příspěvek.	
Přidání kategorií	Uložení dat	Přihlášený návštěvník může přidávat nové kategorie příspěvků a otázek.	

Otázkový modul	Získání a uložení dat	Návštěvník odpovídá na otázku. Odpověď se ukládá do databáze.	
Přidání otázky	Uložení dat	Přihlášený návštěvník může přidávat nové otázky.	

2.2 Prioritizace částí aplikace

- Vzhledem k tomu, že se jedná o neziskovou aplikaci, není nutná bezchybná funkcionality.
- Důležitá je uživatelská přívětivost.

Část systému	Pravděpodobnost selhání	Vysvětlení	Priorita
Registrační modul	Střední	Jeden z paramterů registrace je IP adresa návštěvníka, která se chytá pomocí externí API se, kterou nejsou velké zkušenosti. Hesla se šifrují upraveným algoritmem, není jistá stoprocentní funkcionality.	Vysoká
Přihlašovací modul	Střední	Přihlášení probíhá pomocí JWT algoritmu. Který je implementován externí knihovnou.	Vysoká
Seznam příspěvků – blog modul	Nízká	Jedná se jednoduché získání dat.	Střední
Přidání příspěvku	Vysoká	Hrozí riziko spamu, nebo nechtěného obsahu.	Střední
Přidání kategorií	Střední	Kategorie jsou prozatím vedlejší a nebylo jim určeno velké pozornosti.	Nízká
Otázkový modul	Vysoká	Modul je vysoce nedokončený.	Nízká
Přidání otázky	Vysoká	Modul je vysoce nedokončený.	Nízká

Pozn.: Otázky a kategorie jsou pro aktuální verzi aplikace nepodstatné, ale v kompletní verzi budou velkou částí.

2.3 Test levels

Část systému	Priorita	Test levels				
		Revize	Vývojářské testy	Systémové testy	UAT	Testy v produkci
Registrační modul	Vysoká		Střední	Vysoká	Střední	Střední
Přihlašovací modul	Vysoká		Střední	Vysoká	Střední	Střední
Seznam příspěvků – blog modul	Střední		Nízká	Nízká	Střední	Nízká
Přidání příspěvku	Střední	ANO	Střední	Střední	Nízká	Střední
Přidání kategorií	Nízká		Nízká	Nízká	Nízká	Nízká

Otázkový modul	Nízká		Nízká	Nízká	Nízká	Nízká
Přidání otázky	Nízká		Nízká	Nízká	Nízká	Nízká

3. Testovací scénáře

3.1 Testy vstupů – třídy ekvivalence

Registrační formulář – registrace návštěvníka

Z důvodu absence netriviálních formulářů jsou zde vypsány alespoň některé “triviální.” Důležité pro testovací scénář jsou později při zpracování zadaných validních dat.

Register

Nickname

Email address

Password

Register!

1. Zvolení přezdívky

Typ třídy ekvivalence	Co znamená pro aplikaci	Příklad: input pro přezdívku
Nevalidní třída ekvivalence z technického pohledu	Zobrazí se hláška „špatná délka přezdívky” a doporučí správnou délku	Jeden znak Dvouciferné číslo Nevyplněno
Nevalidní třída z business pohledu	Zobrazí se hláška “přezdívka již existuje, zvolte prosím jinou”	Již existující přezdívka v databázi.
Validní třída ekvivalence	Správný průběh aplikací	Řetězec s 5 znaky.

Mezní podmínky:

X – vstup, řetězec

X.length – délka řetězce

X.length < 3 - nevalidní vstup

X.length > 17 - nevalidní vstup

3 <= *X.length* <= 17 – validní vstup

2. Zvolení emailu

Typ třídy ekvivalence	Co znamená pro aplikaci	Příklad: input pro email
Nevalidní třída ekvivalence z technického pohledu	Zobrazí se hláška „nesprávný formát emailu“ a vyzve uživatele k opravě	Jeden znak Dvouciferné číslo Nevyplněno
Nevalidní třída z business pohledu	Zobrazí se hláška “email již existuje, zvolte prosím jiný“	Již existující email v databázi.
Validní třída ekvivalence	Správný průběh aplikací	Řetězec ve formátu <code>*@*.*</code>

Mezní podmínky:

X – vstup, řetězec

$X \neq$ `*@*.*` - nevalidní vstup

$X ==$ `*@*.*` - validní vstup

3. Zvolení hesla

Typ třídy ekvivalence	Co znamená pro aplikaci	Příklad: input pro heslo
Nevalidní třída ekvivalence z technického pohledu	Zobrazí se hláška „špatná délka přezdívký“ a doporučí správnou délku	Jeden znak Dvouciferné číslo Nevyplněno
Nevalidní třída z business pohledu	Zobrazí se hláška “heslo neobsahuje číslo“	Řetězec pouze s abecedními znaky
Validní třída ekvivalence	Správný průběh aplikací	Řetězec s 8 znaky a číslem.

Mezní podmínky:

X – vstup, řetězec

$X.length$ – délka řetězce

$X.length < 6$ - nevalidní vstup

$X.length > 21$ - nevalidní vstup

$6 \leq X.length \leq 21$ – validní vstup

Přidání nového příspěvku přihlášeným uživatelem

Create new post

Title

Content

Type content

1. Zvolení názvu

Typ třídy ekvivalence	Co znamená pro aplikaci	Příklad: input pro název
Nevalidní třída ekvivalence z technického pohledu	Zobrazí se hláška „špatná délka názvu“ a doporučí správnou délku	Jeden znak Dvouciferné číslo Nevyplněno
Nevalidní třída z business pohledu	Zobrazí se hláška “příspěvek s tímto názvem již existuje, zvolte prosím jiný“	Již existující příspěvek se stejným názvem v databázi.
Validní třída ekvivalence	Správný průběh aplikací	Řetězec s 10 znaky.

Mezní podmínky:

X – vstup, řetězec

$X.length$ – délka řetězce

$X.length < 5$ - nevalidní vstup

$X.length \geq 5$ - validní vstup

Přidání nové kategorie přihlášeným uživatelem

Create new categories

New post category name

Current post categories:

- test

1. Zvolení názvu

Typ třídy ekvivalence	Co znamená pro aplikaci	Příklad: input pro název
Nevalidní třída ekvivalence z technického pohledu	Zobrazí se hláška „špatná délka názvu“ a doporučí správnou délku	Jeden znak Dvouciferné číslo Nevyplněno
Nevalidní třída z business pohledu	Zobrazí se hláška “kategorie s tímto názvem již existuje, zvolte prosím jiný“	Již existující příspěvek se stejným názvem v databázi.
Validní třída ekvivalence	Zobrazení vytvořené kategorie v seznamu pod formulářem	Řetězec s 4 znaky.

Mezní podmínky:

X – vstup, řetězec

X.length – délka řetězce

X.length < 4 - nevalidní vstup

X.length >= 4 - validní vstup

3.2 Testy vstupů – kombinace dat

Z důvodu absence netriviálních formulářů je zde alespoň “triviální” registrace.

Registrační formulář – registrace návštěvníka

Validní je kombinace číslo 17.

	Heslo	Přezdívková délka	Email
1	Méně než 6 znaků (nevalidní)	Méně než 3 znaky (nevalidní)	Má formát emailu (validní)
2	Méně než 6 znaků (nevalidní)	Méně než 3 znaky (nevalidní)	Nemá formát emailu (nevalidní)
3	Méně než 6 znaků (nevalidní)	Více než 17 znaků (nevalidní)	Má formát emailu (validní)
4	Méně než 6 znaků (nevalidní)	Více než 17 znaků (nevalidní)	Nemá formát emailu (nevalidní)
5	Méně než 6 znaků (nevalidní)	Řetězec má délku mezi 3 a 17 znaky. (validní)	Má formát emailu (validní)
6	Méně než 6 znaků (nevalidní)	Řetězec má délku mezi 3 a 17 znaky. (validní)	Nemá formát emailu (nevalidní)
7	Více než 21 znaků (nevalidní)	Méně než 3 znaky (nevalidní)	Má formát emailu (validní)

8	Více než 21 znaků (nevalidní)	Méně než 3 znaky (nevalidní)	Nemá formát emailu (nevalidní)
9	Více než 21 znaků (nevalidní)	Více než 17 znaků (nevalidní)	Má formát emailu (validní)
10	Více než 21 znaků (nevalidní)	Více než 17 znaků (nevalidní)	Nemá formát emailu (nevalidní)
11	Více než 21 znaků (nevalidní)	Řetězec má délku mezi 3 a 17 znaky. (validní)	Má formát emailu (validní)
12	Více než 21 znaků (nevalidní)	Řetězec má délku mezi 3 a 17 znaky. (validní)	Nemá formát emailu (nevalidní)
13	Řetězec má délku mezi 6 a 20 znaky. (validní)	Méně než 3 znaky (nevalidní)	Má formát emailu (validní)
14	Řetězec má délku mezi 6 a 20 znaky. (validní)	Méně než 3 znaky (nevalidní)	Nemá formát emailu (nevalidní)
15	Řetězec má délku mezi 6 a 20 znaky. (validní)	Více než 17 znaků (nevalidní)	Má formát emailu (validní)
16	Řetězec má délku mezi 6 a 20 znaky. (validní)	Více než 17 znaků (nevalidní)	Nemá formát emailu (nevalidní)
17	Řetězec má délku mezi 6 a 20 znaky. (validní)	Řetězec má délku mezi 3 a 17 znaky. (validní)	Má formát emailu (validní)
18	Řetězec má délku mezi 6 a 20 znaky. (validní)	Řetězec má délku mezi 3 a 17 znaky. (validní)	Nemá formát emailu (nevalidní)

3.3 Detailní testovací scénáře

Testovací scénář přidání příspěvku

- Otevřete aplikaci. V menu klikněte na tlačítko „Login“.
- Do inputu Nickname vložte „novakjan2“ a do inputu Password „aaa2“
- Nyní jste přihlášení. V menu byste měli vidět novou položku „Create Post“. Klikněte na ni.
- Do formuláře vyplňte data z následující tabulky. Očekávaný výsledek je ve sloupci výsledek:

Title	Content	Categories	Výsledek
<i>Nechte prázdný</i>	„Lorem ipsum“	Vyberte a zaškrtněte alespoň 2 kategorie ze seznamu.	Zobrazí se červený text „Title should have more than 5 characters“
„Lorem ipsum“	„Lorem ipsum“	<i>Nevybírejte žádné kategorie</i>	Zobrazí se červený text „Post should have at least one category“
„Lorem ipsum“	<i>Nechte prázdný</i>	Vyberte a zaškrtněte alespoň 2 kategorie ze seznamu.	Zobrazí se červený text „Post should have some content“
„Lorem ipsum“	„Lorem ipsum“	Vyberte a zaškrtněte alespoň 2 kategorie ze seznamu.	Příspěvek se přidal a jste přesměrováni na /blog

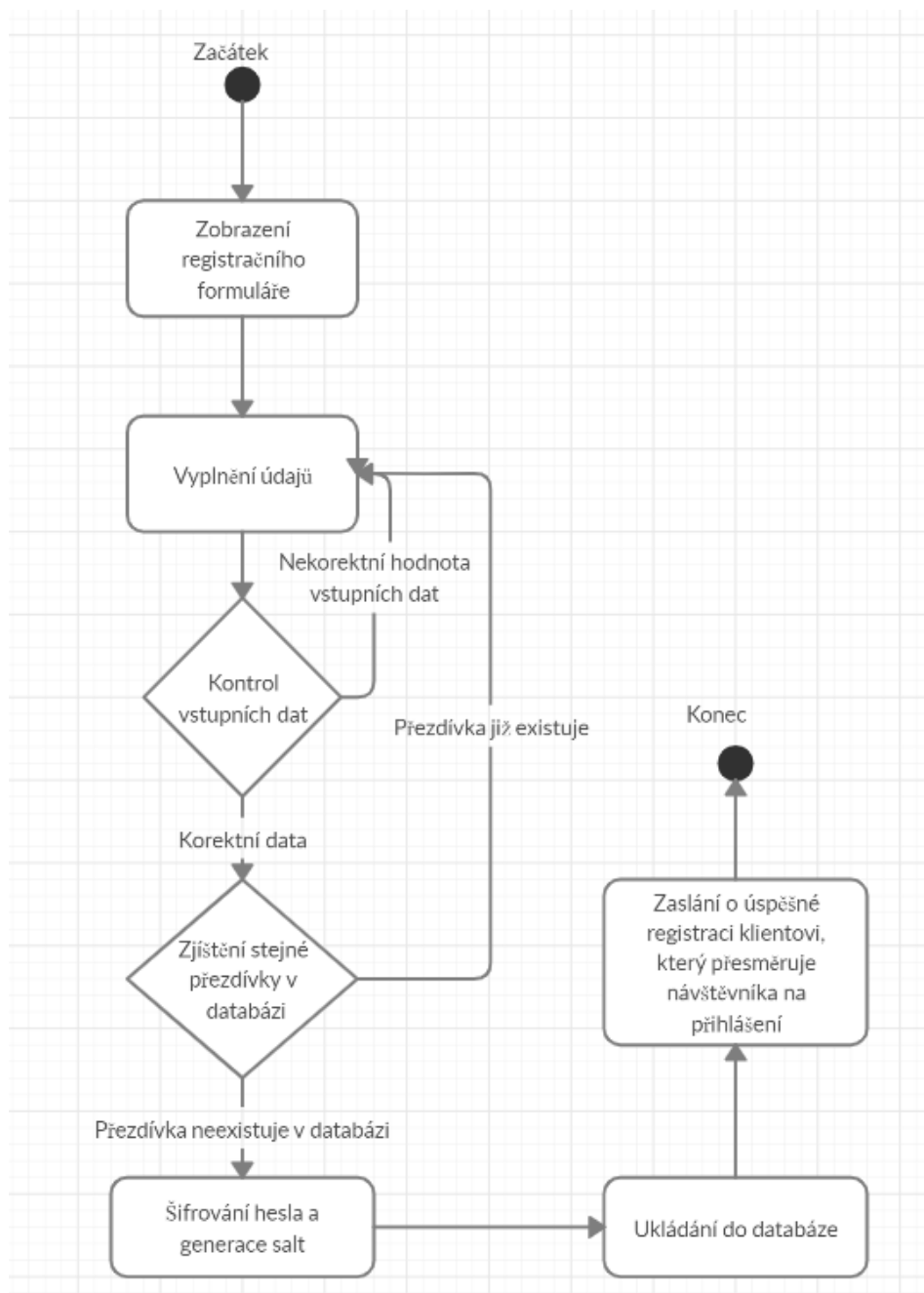
- Zkontrolujte přidání příspěvku, kliknutím na tlačítko „Blog“ v menu. Na posledním místě je vytvořený příspěvek.
- Klikněte na příspěvek.
- Zobrazí se detail příspěvku, zkontrolujte, že jsou přidány zvolené kategorie.

3.4 Testy průchodů

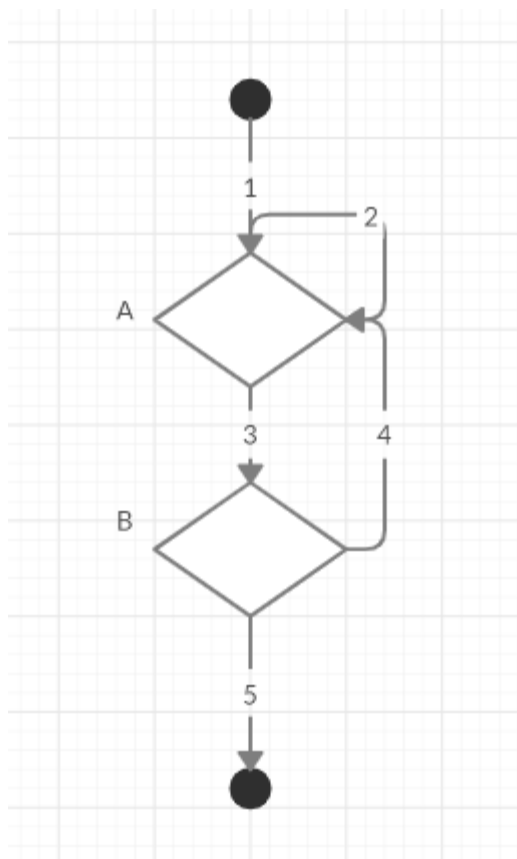
Registrace návštěvníka

UML diagram

Pozn: Používám internetový nástroj Creatly, z důvodu předchozích zkušeností.



Orientovaný graf



Větvící body	Vstupní akce	Výstupní akce	Kombinace pro test
A	1, 2, 4	2, 3	1-2, 2-2, 2-3
B	3	4, 5	3-4, 4-5

Testovací scénář

Zobrazení registračního formuláře – vyplnění údajů – kontrola vstupních údajů – nekorektní hodnota vstupních dat – vyplnění údajů - kontrola vstupních údajů – korektní data – zjištění stejné příjmení v databázi – příjmení již existuje - vyplnění údajů - kontrola vstupních údajů – korektní data – zjištění stejné příjmení v databázi – příjmení neexistuje v databázi – šifrování hesla a generace salt – ukládání do databáze – zaslání o úspěšné registraci klientovi, který přesměruje návštěvníka na přihlášení

4. JUnit testy v aplikaci

Pozn: Všechny metody se nachází v Backend (Java) části projektu, ve standartní test složce.

Přehled a krátký popis jednotlivých testů v aplikaci:

Název souboru	Název testované metody	Krátký popis
PostControllerTest	getAllPostCategories	Používá Mockito pro mockování databázového souboru PostDao. Testuje, že metoda vrátí správně očekávaný list příspěvků.
	getAllPostsFromCategory	Používá Mockito pro mockování databázového souboru PostCategoryDao. Testuje, že metoda vrátí správně očekávané příspěvky z dané kategorie.
QuestionControllerTest	getRandomQuestions	Používá Mockito pro mockování databázového souboru QuestionDao a třídy pro umožnění Random. Testuje, že metoda vrátí očekávanou otázku na "náhodné" pozici.
UserControllerTest	getAllVisitors	Používá Mockito pro mockování databázového souboru visitorDao. Testuje, že metoda vrátí správně očekávaný list návštěvníků.
	registerVisitor	Pokouší se registrovat návštěvníka s nevalidním heslem. Očekává errorový string v listu errorů.
JwtTokenTest	generateToken	Integrační test, kde používám framework Spring s knihovnou SpringSecurity a upravenou rozšířenou knihovnu Jwt. Pro generování tokenu. Testuje, že metoda vrátí validní token začínající očekávaným slovem.
JwtUserDetailsServiceTest	loadUserByUsername	Integrační test, kde používám framework Spring s knihovnou SpringSecurity. Testuje, že metoda vrátí správnou výjimku, když nenajde uživatele v databázi.
RegisteredVisitorTest	registerPassword	Procesní test, testující korektní šifrování hasla, které probíhá generováním saltu a následným hashováním. Testuje, že metoda vrátí stejný hash jako se očekává.
	registerValidation	Parametrizovaný procesní test, testuje správný proces vygenerování errorových stringů nevalidnímu registrovanému návštěvníkovi. Očekává že tento list errorů není prázdný.