

# **TATWIN – PROJEKT**

## **TESTOWANIE STRONY: WORLDOFTANKS.EU/PL/**

**AUTOR: MATEUSZ GŁOWACKI, U-14665**

**POLITECHNIKA KOSZALIŃSKA 2019/2020**

Spis treści:

1. Wstęp
2. Opis poszczególnych plików dołączonych do projektu
3. Opis przypadków testowych wraz z uwagami

### 1. Wstęp

Projekt zakładał wykonanie testów zgodnie z przekazanymi wytycznymi. Na potrzeby testów zostało utworzone jedno konto do gry – nie zawiera ono niczego, każdy może utworzyć darmowe konto.

### 2. Opis poszczególnych plików dołączonych do projektu

W projekcie zostały utworzone dwa pakiety: pakiet poms oraz pakiet project.

Pakiet poms zawiera pliki odpowiedzialne za Page object model – pliki zawierające lokalizatory oraz czynności wykonywane na tych lokalizatorach.

Plik Login.java zawiera lokalizatory strony logowania oraz informacji o użytkowniku – zarządzanie kontem.

Plik MainWebsite.java zawiera lokalizatory strony głównej

Plik Register.java zawiera lokalizatory strony służącej do utworzenia konta.

Plik Tankpedia.java zawiera lokalizatory strony „czołgopedii”

Pakiet project zawiera pliki z testami. Podzielone one zostały według punktów podanych w specyfikacji za wyjątkiem punktów 2 oraz 3 – zostały one uwzględnione w ramach jednego pliku ponieważ podpunkt 2 jest wymagany aby zrealizować podpunkt 3.

Plik WoTLogin.java zawiera testy modułu logowania oraz pobierania danych użytkownika

Plik WoTRegister3.java zawiera testy modułu rejestracji użytkownika

Plik WoTTank.java zawiera testy modułu wyszukiwania czołgów na stronie czołgopedii.

Folder test-output w katalogu bazowym zawiera pliki:

wotlogin.pdf, wotregister.pdf, wottank.pdf. Pliki te zawierają raporty czy przypadki testowe wykonują się poprawnie.

### 3. Opis przypadków testowych wraz z uwagami:

Ważne jest aby w każdym z plików testowych ustawić własną ścieżkę do webdrivera.

Opis zacznę od pliku rejestracji użytkownika.

- Test pierwszy sprawdza czy pomyślnie uda nam się przejść z strony głównej do strony logowania.
- Test drugi zaznacza nam pole akceptacji umowy użytkownika – bez tego nie podejmiemy nawet próby rejestracji użytkownika.
- Test trzeci zakłada podanie nieprawidłowego formatu adresu email a następnie odczytanie alertu.
- Test czwarty zakłada podanie już wykorzystanego adresu email oraz odczytanie alertu.
- Test piąty zakłada podanie nieprawidłowej nazwy użytkownika poprzez umieszczenie spacji oraz odczytanie alertu.
- Test szósty zakłada podanie nieprawidłowej nazwy użytkownika poprzez umieszczenie znaków specjalnych oraz polskich oraz odczytanie alertu.
- Test siódmy zakłada podanie zbyt krótkiej nazwy użytkownika oraz odczytanie alertu.
- Test ósmy zakłada podanie już zajętej nazwy użytkownika oraz odczytanie alertu.
- Test dziewiąty zakłada wpisanie hasła z znakiem spacji – co jest niedozwolone oraz odczytanie alertu.
- Test dziesiąty zakłada wpisanie hasła z znakami specjalnymi oraz odczytanie alertu.
- Test jedenasty zakłada wpisanie zbyt krótkiego hasła oraz odczytanie alertu.
- Test dwunasty zakłada wpisanie niezgodnego hasła w pole powtórzenia hasła oraz odczytanie alertu
- Test trzynasty zakłada wpisanie wszystkich poprawnych danych LECZ bez kliknięcia przycisku rejestruj, zgodnie z wymaganiami tak aby nie „zaśmiecać” serwera.

#### Uwagi:

Może się zdarzyć że testy będą wykonywać się na tyle szybko że komunikaty nie zdążą się zmienić. Starałem się temu zapobiec stosując różne zmienne do każdego z możliwych komunikatów. Dodatkowo losowo może wyskoczyć że jeden z elementów jest nieklikalny, gdzie w rzeczywistości jest on cały czas widoczny oraz klikalny – nie wiem czym jest spowodowana losowość tego zdarzenia.

Testy w tym pliku posiadają priorytetyzację ponieważ jest ona niezbędna – najważniejszy priorytet ma uruchomienie strony rejestracji z strony głównej. Następnie priorytet ma zaznaczenie umowy użytkownika aby w następnych testach nie zmieniało się to cały czas z odklikniętego na kliknięte. Następne testy posiadają priority 3 – wykonują się niezależnie, mogą się wykonywać w dowolnej kolejności. Ostatni test ma priority 4 ponieważ w przypadku gdybyśmy chcieli sprawdzić funkcję POST na serwer wystarczy jedynie odkomentować kliknięcie przycisku rejestracji.

Plik logowania:

- Test pierwszy sprawdza czy pomyślnie przejdziemy z strony głównej na stronę logowania.
- Test drugi sprawdza czy uda nam się zalogować bez podania hasła, a jeżeli nie odczytuje komunikat błędu.
- Test trzeci sprawdza czy uda nam się zalogować bez podania adresu email, a jeżeli nie odczytuje komunikat błędu.
- Test czwarty sprawdza czy uda nam się zalogować błędnymi danymi, a jeżeli nie odczytuje komunikat błędu.
- Test piąty sprawdza czy uda nam się zalogować poprzez podanie prawdziwych danych.
- Test szósty sprawdza czy uda nam się przejść do podstrony zarządzania kontem.
- Test siódmy odczytuje dane na stronie zarządzania kontem a następnie wypisuje je w okienku konsoli.

Uwagi:

Test pierwszy używa priorytetu 1, ponieważ jest to przejście na stronę logowania.

Testy zabezpieczeń używają priorytetu 2 oraz mogą być wykonane dowolnie.

Test piąty używa priorytetu trzeciego, ponieważ nie chciałem tworzyć osobnego pliku z testem tego modułu, a jest on niezbędny do następnych testów.

Test szósty używa priorytetu czwartego ponieważ jest to niezbędne – bez zalogowania nie wyświetli nam się okienko użytkownika. Dodatkowo w pojedynczych sytuacjach może zdarzyć się że pomimo kliknięcia na nazwę użytkownika aplet z dalszym odnośnikiem nie wysuwa się – jest to najprawdopodobniej błąd strony.

Test siódmy używa priorytetu piątego ponieważ niemożliwe jest wyświetlenie tych danych bez zalogowania oraz przejścia do tej podstrony.

Plik czołgopedii:

- Pierwszy test zaznacza jeden filtr z pierwszego typu oraz zlicza ilość pasujących czołgów.
- Drugi test zaznacza jeden filtr z drugiego typu oraz zlicza ilość pasujących czołgów.
- Trzeci test zaznacza jeden filtr z trzeciego typu oraz zlicza ilość pasujących czołgów.
- Czwarty test zaznacza jeden filtr z czwartego typu oraz zlicza ilość pasujących czołgów.
- Piąty test zaznacza dwa filtry z pierwszego typu oraz zlicza ilość pasujących czołgów.
- Szósty test zaznacza dwa filtry z drugiego typu oraz zlicza ilość pasujących czołgów.
- Siódmy test zaznacza dwa filtry z trzeciego typu oraz zlicza ilość pasujących czołgów.
- Ósmy test zaznacza dwa filtry z czwartego typu oraz zlicza ilość pasujących czołgów.
- Dziewiąty test zaznacza po jednym filtrze z pierwszej oraz drugiej kategorii oraz zlicza ilość pasujących czołgów.
- Dziesiąty test zaznacza po jednym filtrze z pierwszej oraz trzeciej kategorii oraz zlicza ilość pasujących czołgów.
- Jedenasty test zaznacza po jednym filtrze z pierwszej oraz czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Dwunasty test zaznacza po jednym filtrze z drugiej oraz trzeciej kategorii oraz zlicza ilość pasujących czołgów.
- Trzynasty test zaznacza po jednym filtrze z drugiej oraz czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Czternasty test zaznacza po jednym filtrze z trzeciej oraz czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Piętnasty test zaznacza po dwa filtry z pierwszej oraz drugiej kategorii oraz zlicza ilość pasujących czołgów.
- Szesnasty test zaznacza po dwa filtry z pierwszej oraz trzeciej kategorii oraz zlicza ilość pasujących czołgów.
- Siedemnasty test zaznacza po dwa filtry z pierwszej oraz czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Osiemnasty test zaznacza po dwa filtry z drugiej oraz trzeciej kategorii oraz zlicza ilość pasujących czołgów.
- Dziewiętnasty test zaznacza po dwa filtry z drugiej oraz czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Dwudziesty test zaznacza po dwa filtry z trzeciej oraz czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Dwudziesty pierwszy test zaznacza co najmniej jeden filtr z pierwszej, drugiej i czwartej kategorii oraz zlicza ilość pasujących czołgów.
- Dwudziesty drugi test zaznacza co najmniej jeden filtr z każdej kategorii oraz zlicza ilość pasujących czołgów.
- Dwudziesty trzeci test zaznacza kombinację nie istniejącą w chwili obecnej w grze.

Uwagi:

Testy wykonują się całkowicie niezależnie

Wybór filtrów można dowolnie modyfikować na potrzeby testu, wystarczy jedynie zmieniać odwołania do pliku Tankpedia.java. Po uruchomieniu testów kilka razy okazało się iż niektóre kombinacje pokazują wartość zero, co po weryfikacji jest poprawne, więc test ostatni jest tak naprawę nie potrzebny. Dodatkowo w ostatnim teście użycie asercji oczekuj:0 daje niepowodzenie mimo otrzymania pożądanej wartości. Zostało to udokumentowane w komentarzu w linii 549 kodu pliku WoTTank.java