

Semestrální práce z předmětu TS1

Testování webové aplikace MyPlant

Autoři: Ondřej Bureš a Olesia Cheremnykh
Školní rok: 2021/2022

Popis aplikace	4
Návrh testovací strategie	4
Části aplikace:	4
Prioritizace komponent	5
Test levels	6
Testovací scénáře	7
Vstupy aplikace	7
Registrační formulář	7
Přihlašovací formulář	8
Formulář pro vytvoření příspěvku	9
Přidat komentář	10
Testy průchodů	11
Proces registrace	11
Proces přidání příspěvku	13
Proces úpravy příspěvku	15
Proces odstranění příspěvku	17
Detailní testovací scénáře	19
Test case 1	19
Test case 2	21
Implementace testů	23
Unit testy	23
Používáme phpUnit s využitím syntaxe frameworku	23
Category_Model_test	23
Category_model_mocking_db_test	23
Post_Model_test	23
Post_model_mocking_db_test	23
User_Model_test	23
User_model_mocking_db_test	23
Comment_model_mocking_db_test	24
Welcome_test	24
Integrační testy	24
Testování registrace	24
Testování loginu	24
Testování CRUD pro příspěvky	24
Testování napsání komentářů	24
Testování vyhledání příspěvku podle kategorie a test funkčnosti filtrace	24

Testování správného nastavení LearnMore odkazů	25
Testování nočního a denního režimu	25
Testování stránkování a počtu článků	25

Popis aplikace

MyPlant je online platforma pro lidi, kteří mají rádi téma rostlin, díky které mohou sdílet fotku své domácí rostliny s celým světem. Aplikace uživatelům umožňuje registraci a následné přihlášení, přidání příspěvků, komentování příspěvků a nahrávání fotek. Dále aplikace nabízí filtrování v zobrazení kategorie příspěvků a obecné prolinkování na další tématické weby.

Aplikace je psána v duchu MVC architektury jako semestrální práce na předmět ZWA. Využívá základní webové technologie HTML, CSS a JavaScript pro validaci. Serverová část aplikace je pak napsána v jazyce PHP s využitím frameworku Codeigniter. Data jsou pak uloženy ve školním databázovém MySQL systému a přístup a manipulace s nimi je řešena přes aplikaci adminer <http://wa.toad.cz/adminer/>.

Detailnější popis aplikace a kompletní dokumentace je pak k nalezení zde:

<http://wa.toad.cz/~chereole/Documentation/MyPlant.pdf>

Návrh testovací strategie

Části aplikace:

Aplikace je členěna v klasické MVC architektuře, takže má základní dělení na části Model, View a Controller. Ty se pak dále dělí

Controller

Categories - Zajišťuje vyhledávání postů podle zvolené kategorie

Comments - Zajišťuje přidávání komentářů a jejich validaci

Pages - Vyvolává jednotlivá views, pokud existují

Posts - Zajišťuje správu postů, jejich přidávání, odebírání a upravování, dále zajišťuje stránkování volání view zobrazujícího záznamy z databáze.

Users - Zajišťuje autentizaci přihlášení a odhlášení a dále se stará o registraci a hashování komunikace.

Model

Category model - Výchozí model pro kategorie zajišťující funkce pro komunikaci s daty z databáze připravující data pro controller.

Comment model - Výchozí model pro komentáře zajišťující funkce pro komunikaci s daty z databáze připravující data pro controller.

Post model - Výchozí model pro příspěvky zajišťující funkce pro komunikaci s daty z databáze připravující data pro controller.

User model - Výchozí model pro uživatele zajišťující funkce pro komunikaci s daty z databáze připravující data pro controller.

View

Login - Zobrazení přihlašovacího formuláře

Register - Zobrazení registračního formuláře

Category Index - Zobrazuje všechny kategorie na stránce Categories

Home - Zobrazení hlavní stránky

Learn more - Zobrazuje stránku Learn More s prokliky na tématické stránky

Create - Zobrazuje formulář na přidání příspěvku

Edit - Zobrazuje upravovací formulář pro změnu příspěvku

Post Index - Zobrazuje stránku posts

View - Zobrazuje detail konkrétního příspěvku s příslušnými komentáři a formulářem pro vložení nového komentáře.

Prioritizace komponent

Zde jsme vzali v potaz jednotlivé programové složky naší aplikace a určili jsme důležitost fungování jednotlivých částí. Obecně se dá říci, že tím, že z důvodu velikosti aplikace je většina komponent klíčová pro celý běh.

Komponenta	Název	High priority	Medium priority	Low priority
Controller	Categories		X	
	Comments		X	
	Pages	X		
	Posts	X		
	Users	X		
Model	Category			X
	Comment		X	
	Post	X		
	User	X		
View	Login	X		
	Register	X		
	Category Index		X	
	Learn more			X
	Home			X
	Create	X		
	Edit		X	
	Post index	X		
	View	x		

Test levels

Quality characteristic Část systému /funkce	Třída rizika	Test levels					Unit testy	Procesní testy
		Nutná revize	Vývojářské testy	Systémové testy	UAT	Test v produkci		
Bezchybná funkcionalita								
Registrace	A			Střední	Střední		Ano	Ano
Login	A			Střední	Střední		Ano	Ano
Přidání příspěvku	A			Střední	Střední		Ano	Ano
Zobrazení příspěvků	A				Střední			Ano
Zobrazení detailu příspěvku	B	Ano			Střední			Ano
Okomentování příspěvku	B	Ano			Střední			Ano
Upravení příspěvku	B	Ano			Střední			Ano
Odstranění příspěvku	B				Střední			Ano
Filtrace podle kategorií	B			Nízká	Vysoká		Ano	Ano
Funkční odkazy v LearnMore	B				Vysoká			Ano
Načtení příspěvků z DB	B			Vysoká			Ano	
Uložení příspěvku do DB	B			Vysoká			Ano	
Uložení komentářů do databáze	B			Vysoká			Ano	
Uložení profilu do DB	B			Vysoká			Ano	
Načtení kategorií z DB	B			Vysoká			Ano	
Načtení uživatelů z DB	B			Vysoká			Ano	
Načtení komentářů z DB	B			Vysoká			Ano	
Uživatelská přívětivost								
Denní režim	C				Vysoká			Ano
Noční režim	C	Ano			Vysoká			Ano
Stránkování	B				Nízká			Ano
Výkonnost								
Přidání příspěvku	A							
Zobrazení příspěvků	A							
Zobrazení detailu příspěvku	B							
Okomentování příspěvku	B							
Upravení příspěvku	B							
Odstranění příspěvku	B							
Bezpečnost								
Data uživatelů	B							

Testovací scénáře

Sign Up

Enter your name*

Enter your email

Enter your username*

Enter your password*

Confirm your password*

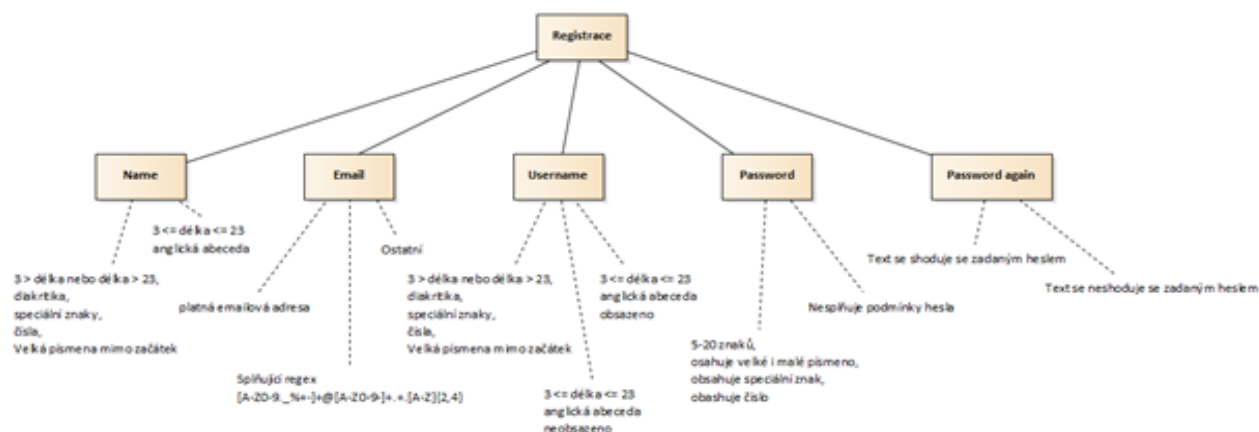
Submit

Vstupy aplikace

Registrační formulář

V tomto případě se jedná o standardní registrační formulář, kontrolovaný JS validací na frontendu a záložní preventivní PHP validací na serveru. Pokud se zaměříme na mezní podmínky, tak to budou minimální a maximální délka pro Name, Username a Password, u ostatních atributů pak mezní podmínky nelze určit

Vstupy podle EC pro registrační formulář



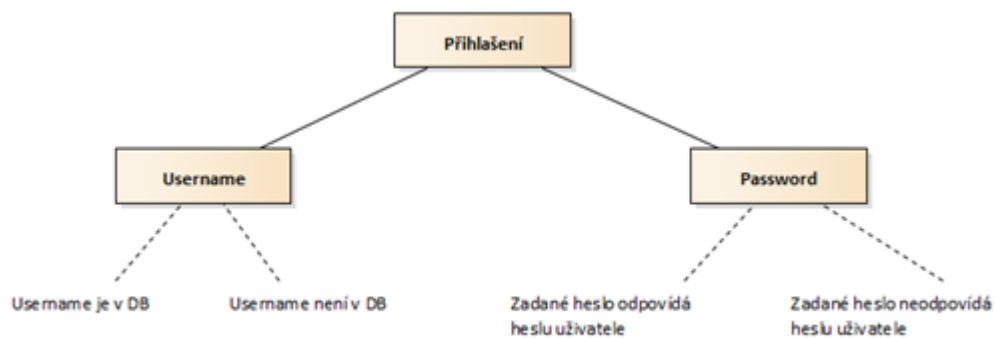
Pairwise test casey

case	Name	Email	Username	Password	Password again	pairings
1	správný formát	platná emailová adresa	spatný formát	heslo splňuje podmínky	heslo je shodné	10
2	spatný formát	platná emailová adresa	správný formát neobsazeno	heslo nesplňuje podmínky	heslo není shodné	10
3	spatný formát	neplatná emailová adresa	spatný formát	heslo splňuje podmínky	heslo není shodné	8
4	správný formát	platná emailová adresa	správný formát neobsazeno	heslo nesplňuje podmínky	heslo je shodné	8
5	správný formát	nesplňující formát emailu	správný formát obsazeno	heslo splňuje podmínky	heslo není shodné	8
6	spatný formát	nesplňující formát emailu	správný formát obsazeno	heslo nesplňuje podmínky	heslo je shodné	7
7	~správný formát	platná emailová adresa	spatný formát	heslo nesplňuje podmínky	~heslo není shodné	1
8	~spatný formát	platná emailová adresa	správný formát neobsazeno	heslo splňuje podmínky	~heslo je shodné	1
9	~správný formát	neplatná emailová adresa	správný formát obsazeno	~heslo splňuje podmínky	~heslo je shodné	1
10	~spatný formát	nesplňující formát emailu	spatný formát	~heslo nesplňuje podmínky	~heslo je shodné	1
11	~správný formát	nesplňující formát emailu	správný formát neobsazeno	~heslo splňuje podmínky	~heslo není shodné	1
12	~spatný formát	platná emailová adresa	správný formát obsazeno	~heslo nesplňuje podmínky	~heslo není shodné	1

Přihlašovací formulář

A green rounded rectangle containing the text "Sign In" at the top. Below it are two input fields: "Enter Username" and "Enter Password". At the bottom is a "Login" button.

podle EC pro přihlašovací formulář



Vstupy

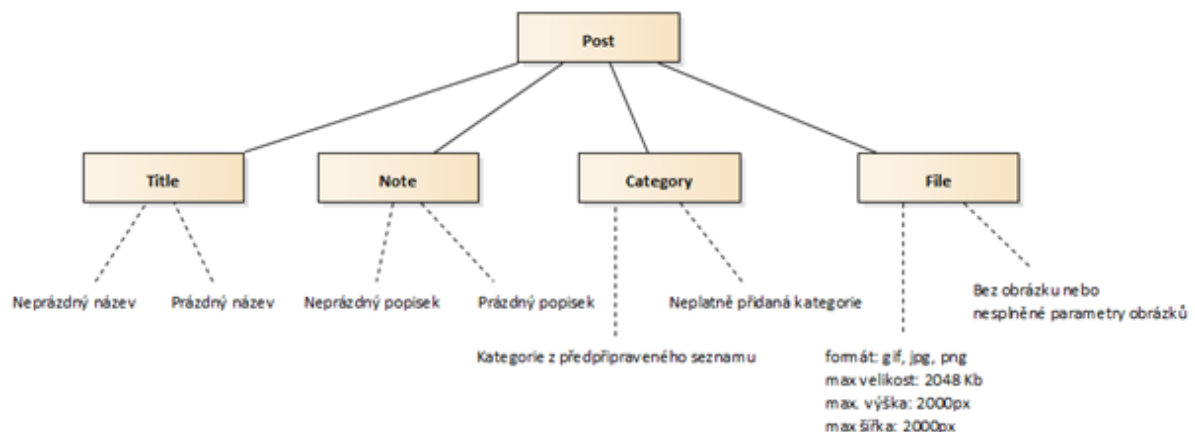
Pairwise test casey

case	Username	Password	pairings
1	Username je v DB	Heslo odpovídá	1
2	Username je v DB	Heslo neodpovídá	1
3	Username není v DB	Heslo odpovídá	1
4	Username není v DB	Heslo neodpovídá	1

Formulář pro vytvoření příspěvku

V tomto případě můžeme nalézt mezní podmínku u nepovinného pole pro nahrání obrázku. Vzniklé mezní podmínky by byly maximální velikost obrázku a její maximální výška a šířka.

Vstupy podle EC pro vytvoření příspěvku



Pairwise test casey

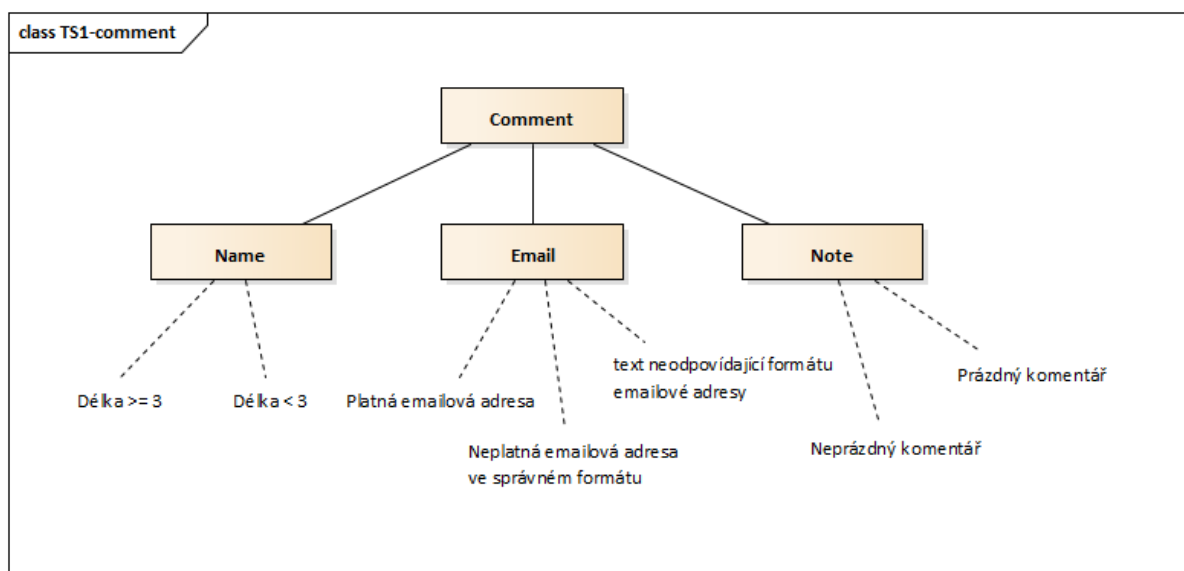
case	Title	Note	Category	File	pairings
1	neprazdny nazev	neprazdny popis	kategorie ze seznamu	spravny format a rozmery	6
2	neprazdny nazev	prazdny popis	neplatna kategorie	neplatny format a rozmery	6
3	prazdny nazev	neprazdny popis	neplatna kategorie	spravny format a rozmery	5
4	prazdny nazev	prazdny popis	kategorie ze seznamu	neplatny format a rozmery	5
5	~neprazdny nazev	neprazdny popis	~kategorie ze seznamu	neplatny format a rozmery	1
6	~neprazdny nazev	prazdny popis	~neplatna kategorie	spravny format a rozmery	1

Přidat komentář

Add Comment

Mezní podmínkou by zde byla opět délka přihlašovacího jména. Za objevenou chybu by se pak dalo považovat, že z hlediska programu není v tomto případě řešena maximální délka emailu a textu v poli write something. Zároveň by se dalo považovat za chybu vyplňování jména a emailu v místě, kam se nedá dostat bez přihlášení.

Vstupy podle EC pro přidání komentáře

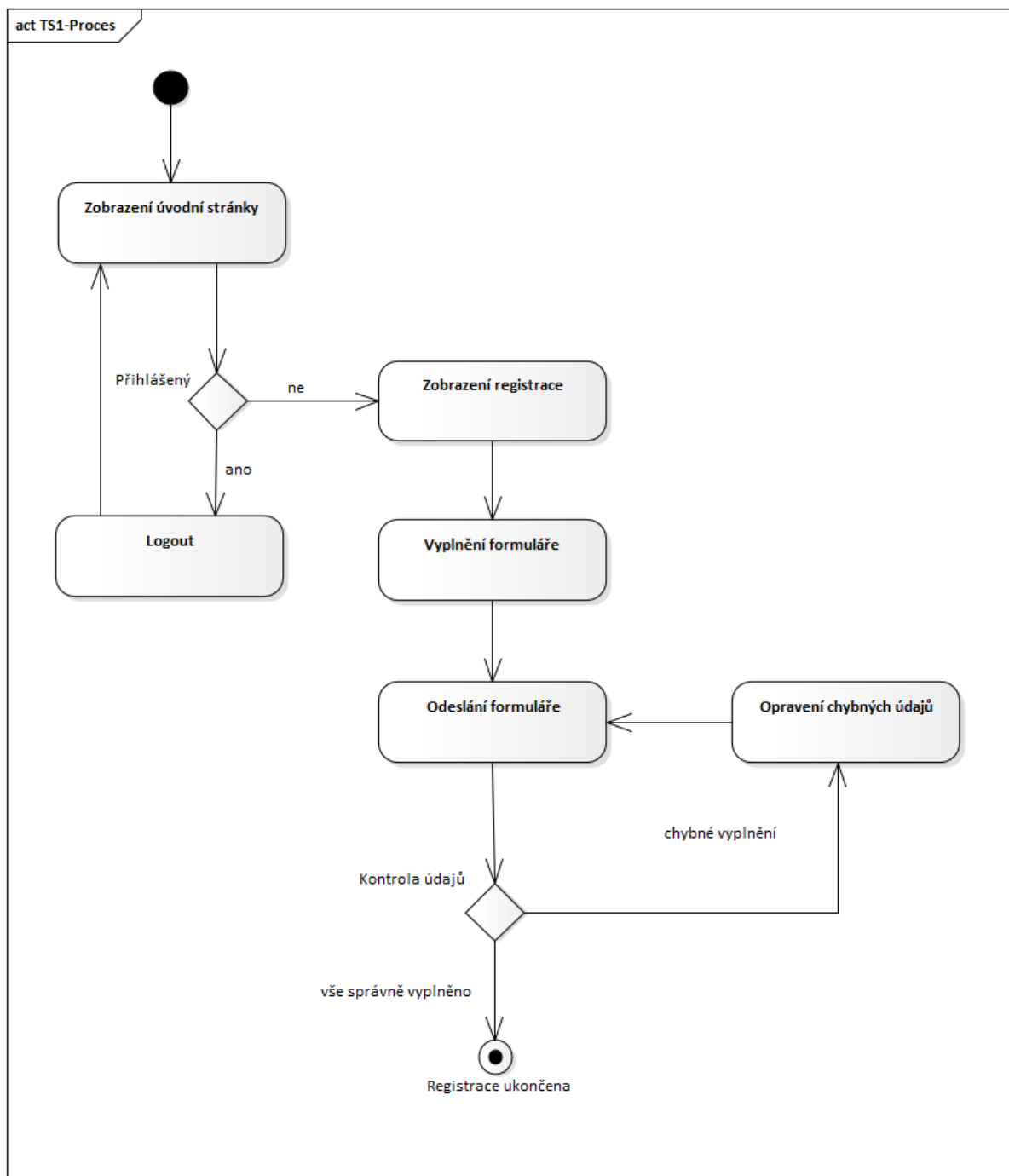


Pairwise test casey

case	Name	Email	Note	pairings
1	delka>=3	platny email	prazdny komentar	3
2	delka < 3	platny email	neprazdny komentar	3
3	delka>=3	neplatny email	neprazdny komentar	3
4	delka < 3	neplatny email	prazdny komentar	3
5	delka>=3	spatny format	prazdny komentar	2
6	delka < 3	spatny format	neprazdny komentar	2

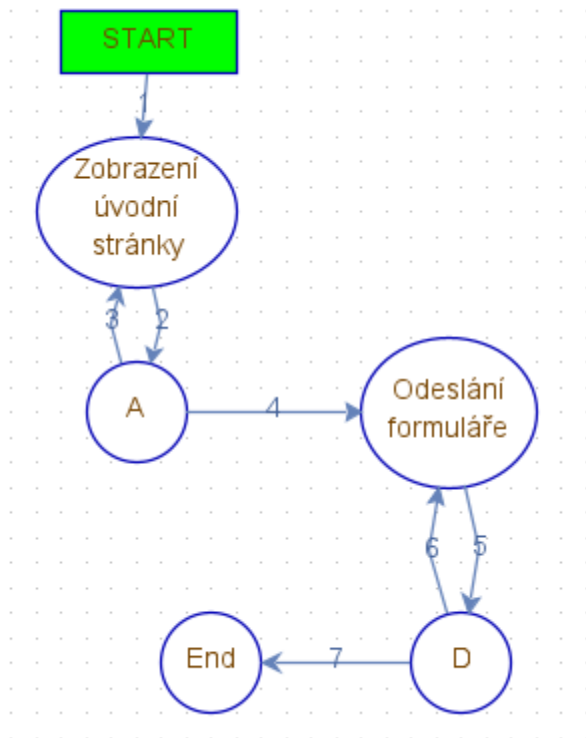
Testy průchodů

Proces registrace



Převedení do orientovaného grafu

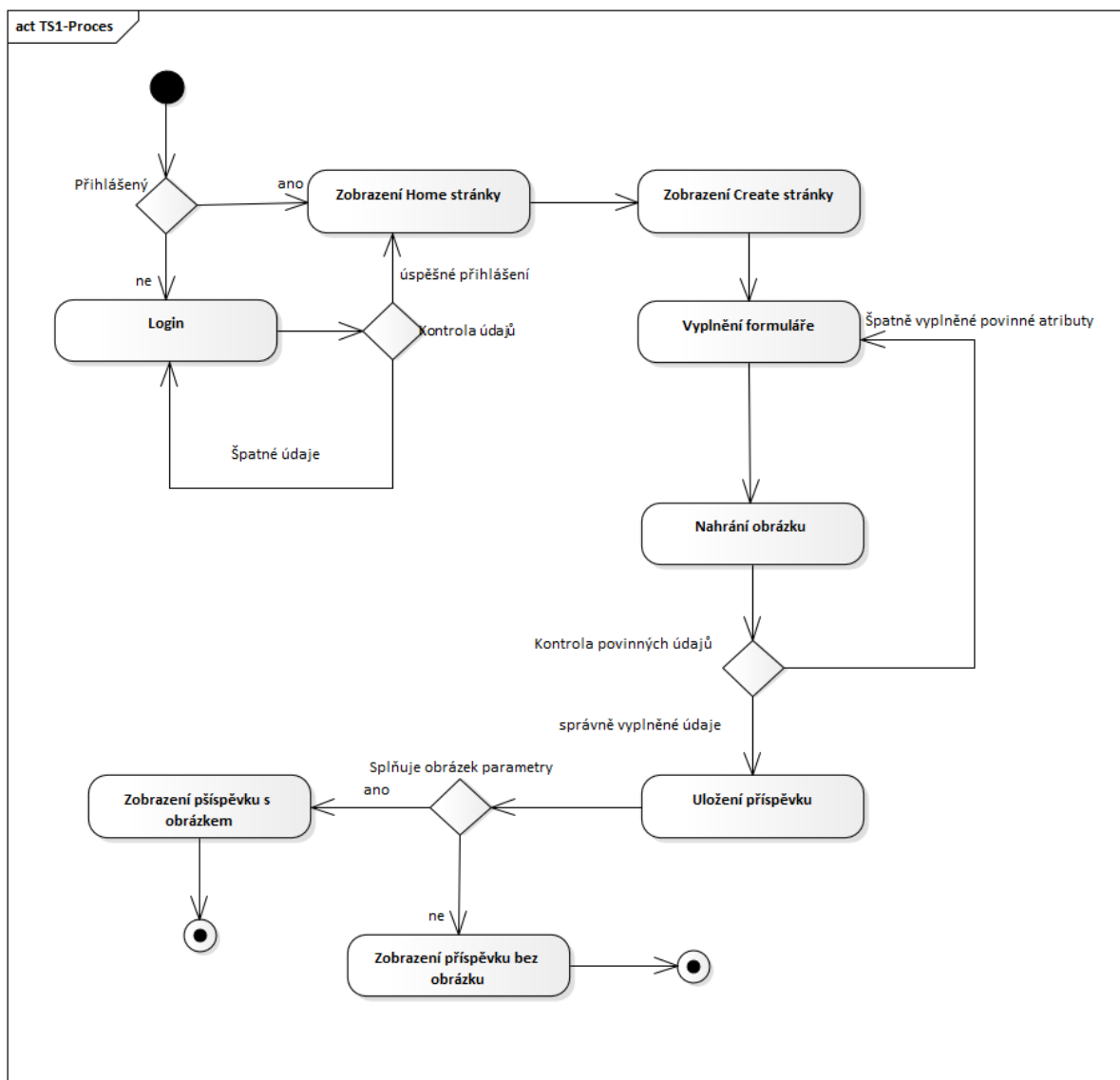
Z důvodu zvýšení přehlednosti a lepšímu odpovídání přechodů, jsou zde vyznačeny i nerozhodovací body (zobrazení úvodní stránky a odeslání formuláře). Víme, že výsledná testovací sekvence, tak je o 2 kroky delší, než by musela nezbytně být, ovšem v praxi to nic nemění ani na délce testu ani na množství testů.



Vygenerované testovací sekvence s TDL = 2

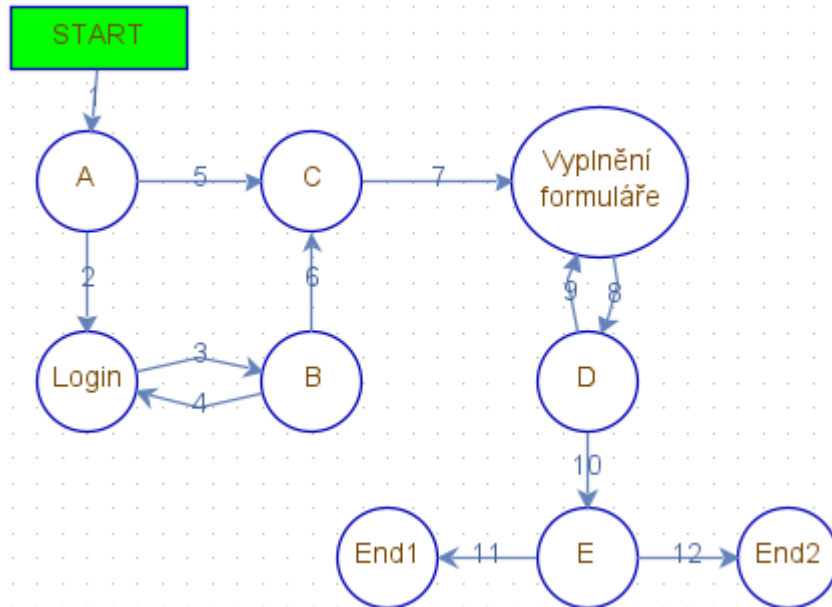
0; 1 - 2 - 3 - 2 - 4 - 5 - 6 - 5 - 7

Proces přidání příspěvku



Převedení do orientovaného grafu

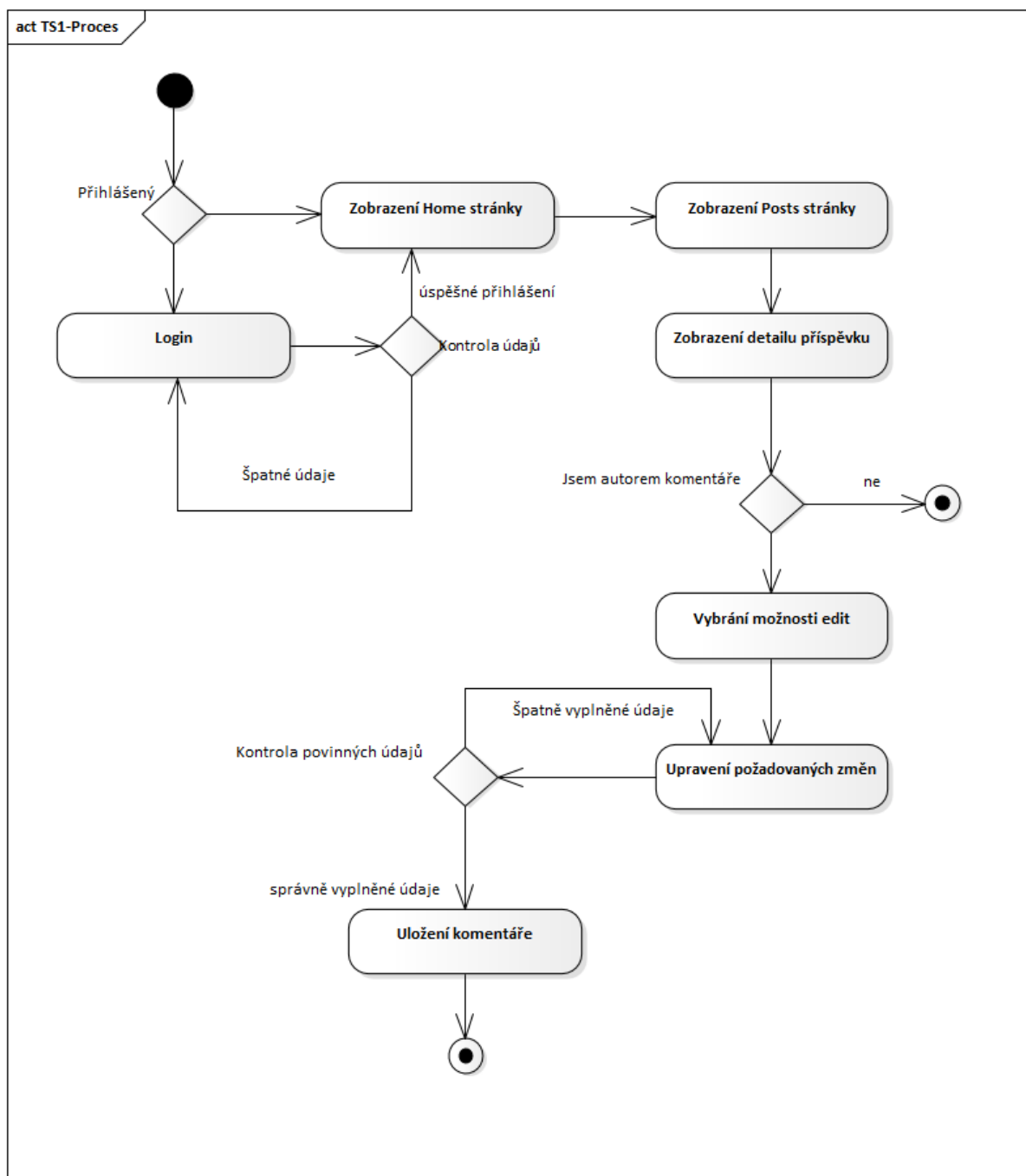
Opět platí již výše zmíněné, že jsou zde přidány 2 nerozhodovací body (Login a vyplnění formuláře), které ovšem nijak nemění množství vytvořených testovacích sekvencí.



Vygenerované testovací sekvence s TDL = 2

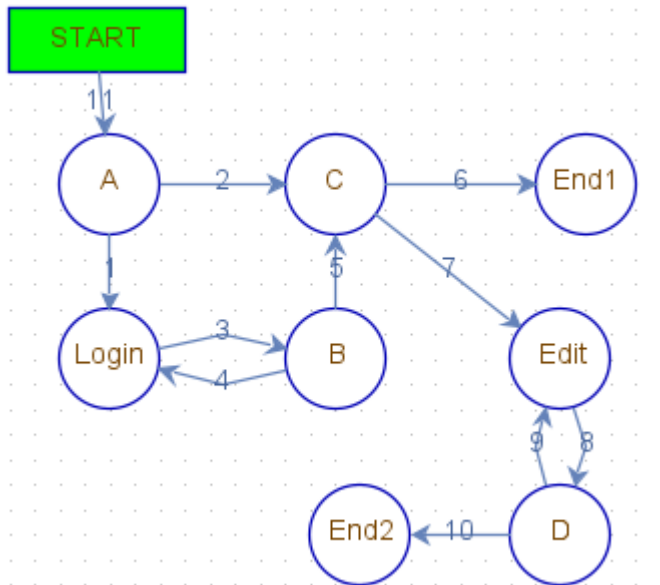
- 0; 1 - 2 - 3 - 4 - 3 - 6 - 7 - 8 - 9 - 8 - 10 - 11
- 1; 1 - 5 - 7 - 8 - 9 - 8 - 10 - 11
- 2; 1 - 5 - 7 - 8 - 10 - 12

Proces úpravy příspěvku



Převedení do orientovaného grafu

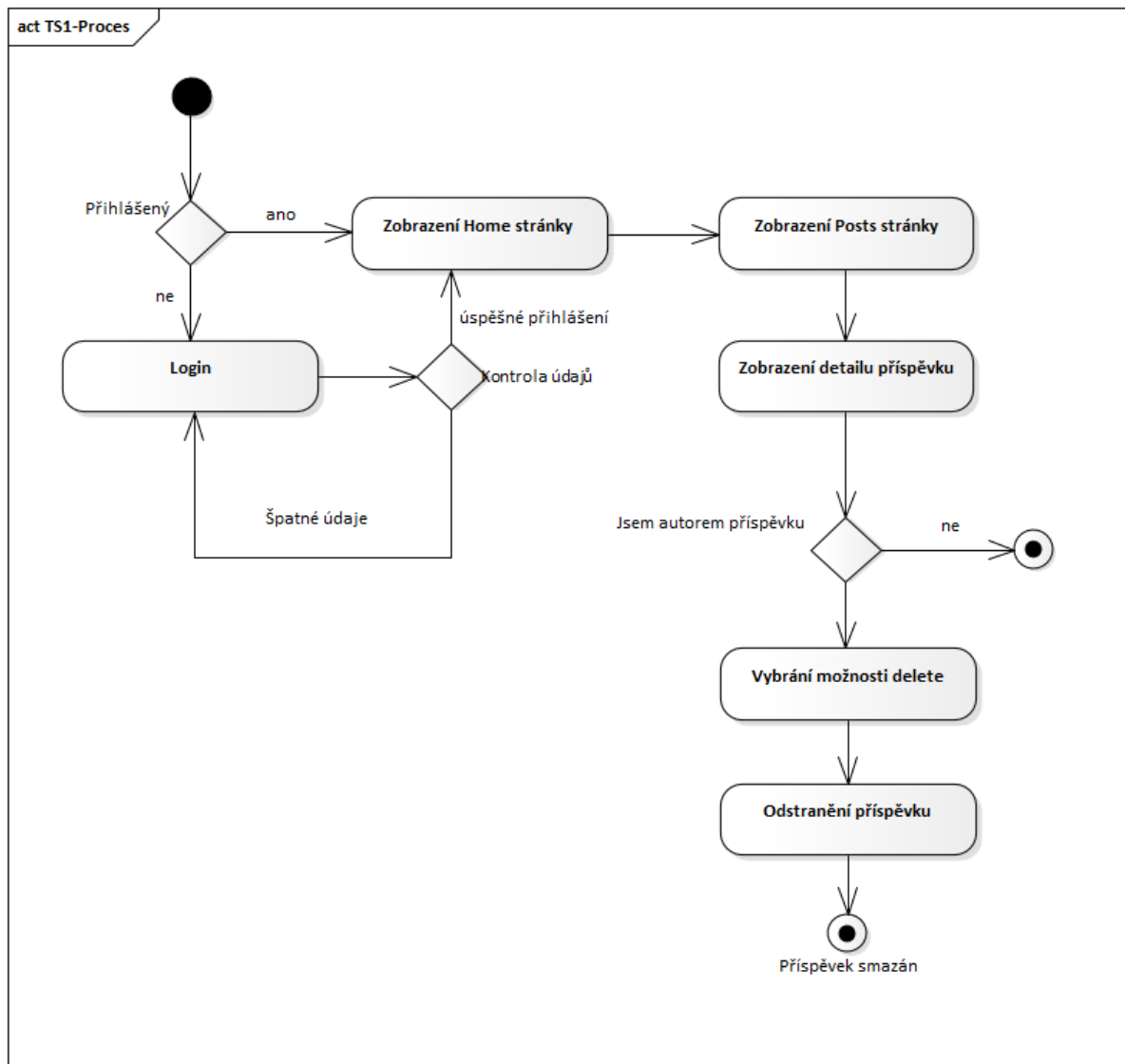
Opět platí již výše zmíněné, že jsou zde přidány 2 nerozhodovací body (Login a edit formuláře), které ovšem nijak nemění množství vytvořených testovacích sekvencí.



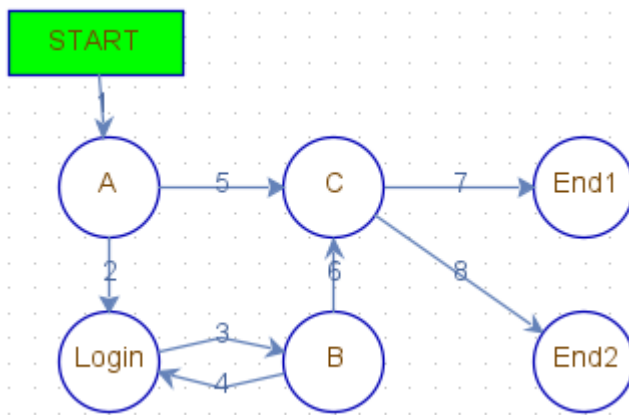
Vygenerované testovací sekvence s TDL = 2

- 0; 11 - 1 - 3 - 4 - 3 - 5 - 6
- 1; 11 - 2 - 6
- 2; 11 - 2 - 7 - 8 - 9 - 8 - 10
- 3; 11 - 1 - 3 - 5 - 7 - 8 - 9 - 8 - 10

Proces odstranění příspěvku



Převedení do orientovaného grafu



Vygenerované testovací sekvence s TDL = 2

- 0; 1 - 2 - 3 - 4 - 3 - 6 - 7
- 1; 1 - 5 - 7
- 2; 1 - 5 - 8
- 3; 1 - 2 - 3 - 6 - 8

Detailní testovací scénáře

Test case 1

Parametr	Obsah
ID testu	TS1-MP-01
Název testu	Test registrace
Hloubka detailu	Nízká
Shrnutí testu	Zadání neplatných údajů, negativní průchod; Zadání platných údajů, pozitivní průchod
Popis testu	Uživatel se pokusí registrovat do systému, pomocí registračního formuláře. Nejprve vyplní údaje nesplňující požadované podmínky. Po neúspěšné registraci chybná pole opraví a otestuje registraci úspěšnou.
Vstupní podmínky	-
Testovací data	1. registrační neúspěšný pokus: name = "Cu", email = "dfafsdf@", username = "ctiměřice24", password = "ahoj1", passwordAgain = "neahoj2" 2. registrační úspěšný pokus: name = "Olesinkka", email = "Chereole1@espargadoj.org", username = "Chereolenlka", password = "TS1ahoj!", passwordAgain = "TS1ahoj!"
Očekávaný výsledek	V první půlce testu se nepodaří registraci odeslat (neprojde přes validaci), v druhém se podaří uživatele úspěšně registrovat)
Autor	Ondřej Bureš a Olesia Cheremnykh
Priorita	Vysoká
Testovací prostředí	wa.toad.cz

Testovací kroky

Parametr	Obsah
Název kroku testu	Zvolení registrace
Očekávaný krok testu	Zvol z hlavního menu možnost registrace
Očekávaný výsledek kroku testu	Přesměrování na registrační formulář
Název kroku testu	Špatné vyplnění formuláře
Očekávaný krok testu	Vyplň formulář podle 1. pokusu testovacích dat
Očekávaný výsledek kroku testu	formulář bude oznamovat chybová varování
Název kroku testu	Odeslání formuláře
Očekávaný krok testu	Klikni na tlačítko Submit
Očekávaný výsledek kroku testu	Odesílání se nezdaří, k odeslání na server vůbec nedojde
Název kroku testu	Správné vyplnění formuláře
Očekávaný krok testu	Vyplň formulář podle 2. pokusu testovacích dat
Očekávaný výsledek kroku testu	Chybová varování zmizí
Název kroku testu	Odeslání formuláře
Očekávaný krok testu	Klikni na tlačítko Submit
Očekávaný výsledek kroku testu	Formulář se podaří odeslat a dojde k úspěšné registraci

Test case 2

Parametr	Obsah
ID testu	TS1-MP-02
Název testu	Login
Hloubka detailu	Nízká
Shrnutí testu	Zadání neplatných údajů, negativní průchod; Zadání platných údajů, pozitivní průchod
Popis testu	Uživatel se pokusí přihlásit do systému, pomocí přihlašovacího formuláře. Nejprve vyplní údaje nesplňující požadované podmínky. Po neúspěšném přihlášení pole opraví a otestuje úspěšné přihlášení.
Vstupní podmínky	-
Testovací data	1. neúspěšný pokus o přihlášení Username = Chereolenlka, password = "TS1adshoj!" 2. úspěšný pokus o přihlášení Username = Chereolenlka, password = "TS1ahoj!"
Očekávaný výsledek	V první půlce testu se nepodaří přihlásit, v druhém se podaří uživatele úspěšně přihlásit do aplikace
Autor	Ondřej Bureš a Olesia Cheremnykh
Priorita	Vysoká
Testovací prostředí	wa.toad.cz

Testovací kroky

Parametr	Obsah
Název kroku testu	Zvolení přihlášení
Očekávaný krok testu	Zvol z hlavního menu možnost login
Očekávaný výsledek kroku testu	Přesměrování na přihlašovací formulář
Název kroku testu	Špatné vyplnění formuláře
Očekávaný krok testu	Vyplň formulář podle 1. pokusu testovacích dat
Očekávaný výsledek kroku testu	Formulář je vyplněný
Název kroku testu	Odeslání formuláře
Očekávaný krok testu	Klikni na tlačítko Submit
Očekávaný výsledek kroku testu	Formulář se odešle, ale k přihlášení nedojde
Název kroku testu	Správné vyplnění formuláře
Očekávaný krok testu	Vyplň formulář podle 2. pokusu testovacích dat
Očekávaný výsledek kroku testu	Formulář je vyplněný
Název kroku testu	Odeslání formuláře
Očekávaný krok testu	Klikni na tlačítko Submit
Očekávaný výsledek kroku testu	Formulář se podaří odeslat a dojde k úspěšnému přihlášení do aplikace

Implementace testů

Unit testy

Používáme PHPUnit s využitím syntaxe frameworku

Unit testy jsme realizovali pomocí testovacího frameworku PHPUnit. Primárně jsme se pak zaměřovali na metody a funkce modelů jednotlivých stránek. Konkrétně pak pro stránky Category, Comment, Post a User. Dále jsme udělali unit testy i pro controller Welcome stránky frameworku [CodeIgniter](#).

Modely jsou pak testovány 2 způsoby. Prvním jsou unit testy kontrolující návratové hodnoty z databáze. Druhý způsob pak pro odstínění databáze využívá mockování a testuje pouze funkce samotné. Pojmenování testů vychází ze stránek, na které jsou testy cílené.

Category_Model_test

Zde testujeme, jestli metoda `get_categories` opravdu vrací všechny kategorie příspěvků a jestli metoda `get_category` na základě id, vrátí správnou kategorii příspěvků.

Category_model_mocking_db_test

Zde za využití mockované DB a nastavení chování, které bychom od databáze očekávali, testujeme nejprve metodu `get_categories` a správné množství zavolání funkcí, které mají být zavolány, následně testujeme, funkčnost metody `get_categories` bez komunikace z DB. Druhý test pak obdobně kontroluje metodu `get_category` pro nahrání pouze jedné konkrétní hodnoty.

Post_Model_test

Zde testujeme, nejprve metodu `get_posts`, a testujeme jestli při nezadání slugu pro URL vrátí všechny posty uložené v naší databázi. Dále testujeme, jestli se při poslání slugu, tento slug přidá do URL. Nakonec testujeme zda se podaří vytvořit nový post.

Post_model_mocking_db_test

Obdobně jako u `category_model_mocking_db_test` vytváříme mock databáze a testujeme metody `get_posts` a následně metodu `create_post` pro úspěšné i neúspěšné vytvoření příspěvku.

User_Model_test

Zde testujeme nejprve, zda při platné registraci dojde k úspěšnému insertu. Dále pak testujeme, jestli se při přihlášení uživatele opravdu přihlásíme ke správnému účtu a pokud při napsání špatných parametrů neodje k úspěšnému přihlášení. Poté testujeme vyhodnocení existence a neexistence emailu a uživatelského jména v databázi.

User_model_mocking_db_test

Obdobně jako u `category_model_mocking_db_test` vytváříme mock databáze a testujeme metodu `get_users`.

Comment_model_mocking_db_test

Obdobně jako u `category_model_mocking_db_test` vytváříme mock databáze a testujeme metodu `get_comments` podle `post_id`.

Welcome_test

Zde testujeme stránku frameworku Codeignite. Testujeme, že nám GET request vrátí požadovanou stránku. Dále testujeme vrácení HTTP stavového kódu 404, při GET požadavku na neplatné URL. Poslední test je pak spíše na ukázkou, je to vzorový test pro testování s PHPUnit a testuje správné nastavení konfigurační cesty ke složce `application`.

Integrační testy

Všechny procesní a integrační testy jsou řešené za využití frameworku Selenia. Testy pokrývají prakticky všechny funkcionality, které aplikace nabízí. Prakticky ve všech souborech se z důvodu nastavení systému musíme přihlásit. Proto přihlášení není dále zmiňováno.

Testování registrace

V testovacím souboru `testRegister` testujeme vytváření nového účtu pro používání aplikace. Testujeme kompletní průchod, kdy nejprve se snažíme registrovat pomocí neplatných dat a po odmítnutí testujeme již úspěšně data platná.

Testování loginu

V testovacím souboru `testLogin` testujeme přihlášení účtu do aplikace. Testujeme kompletní průchod, kdy nejprve se snažíme přihlásit pomocí neplatného hesla a po odmítnutí přepisujeme přihlašovací údaje to platné podoby.

Testování CRUD pro příspěvky

V testovacím souboru `testPost` testujeme postupně vytváření nového příspěvku, kde nejprve vyplňujeme neplatné údaje. Po odmítnutí je upravujeme na platné a nahráváme nový příspěvek. Tento příspěvek poté hledáme, a následně ho upravujeme. Po upravení testujeme i smazání, a že se nám ho nepodaří dohledat.

Testování napsání komentářů

V testovacím souboru `testComments`, testujeme okomentování vybraného příspěvku.

Testování vyhledání příspěvku podle kategorie a test funkčnosti filtrace

V testovacím souboru `TestCategory` testujeme nejprve průchod a filtrování příspěvků pro všechny kategorie. Následně v dalším testu vytváříme nový příspěvek, který pak pomocí filtrace kategorií hledáme.

Testování správného nastavení LearnMore odkazů

V testovacím souboru TestLearnMoreLinks testujeme postupně všechny odkazy, jestli jsou funkční a odkazují na stránky, které mají.

Testování nočního a denního režimu

Testovací soubor testMods prochází všemi částmi aplikace a na všech testuje, zda funguje přepínání mezi denním a nočním režimem. Při vizuálním sledování průběhu testu, jsme zjistili, že aplikační informativní hláška o úspěšném přihlášení nepřebírá vlastnosti nočního režimu.

Testování stránkování a počtu článků

V testovacím souboru testPaging, testujeme, zda je možné využívat stránkování a přecházet na poslední stránku nejstarších postů i na stránku s posty nejnovějšími. Současně testujeme, že celkový počet příspěvků.