# Testy jednostkowe klasa User - autentykacja

**1. Cel testów**

Celem testów było sprawdzenie poprawności implementacji metody signIn w klasie User, która odpowiada za logowanie użytkownika w aplikacji. Przetestowano trzy scenariusze:

1. Poprawne logowanie.
2. Nieudane logowanie z powodu błędnych danych logowania.
3. Obsługa wyjątku w przypadku błędu bazy danych.

**2. Konfiguracja środowiska**

* **Język programowania:** PHP 8.2.12
* **Framework testowy:** PHPUnit 11.5.2
* **Mocki:** Do testów jednostkowych wykorzystano mocki klas PDO i PDOStatement.
* **Środowisko uruchomieniowe:** XAMPP

**3. Testy i wyniki**

**Test 1: testSignIn\_SuccessfulLogin**

* **Opis:** Testuje poprawne logowanie użytkownika przy użyciu poprawnych danych uwierzytelniających.
* **Przygotowanie:**
  + Mockowanie klasy PDOStatement, aby metoda fetch zwróciła symulowane dane użytkownika.
  + Mockowanie klasy PDO, aby metoda prepare zwracała odpowiedniego mocka.
* **Oczekiwania:**
  + Atrybut userData w obiekcie klasy User powinien zawierać dane użytkownika.
  + W sesji powinien pojawić się klucz user.
* **Wynik:** Test zakończony sukcesem. Wszystkie oczekiwania spełnione.

**Test 2: testSignIn\_FailedLogin**

* **Opis:** Testuje przypadek nieudanego logowania z powodu podania błędnych danych uwierzytelniających.
* **Przygotowanie:**
  + Mockowanie klasy PDOStatement, aby metoda fetch zwróciła false.
  + Mockowanie klasy PDO, aby metoda prepare zwracała odpowiedniego mocka.
* **Oczekiwania:**
  + Atrybut userData w obiekcie klasy User powinien być null.
  + Sesja nie powinna zawierać klucza user.
* **Wynik:** Test zakończony sukcesem. Wszystkie oczekiwania spełnione.

**Test 3: testSignIn\_ThrowsExceptionOnDatabaseError**

* **Opis:** Testuje przypadek, gdy logowanie kończy się błędem z powodu problemów z bazą danych.
* **Przygotowanie:**
  + Mockowanie klasy PDO, aby metoda prepare rzucała wyjątek typu PDOException.
* **Oczekiwania:**
  + Metoda signIn powinna rzucić wyjątek typu Exception.
* **Wynik:** Test zakończony sukcesem. Wyjątek został poprawnie obsłużony.

**4. Statystyki testów**

* **Łączna liczba testów:** 3
* **Łączna liczba asercji:** 5
* **Czas wykonania:** 00:02.572 sekundy
* **Zużycie pamięci:** 8 MB
* **Status:** Wszystkie testy zakończone sukcesem.

**5. Wnioski**

Wszystkie zaplanowane scenariusze zostały pomyślnie przetestowane. Metoda signIn działa poprawnie w przypadkach:

* Prawidłowych danych logowania.
* Błędnych danych logowania.
* Obsługi sytuacji wyjątkowych w bazie danych.

Kod spełnia wymagania funkcjonalne dla testowanych przypadków, a środowisko testowe działa zgodnie z oczekiwaniami. Warto kontynuować testy jednostkowe dla innych metod klasy User oraz pozostałych klas w aplikacji.

Wyniki konsola:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Kod testów:

<?php

*require\_once* \_\_DIR\_\_ . '/../dataBase/dataBaseConnection.php';

*require\_once* \_\_DIR\_\_ . '/../users/User.php';

*use* PHPUnit\Framework\TestCase;

*class* AuthTest *extends* TestCase

{

*private* $user;

*private* $pdo;

*protected* *function* **setUp**(): void

    {

*$this*->pdo = *$this*->**createMock**(PDO::*class*); *// Mockowanie PDO*

*$this*->user = *new* User(*$this*->pdo); *// Tworzenie instancji klasy User*

    }

*// logowanie z sukcesem*

*public* *function* **testSignIn\_SuccessfulLogin**()

    {

        $mockData = [

            'id' => 1,

            'login' => 'user1',

            'password' => 'hashed\_password',

            'rola' => 'admin'

        ];

        $stmt = *$this*->**createMock**(PDOStatement::*class*);

        $stmt->**method**('fetch')->**willReturn**($mockData);

*$this*->pdo->**method**('prepare')->**willReturn**($stmt);

*$this*->user->**signIn**('user1', 'password');

*$this*->**assertEquals**($mockData, *$this*->user->userData);

*$this*->**assertTrue**(isset($\_SESSION['user']));

    }

*// logowanie z błędem*

*public* *function* **testSignIn\_FailedLogin**()

    {

        $stmt = *$this*->**createMock**(PDOStatement::*class*);

        $stmt->**method**('fetch')->**willReturn**(false);

*$this*->pdo->**method**('prepare')->**willReturn**($stmt);

*$this*->user->**signIn**('user1', 'wrongpassword');

*$this*->**assertNull**(*$this*->user->userData);

*$this*->**assertNull**($\_SESSION['user'] ?? null);

    }

*// obsługa błędu przy próbie zalogowaniu*

*public* *function* **testSignIn\_ThrowsExceptionOnDatabaseError**()

    {

*$this*->**expectException**(Exception::*class*);

*$this*->pdo->**method**('prepare')->**willThrowException**(*new* PDOException('Database error'));

*$this*->user->**signIn**('user1', 'password');

    }

}

# Testy jednostkowe klasa Manager – główne funkcjonalności aplikacji

**1. Cel testów**

Celem testów było sprawdzenie poprawności działania metod klasy Manager. Przetestowano zarówno funkcjonalności zarządzania pracownikami i pojazdami, jak i obsługę wyjątków oraz generowanie raportów.

**2. Konfiguracja środowiska**

* **Język programowania:** PHP 8.2.12
* **Framework testowy:** PHPUnit 11.5.2
* **Mocki:** W testach jednostkowych wykorzystano mocki klasy Manager z użyciem metody getMockBuilder.
* **Środowisko uruchomieniowe:** XAMPP

**3. Testy i wyniki**

**Testy metod klasy Manager:**

1. **testGetAllEmployers**
   * **Opis:** Pobranie listy wszystkich pracowników.
   * **Oczekiwania:** Lista pracowników zawiera dwa elementy, a pierwszy z nich to "Jan Kowalski".
   * **Wynik:** Sukces.
2. **testAddEmployer**
   * **Opis:** Dodanie nowego pracownika.
   * **Oczekiwania:** Metoda addEmployer zwraca true po poprawnym dodaniu pracownika.
   * **Wynik:** Sukces.
3. **testDeleteEmployer**
   * **Opis:** Usunięcie pracownika.
   * **Oczekiwania:** Metoda deleteEmployer zwraca true po poprawnym usunięciu pracownika.
   * **Wynik:** Sukces.
4. **testGetAssigments**
   * **Opis:** Pobranie listy przydziałów pojazdów.
   * **Oczekiwania:** Lista zawiera dwa elementy, a pierwszy przydział dotyczy Toyoty Corolli.
   * **Wynik:** Sukces.
5. **testDeleteVehicleThrowsException**
   * **Opis:** Obsługa wyjątku podczas usuwania pojazdu w użyciu.
   * **Oczekiwania:** Metoda deleteVehicle rzuca wyjątek z odpowiednim komunikatem.
   * **Wynik:** Sukces.
6. **testGetVehicleById**
   * **Opis:** Pobranie pojazdu po ID.
   * **Oczekiwania:** Pobrany pojazd to Toyota Corolla.
   * **Wynik:** Sukces.
7. **testAddVehicle**
   * **Opis:** Wprowadzenie nowego pojazdu do floty.
   * **Oczekiwania:** Metoda addVehicle zwraca true po poprawnym dodaniu pojazdu.
   * **Wynik:** Sukces.
8. **testEditEmployer**
   * **Opis:** Edycja danych pracownika.
   * **Oczekiwania:** Metoda editEmployer zwraca true po poprawnej edycji danych.
   * **Wynik:** Sukces.
9. **testAcceptRequest**
   * **Opis:** Akceptacja zapytania o przydział pojazdu.
   * **Oczekiwania:** Metoda accpetRequest jest wywoływana poprawnie dla podanego ID.
   * **Wynik:** Sukces.
10. **testDeclineRequest**
    * **Opis:** Odrzucenie zapytania o przydział pojazdu.
    * **Oczekiwania:** Metoda declineRequest jest wywoływana poprawnie dla podanego ID.
    * **Wynik:** Sukces.
11. **testDeleteAsigment**
    * **Opis:** Usunięcie przydziału pojazdu.
    * **Oczekiwania:** Metoda deleteAsigment jest wywoływana poprawnie dla podanego ID.
    * **Wynik:** Sukces.
12. **testGenerateReport**
    * **Opis:** Generowanie raportu.
    * **Oczekiwania:** Wygenerowany raport zawiera dwa elementy, z których pierwszy odnosi się do Toyoty Corolli z 5 wypożyczeniami.
    * **Wynik:** Sukces.

**4. Statystyki testów**

* **Łączna liczba testów:** 12
* **Łączna liczba asercji:** 12
* **Czas wykonania:** 00:00.388 sekundy
* **Zużycie pamięci:** 8 MB
* **Status:** Wszystkie testy zakończone sukcesem.

**5. Wnioski**

Wszystkie przetestowane metody klasy Manager działają poprawnie w różnych scenariuszach, w tym w obsłudze wyjątków i zarządzaniu zasobami aplikacji. Kod spełnia założenia funkcjonalne i jest zgodny z wymaganiami. Testy potwierdzają stabilność implementacji oraz jej zgodność z założeniami projektowymi.

Rekomenduje się przeprowadzenie testów integracyjnych oraz dodatkowych testów jednostkowych dla bardziej złożonych scenariuszy.

Konsola wynik:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Kod testów:

<?php

*require\_once* \_\_DIR\_\_ . '/../dataBase/dataBaseConnection.php';

*require\_once* \_\_DIR\_\_ . '/../users/User.php';

*require\_once* \_\_DIR\_\_ . '/../users/Manager.php';

*use* PHPUnit\Framework\TestCase;

*class* ManagerTest *extends* TestCase

{

*// pobranie wszystkich pracowników*

*public* *function* **testGetAllEmployers**()

    {

*// Tworzymy mock klasy Manager*

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['getAllEmployers'])

            ->**getMock**();

*// Zwracamy przykładowe dane z mocka*

        $managerMock->**method**('getAllEmployers')

            ->**willReturn**([

                ['imie' => 'Jan', 'nazwisko' => 'Kowalski'],

                ['imie' => 'Anna', 'nazwisko' => 'Nowak']

            ]);

*// Testowanie metody*

        $result = $managerMock->**getAllEmployers**();

*$this*->**assertCount**(2, $result); *// Sprawdzamy, czy wynik ma 2 elementy*

*$this*->**assertEquals**('Jan', $result[0]['imie']);

    }

*// dodanie pracownika*

*public* *function* **testAddEmployer**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['addEmployer'])

            ->**getMock**();

*// Sprawdzamy, czy metoda `addEmployer` zwróci `true` po poprawnym dodaniu*

        $managerMock->**method**('addEmployer')

            ->**willReturn**(true);

        $result = $managerMock->**addEmployer**('Jan', 'Kowalski', '12345678901', 'jan.kowalski@example.com', 'jankowalski', 'pracownik', 'hashed\_password');

*$this*->**assertTrue**($result); *// Oczekujemy, że wynik będzie `true`*

    }

*// usuwanie pracownika*

*public* *function* **testDeleteEmployer**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['deleteEmployer'])

            ->**getMock**();

*// Ustawiamy, że metoda `deleteEmployer` zwróci `true`*

        $managerMock->**method**('deleteEmployer')

            ->**willReturn**(true);

        $result = $managerMock->**deleteEmployer**(1); *// Przykładowe ID*

*$this*->**assertTrue**($result); *// Oczekujemy, że wynik będzie `true`*

    }

*// pobranie listy przydziałów pojazdów*

*public* *function* **testGetAssigments**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['getAssigments'])

            ->**getMock**();

*// Zwracamy przykładowe dane z mocka*

        $managerMock->**method**('getAssigments')

            ->**willReturn**([

                ['id\_wypozyczenia' => 1, 'marka' => 'Toyota', 'model' => 'Corolla', 'imie' => 'Jan', 'nazwisko' => 'Kowalski'],

                ['id\_wypozyczenia' => 2, 'marka' => 'BMW', 'model' => 'X5', 'imie' => 'Anna', 'nazwisko' => 'Nowak']

            ]);

*// Testowanie metody*

        $result = $managerMock->**getAssigments**();

*$this*->**assertCount**(2, $result); *// Oczekujemy 2 przydziały*

*$this*->**assertEquals**('Toyota', $result[0]['marka']);

    }

*// usuwanie pojazdu - sprawdzamy czy walidacja jest ok*

*public* *function* **testDeleteVehicleThrowsException**()

    {

*$this*->**expectException**(Exception::*class*);

*$this*->**expectExceptionMessage**('Pojazd w użyciu, najpierw zakończ przydział.');

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['deleteVehicle'])

            ->**getMock**();

*// Ustawiamy, że wywołanie tej metody spowoduje wyjątek*

        $managerMock->**method**('deleteVehicle')

            ->**willThrowException**(*new* Exception('Pojazd w użyciu, najpierw zakończ przydział.'));

        $managerMock->**deleteVehicle**(1); *// Przykładowe ID pojazdu*

    }

*// pobranie pojazdu po id*

*public* *function* **testGetVehicleById**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['getVehicleById'])

            ->**getMock**();

*// Zwracamy przykładowe dane z mocka*

        $managerMock->**method**('getVehicleById')

            ->**willReturn**(['id\_pojazdu' => 1, 'marka' => 'Toyota', 'model' => 'Corolla']);

*// Testowanie metody*

        $result = $managerMock->**getVehicleById**(1);

*$this*->**assertEquals**('Toyota', $result['marka']);

    }

*// wprowadzenie pojazdu do floty*

*public* *function* **testAddVehicle**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['addVehicle'])

            ->**getMock**();

*// Sprawdzamy, czy metoda `addVehicle` zwróci `true`*

        $managerMock->**method**('addVehicle')

            ->**willReturn**(true);

        $result = $managerMock->**addVehicle**('Toyota', 'Corolla', 'serwis', 2020, 'dostepny', '2025-12-31');

*$this*->**assertTrue**($result); *// Oczekujemy, że wynik będzie `true`*

    }

*// edycycja danych pracownika*

*public* *function* **testEditEmployer**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['editEmployer'])

            ->**getMock**();

*// Ustawiamy, że metoda `editEmployer` zwróci `true`*

        $managerMock->**method**('editEmployer')

            ->**willReturn**(true);

        $result = $managerMock->**editEmployer**(1, 'Jan', 'Kowalski', 'jan@kowalski.com', '12345678901', 'jankowalski', 'serwisant', 'new\_hashed\_password');

*$this*->**assertTrue**($result); *// Oczekujemy, że wynik będzie `true`*

    }

*// akceptacja zapytania o przydział na pojazd*

*public* *function* **testAcceptRequest**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['accpetRequest'])

            ->**getMock**();

*// Ustawiamy, że metoda `accpetRequest` nie zwróci żadnej wartości (void)*

        $managerMock->**expects**(*$this*->**once**())

            ->**method**('accpetRequest')

            ->**with**(*$this*->**equalTo**(1)); *// Zakładając, że id zapytania to 1*

*// Testowanie, że metoda nie rzuca żadnym błędem*

        $managerMock->**accpetRequest**(1);

    }

*// odrzucenie zapytania o przydział*

*public* *function* **testDeclineRequest**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['declineRequest'])

            ->**getMock**();

*// Ustawiamy, że metoda `declineRequest` nie zwróci żadnej wartości (void)*

        $managerMock->**expects**(*$this*->**once**())

            ->**method**('declineRequest')

            ->**with**(*$this*->**equalTo**(1)); *// Zakładając, że id zapytania to 1*

*// Testowanie, że metoda nie rzuca żadnym błędem*

        $managerMock->**declineRequest**(1);

    }

*// usunięcie przydziału*

*public* *function* **testDeleteAsigment**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['deleteAsigment'])

            ->**getMock**();

*// Ustawiamy, że metoda `deleteAsigment` nie zwróci żadnej wartości (void)*

        $managerMock->**expects**(*$this*->**once**())

            ->**method**('deleteAsigment')

            ->**with**(*$this*->**equalTo**(1)); *// Zakładając, że id przydziału to 1*

*// Testowanie, że metoda nie rzuca żadnym błędem*

        $managerMock->**deleteAsigment**(1);

    }

*// generowanie raportu*

*public* *function* **testGenerateReport**()

    {

        $managerMock = *$this*->**getMockBuilder**(Manager::*class*)

            ->**disableOriginalConstructor**()

            ->**onlyMethods**(['generateReport'])

            ->**getMock**();

*// Zwracamy przykładowe dane z mocka*

        $managerMock->**method**('generateReport')

            ->**willReturn**([

                ['marka' => 'Toyota', 'model' => 'Corolla', 'liczba\_wypozyczen' => 5],

                ['marka' => 'BMW', 'model' => 'X5', 'liczba\_wypozyczen' => 3]

            ]);

*// Testowanie metody*

        $result = $managerMock->**generateReport**();

*$this*->**assertCount**(2, $result); *// Oczekujemy, że wynik ma 2 elementy*

*$this*->**assertEquals**(5, $result[0]['liczba\_wypozyczen']);

    }

}

# Testy akceptacyjne klasa User – autentykacja

**1. Cel testów**

Testy akceptacyjne miały na celu zweryfikowanie poprawności procesu logowania użytkowników o różnych rolach w systemie oraz obsługi błędnych danych logowania. Sprawdzone zostały następujące scenariusze:

* Logowanie jako różni użytkownicy (admin, kierownik, serwisant, pracownik).
* Obsługa błędnych danych logowania.
* Przekierowanie użytkowników na odpowiednie strony po zalogowaniu.

**2. Wykorzystane narzędzia**

* **Framework testowy:** Codeception v5.1.2
* **PHP:** Wersja 8.x
* **Środowisko testowe:** XAMPP (localhost)
* **Kroki testowe:** Szczegóły każdego kroku testu wyświetlane za pomocą opcji --steps.

**3. Przebieg testów**

**Testy pozytywne**

1. **Logowanie jako admin**
   * **Kroki:**
     1. Otworzono stronę logowania (/index.php).
     2. Wypełniono formularz logowania danymi użytkownika "admin".
     3. Kliknięto przycisk logowania.
     4. Sprawdzono przekierowanie na stronę /views/admin.php.
   * **Wynik:** Test zakończony sukcesem (PASSED).
2. **Logowanie jako kierownik**
   * **Kroki:** Analogiczne do logowania admina, z przekierowaniem na /views/kierownik.php.
   * **Wynik:** Test zakończony sukcesem (PASSED).
3. **Logowanie jako serwisant**
   * **Kroki:** Analogiczne do logowania admina, z przekierowaniem na /views/serwisant.php.
   * **Wynik:** Test zakończony sukcesem (PASSED).
4. **Logowanie jako pracownik**
   * **Kroki:** Analogiczne do logowania admina, z przekierowaniem na /views/pracownik.php.
   * **Wynik:** Test zakończony sukcesem (PASSED).

**Testy negatywne**

1. **Logowanie z błędnymi danymi**
   * **Kroki:**
     1. Otworzono stronę logowania (/index.php).
     2. Wypełniono formularz błędnymi danymi.
     3. Kliknięto przycisk logowania.
     4. Sprawdzono, czy wyświetlony został komunikat "Błędny login lub hasło!".
   * **Wynik:** Test zakończony sukcesem (PASSED).

**Test z prawidłowymi danymi**

1. **Logowanie z poprawnymi danymi**
   * **Kroki:** Analogiczne do logowania admina, z przekierowaniem na /views/admin.php.
   * **Wynik:** Test zakończony sukcesem (PASSED).

**4. Wyniki testów**

* **Liczba scenariuszy testowych:** 6
* **Liczba kroków w scenariuszach:** 36+
* **Liczba scenariuszy zakończonych sukcesem:** 6/6
* **Całkowity czas trwania testów:** Błyskawiczny (średni czas na test: < 1s).

**5. Wnioski**

* System logowania działa poprawnie dla różnych ról użytkowników.
* Obsługa błędnych danych logowania została zaimplementowana poprawnie.
* Wszystkie użytkownicy są przekierowywani na odpowiednie strony zgodnie z ich rolami w systemie.
* Testy negatywne potwierdzają, że nieautoryzowani użytkownicy nie uzyskują dostępu do systemu.

Wynik konsola:

C:\xampp\htdocs\project>php vendor/bin/codecept run tests/Acceptance/LoginRoleRedirectCest.php --steps

Codeception PHP Testing Framework v5.1.2 https://stand-with-ukraine.pp.ua

Tests.Acceptance Tests (6) -----------------------------------------------------------------------------------------------------------------------------------------------

LoginRoleRedirectCest: Try to login as admin

Signature: Tests\Acceptance\LoginRoleRedirectCest:tryToLoginAsAdmin

Test: tests\Acceptance\LoginRoleRedirectCest.php:tryToLoginAsAdmin

Scenario --

I am on page "/index.php"

I fill field "login","admin"

I fill field "password","admin"

I click "Sign In"

I see in current url "/views/admin.php"

PASSED

LoginRoleRedirectCest: Try to login as kierownik

Signature: Tests\Acceptance\LoginRoleRedirectCest:tryToLoginAsKierownik

Test: tests\Acceptance\LoginRoleRedirectCest.php:tryToLoginAsKierownik

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

PASSED

LoginRoleRedirectCest: Try to login as serwisant

Signature: Tests\Acceptance\LoginRoleRedirectCest:tryToLoginAsSerwisant

Test: tests\Acceptance\LoginRoleRedirectCest.php:tryToLoginAsSerwisant

Scenario --

I am on page "/index.php"

I fill field "login","serwisant"

I fill field "password","serwisant"

I click "Sign In"

I see in current url "/views/serwisant.php"

PASSED

LoginRoleRedirectCest: Try to login as pracownik

Signature: Tests\Acceptance\LoginRoleRedirectCest:tryToLoginAsPracownik

Test: tests\Acceptance\LoginRoleRedirectCest.php:tryToLoginAsPracownik

Scenario --

I am on page "/index.php"

I fill field "login","pracownik1"

I fill field "password","pracownik1"

I click "Sign In"

I see in current url "/views/pracownik.php"

PASSED

LoginRoleRedirectCest: Try to login with invalid credentials

Signature: Tests\Acceptance\LoginRoleRedirectCest:tryToLoginWithInvalidCredentials

Test: tests\Acceptance\LoginRoleRedirectCest.php:tryToLoginWithInvalidCredentials

Scenario --

I am on page "/index.php"

I fill field "login","invaliduser"

I fill field "password","invalidpassword"

I click "Sign In"

I see "Błędny login lub hasło!"

PASSED

LoginRoleRedirectCest: Try to login with valid credentials

Signature: Tests\Acceptance\LoginRoleRedirectCest:tryToLoginWithValidCredentials

Test: tests\Acceptance\LoginRoleRedirectCest.php:tryToLoginWithValidCredentials

Scenario --

I am on page "/index.php"

I fill field "login","admin"

I fill field "password","admin"

I click "Sign In"

I see in current url "/views/admin.php"

PASSED

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Time: 00:01.928, Memory: 12.00 MB

OK (6 tests, 6 assertions)

Kod testu:

<?php

*namespace* **Tests\Acceptance**;

*use* Tests\Support\AcceptanceTester;

*class* LoginRoleRedirectCest

{

*// Akcja wykonywana przed każdym testem - przejście na stronę logowania*

*public* *function* **\_before**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/index.php'); *// Strona logowania*

    }

*// Test: Logowanie jako admin*

*public* *function* **tryToLoginAsAdmin**(AcceptanceTester $I)

    {

*// Wypełnij formularz logowania dla użytkownika admin*

        $I->**fillField**('login', 'admin'); *// Użytkownik admin*

        $I->**fillField**('password', 'admin'); *// Hasło admina*

        $I->**click**('Sign In'); *// Kliknij przycisk "Sign In"*

*// Sprawdź, czy użytkownik został przekierowany na stronę admina*

        $I->**seeInCurrentUrl**('/views/admin.php');  *// Przekierowanie na stronę admina*

    }

*// Test: Logowanie jako kierownik*

*public* *function* **tryToLoginAsKierownik**(AcceptanceTester $I)

    {

*// Wypełnij formularz logowania dla użytkownika kierownika*

        $I->**fillField**('login', 'kierownik'); *// Użytkownik kierownik*

        $I->**fillField**('password', 'kierownik'); *// Hasło kierownika*

        $I->**click**('Sign In'); *// Kliknij przycisk "Sign In"*

*// Sprawdź, czy użytkownik został przekierowany na stronę kierownika*

        $I->**seeInCurrentUrl**('/views/kierownik.php');  *// Przekierowanie na stronę kierownika*

    }

*// Test: Logowanie jako serwisant*

*public* *function* **tryToLoginAsSerwisant**(AcceptanceTester $I)

    {

*// Wypełnij formularz logowania dla użytkownika serwisanta*

        $I->**fillField**('login', 'serwisant'); *// Użytkownik serwisant*

        $I->**fillField**('password', 'serwisant'); *// Hasło serwisanta*

        $I->**click**('Sign In'); *// Kliknij przycisk "Sign In"*

*// Sprawdź, czy użytkownik został przekierowany na stronę serwisanta*

        $I->**seeInCurrentUrl**('/views/serwisant.php');  *// Przekierowanie na stronę serwisanta*

    }

*// Test: Logowanie jako pracownik*

*public* *function* **tryToLoginAsPracownik**(AcceptanceTester $I)

    {

*// Wypełnij formularz logowania dla użytkownika pracownika*

        $I->**fillField**('login', 'pracownik1'); *// Użytkownik pracownik*

        $I->**fillField**('password', 'pracownik1'); *// Hasło pracownika*

        $I->**click**('Sign In'); *// Kliknij przycisk "Sign In"*

*// Sprawdź, czy użytkownik został przekierowany na stronę pracownika*

        $I->**seeInCurrentUrl**('/views/pracownik.php');  *// Przekierowanie na stronę pracownika*

    }

*// Test: Logowanie z błędnym loginem i hasłem*

*public* *function* **tryToLoginWithInvalidCredentials**(AcceptanceTester $I)

    {

*// Wypełnij formularz błędnymi danymi*

        $I->**fillField**('login', 'invaliduser');

        $I->**fillField**('password', 'invalidpassword');

        $I->**click**('Sign In');

*// Sprawdź, czy pojawił się komunikat o błędnym loginie*

        $I->**see**('Błędny login lub hasło!');

    }

*public* *function* **tryToLoginWithValidCredentials**(AcceptanceTester $I)

    {

*// Wypełnij formularz błędnymi danymi*

        $I->**fillField**('login', 'admin');

        $I->**fillField**('password', 'admin');

        $I->**click**('Sign In');

        $I->**seeInCurrentUrl**('/views/admin.php');  *// Możesz zmienić na odpowiednią rolę*

    }

}

# Testy akceptacyjne klasa Manager – główne funkcjonalności aplikacji

**Opis:** Testy Akceptacyjne dla funkcji zarządzania pracownikami i pojazdami zostały przeprowadzone w systemie, aby upewnić się, że wszystkie istotne operacje są dostępne, poprawnie działają oraz wyświetlają odpowiednie informacje na stronach. Testy obejmowały takie operacje jak dodawanie, edytowanie, usuwanie pracowników, dodawanie i usuwanie pojazdów, a także akceptowanie lub odrzucanie próśb o przydziały oraz generowanie raportów.

**1. Test: Sprawdzenie dostępności strony dodawania pracownika**

**Scenariusz:**

* Zalogowanie jako kierownik
* Sprawdzenie dostępności formularza dodawania pracownika
* Weryfikacja obecności odpowiednich pól formularza

**Wynik:** Test zakończył się pomyślnie. Po kliknięciu w link do formularza dodawania pracownika użytkownik został przeniesiony na stronę z formularzem, a na niej pojawiły się odpowiednie pola (Imię, Nazwisko, PESEL, Mail, Rola, Login, Hasło).

**2. Test: Dodanie pracownika**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wypełnienie formularza danymi testowymi i kliknięcie "Dodaj"
* Sprawdzenie, czy pracownik został poprawnie dodany

**Wynik:** Test zakończył się sukcesem. Pracownik "Jan Kowalski" został dodany do systemu, a błąd przy dodawaniu nie wystąpił.

**3. Test: Edytowanie danych pracownika**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę edycji wybranego pracownika
* Wprowadzenie nowych danych i zapisanie zmian
* Sprawdzenie, czy zmiany zostały zapisane bez błędów

**Wynik:** Test zakończył się powodzeniem. Dane pracownika zostały pomyślnie edytowane i zapisane bez jakichkolwiek błędów.

**4. Test: Usunięcie pracownika**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę listy pracowników
* Wybór i usunięcie pracownika
* Weryfikacja, że pracownik został usunięty

**Wynik:** Test przeszedł pomyślnie. Po usunięciu pracownika z listy, jego dane zniknęły, a aplikacja nie zgłaszała żadnych błędów.

**5. Test: Pobranie listy wszystkich pracowników**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę z listą pracowników
* Sprawdzenie dostępności elementu zawierającego pracowników

**Wynik:** Test zakończył się sukcesem. Po wejściu na stronę z listą pracowników, pojawiła się lista z pracownikami, a aplikacja wyświetliła odpowiedni element (#pracownicy).

**6. Test: Pobranie listy wszystkich pojazdów**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę z listą pojazdów
* Sprawdzenie dostępności elementu zawierającego pojazdy

**Wynik:** Test zakończył się sukcesem. Lista pojazdów została poprawnie załadowana, a aplikacja wyświetliła element (#pojazdy).

**7. Test: Dodanie pojazdu**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wypełnienie formularza dodawania pojazdu i kliknięcie "Dodaj"
* Weryfikacja, czy pojazd został dodany do systemu

**Wynik:** Test zakończył się pomyślnie. Pojazd "Toyota Yaris" został dodany do systemu i poprawnie wyświetlił się na stronie.

**8. Test: Usunięcie pojazdu**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wybór i usunięcie pojazdu z listy
* Weryfikacja, czy pojazd został usunięty

**Wynik:** Test zakończył się powodzeniem. Pojazd "Toyota Yaris" został usunięty z systemu i już się nie pojawia na liście pojazdów.

**9. Test: Akceptowanie prośby o przydział pojazdu**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę zapytań o przydział
* Kliknięcie przycisku "Zatwierdz przydział"

**Wynik:** Test przeszedł pomyślnie. Po kliknięciu przycisku "Zatwierdz przydział", system odpowiednio zareagował.

**10. Test: Odrzucenie prośby o przydział pojazdu**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę zapytań o przydział
* Kliknięcie przycisku "Odrzuć przydział"

**Wynik:** Test zakończył się sukcesem. Prośba o przydział pojazdu została odrzucona zgodnie z oczekiwaniami.

**11. Test: Usunięcie przydziału pojazdu**

**Scenariusz:**

* Zalogowanie jako kierownik
* Wejście na stronę listy przydzielonych pojazdów
* Kliknięcie "Anuluj przydział"
* Potwierdzenie usunięcia przydziału

**Wynik:** Test zakończył się sukcesem. Po anulowaniu przydziału, pojazd "BMW" zniknął z listy przydzielonych pojazdów.

**12. Test: Generowanie raportu**

**Scenariusz:**

* Zalogowanie jako kierownik
* Kliknięcie przycisku "Raport"
* Sprawdzenie wygenerowanego raportu wypożyczeń pojazdów

**Wynik:** Test zakończył się sukcesem. Raport został wygenerowany poprawnie i wyświetlony w systemie.

**Podsumowanie:**

Wszystkie testy akceptacyjne zakończyły się powodzeniem. Funkcjonalności zarządzania pracownikami oraz pojazdami działają poprawnie, a system odpowiednio reaguje na interakcje użytkowników (kierownika). Funkcje dodawania, edytowania, usuwania oraz przeglądania danych są dostępne i działają zgodnie z wymaganiami.

Konsola wyniki:

ManagerCest: Try to see add employer page

Signature: Tests\Acceptance\ManagerCest:tryToSeeAddEmployerPage

Test: tests\Acceptance\MangerCest.php:tryToSeeAddEmployerPage

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "a[href="?add\_employer"]"

I see in current url "?add\_employer"

I see "Imię:"

I see "Nazwisko:"

I see "PESEL:"

I see "Mail:"

I see "Rola:"

I see "Login:"

I see "Hasło:"

PASSED

ManagerCest: Try to add employer

Signature: Tests\Acceptance\ManagerCest:tryToAddEmployer

Test: tests\Acceptance\MangerCest.php:tryToAddEmployer

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php?add\_employer"

I fill field "imie","Jan"

I fill field "nazwisko","Kowalski"

I fill field "pesel","22334433091"

I fill field "mail","jan.kowalski@test.com"

I select option "rola","pracownik"

I fill field "login","janek"

I fill field "password","securepassword"

I click "Dodaj"

I don't see "Wystąpił błąd podczas dodawania Pracownika."

PASSED

ManagerCest: Try to edit employer

Signature: Tests\Acceptance\ManagerCest:tryToEditEmployer

Test: tests\Acceptance\MangerCest.php:tryToEditEmployer

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Edytuj"

I see in current url "?edit"

I see "Imię:"

I see "Nazwisko:"

I see "PESEL:"

I see "Mail:"

I see "Rola:"

I see "Login:"

I see "Hasło:"

I fill field "imie","edytowane"

I fill field "nazwisko","edytowane"

I fill field "pesel","12345618901"

I fill field "mail","edytowane.edytowane@edytowane.com"

I select option "rola","pracownik"

I fill field "login","edytowane"

I fill field "password","edytowane"

I click "Zapisz zmiany"

I don't see "Dane zostały błędnie wprowadzone, proszę je poprawić!"

PASSED

ManagerCest: Test delete employee

Signature: Tests\Acceptance\ManagerCest:testDeleteEmployee

Test: tests\Acceptance\MangerCest.php:testDeleteEmployee

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/index.php"

I fill field "login","kierownik"

PASSED

ManagerCest: Test get all employers

Signature: Tests\Acceptance\ManagerCest:testGetAllEmployers

Test: tests\Acceptance\MangerCest.php:testGetAllEmployers

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Lista pracowników"

I see "Lista pracowników"

I see element "#pracownicy"

PASSED

ManagerCest: Test get all vehicles

Signature: Tests\Acceptance\ManagerCest:testGetAllVehicles

Test: tests\Acceptance\MangerCest.php:testGetAllVehicles

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Lista pojazdów"

I see "Lista pojazdów"

I see element "#pojazdy"

PASSED

ManagerCest: Test add vehicle

Signature: Tests\Acceptance\ManagerCest:testAddVehicle

Test: tests\Acceptance\MangerCest.php:testAddVehicle

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Dodaj pojazd"

I see in current url "?add\_vehicle"

I fill field "marka","Toyota"

I fill field "model","Yaris"

I fill field "dane\_serwisowe","Wymiana opon"

I fill field "rok\_produkcji","2020"

I select option "status","dostepny"

I fill field "data\_przegladu","2025-01-01"

I click "Dodaj"

I am on page "/views/kierownik.php?pojazdy"

I see "Yaris"

PASSED

ManagerCest: Test delete vehicle

Signature: Tests\Acceptance\ManagerCest:testDeleteVehicle

Test: tests\Acceptance\MangerCest.php:testDeleteVehicle

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Lista pojazdów"

I see "Toyota Yaris","#pojazdy"

PASSED

ManagerCest: Test accept request

Signature: Tests\Acceptance\ManagerCest:testAcceptRequest

Test: tests\Acceptance\MangerCest.php:testAcceptRequest

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Zapytania oczekujące"

I see "Zatwierdz przydział"

PASSED

ManagerCest: Test decline request

Signature: Tests\Acceptance\ManagerCest:testDeclineRequest

Test: tests\Acceptance\MangerCest.php:testDeclineRequest

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Zapytania oczekujące"

I see "Odrzuć przydział"

PASSED

ManagerCest: Test delete assignment

Signature: Tests\Acceptance\ManagerCest:testDeleteAssignment

Test: tests\Acceptance\MangerCest.php:testDeleteAssignment

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I click "Lista przydzielonych pojazdów"

I see in current url "?przydzialy"

I see "BMW"

I click "Anuluj przydział"

I see "Czy na pewno chcesz trwale usunąć przydział pojazdu"

I click "tak"

I don't see "BMW"

PASSED

ManagerCest: Test generate report

Signature: Tests\Acceptance\ManagerCest:testGenerateReport

Test: tests\Acceptance\MangerCest.php:testGenerateReport

Scenario --

I am on page "/index.php"

I fill field "login","kierownik"

I fill field "password","kierownik"

I click "Sign In"

I see in current url "/views/kierownik.php"

I am on page "/views/kierownik.php"

I see in current url "/views/kierownik.php"

I click "Raport"

I see in current url "?report"

I see "Raport wypożyczeń pojazdów"

PASSED

------------------------------------------------------------------------------------------------------------------------

Tests.Unit Tests (0) ---------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------

Time: 00:07.242, Memory: 14.00 MB

Kod testów:

<?php

*namespace* **Tests\Acceptance**;

*use* Tests\Support\AcceptanceTester;

*class* ManagerCest

{

*public* *function* **\_before**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/index.php'); *// Strona logowania*

*// Wypełnij formularz logowania dla użytkownika admin*

        $I->**fillField**('login', 'kierownik'); *// Użytkownik admin*

        $I->**fillField**('password', 'kierownik'); *// Hasło admina*

        $I->**click**('Sign In'); *// Kliknij przycisk "Sign In"*

*// Sprawdź, czy użytkownik został przekierowany na stronę admina*

        $I->**seeInCurrentUrl**('/views/kierownik.php');  *// Przekierowanie na stronę admina*

    }

*//Sprawdzenie dostępności strony dodawania pracownika*

*public* *function* **tryToSeeAddEmployerPage**(AcceptanceTester $I)

    {

*// Wejdź na stronę główną kierownika (lub stronę, która zawiera link do formularza)*

        $I->**amOnPage**('/views/kierownik.php');

*// Kliknij w link "Dodaj pracownika"*

        $I->**click**('a[href="?add\_employer"]'); *// klikaj za pomocą pełnej ścieżki CSS*

*// Sprawdź, czy po kliknięciu w link widok zmienia się na stronę z formularzem*

        $I->**seeInCurrentUrl**('?add\_employer');

*// Sprawdź, czy na stronie pojawiły się odpowiednie pola formularza*

        $I->**see**('Imię:');

        $I->**see**('Nazwisko:');

        $I->**see**('PESEL:');

        $I->**see**('Mail:');

        $I->**see**('Rola:');

        $I->**see**('Login:');

        $I->**see**('Hasło:');

    }

*// test dodwania pracownika*

*public* *function* **tryToAddEmployer**(AcceptanceTester $I)

    {

*// Wejdź na stronę dodawania pracownika*

        $I->**amOnPage**('/views/kierownik.php?add\_employer');

*// Wypełnij formularz danymi testowymi*

        $I->**fillField**('imie', 'Jan');

        $I->**fillField**('nazwisko', 'Kowalski');

        $I->**fillField**('pesel', '22334433091');

        $I->**fillField**('mail', 'jan.kowalski@test.com');

        $I->**selectOption**('rola', 'pracownik');

        $I->**fillField**('login', 'janek');

        $I->**fillField**('password', 'securepassword');

*// Wyślij formularz*

        $I->**click**('Dodaj');

*// Sprawdź, czy pracownik został dodany (np. sprawdzenie, czy przekierowanie działa poprawnie)*

        $I->**dontSee**('Wystąpił błąd podczas dodawania Pracownika.');

    }

*// test edycji danych pracownika*

*public* *function* **tryToEditEmployer**(AcceptanceTester $I)

    {

*// Wejdź na stronę z listą pracowników*

        $I->**amOnPage**('/views/kierownik.php');

*// Kliknij w link "Edytuj" dla wybranego pracownika*

        $I->**click**('Edytuj');

*// Sprawdź, czy otworzył się formularz edycji*

        $I->**seeInCurrentUrl**('?edit');

        $I->**see**('Imię:');

        $I->**see**('Nazwisko:');

        $I->**see**('PESEL:');

        $I->**see**('Mail:');

        $I->**see**('Rola:');

        $I->**see**('Login:');

        $I->**see**('Hasło:');

        $I->**fillField**('imie', 'edytowane');

        $I->**fillField**('nazwisko', 'edytowane');

        $I->**fillField**('pesel', '12345618901');

        $I->**fillField**('mail', 'edytowane.edytowane@edytowane.com');

        $I->**selectOption**('rola', 'pracownik');

        $I->**fillField**('login', 'edytowane');

        $I->**fillField**('password', 'edytowane');

*// Wyślij formularz*

        $I->**click**('Zapisz zmiany');

        $I->**dontSee**('Dane zostały błędnie wprowadzone, proszę je poprawić!');

    }

*// test usuwania pracownika*

*public* *function* **testDeleteEmployee**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/index.php');

        $I->**fillField**('login', 'kierownik');

        $I->**fillField**('password', 'kierownik');

        $I->**click**('Sign In');

        $I->**seeInCurrentUrl**('/views/admin.php');

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Lista pracowników');

        $I->**seeInCurrentUrl**('?employee\_list');

        $I->**see**('Jan Kowalski', '.employee-row'); *// Upewnij się, że pracownik istnieje*

        $I->**click**('Usuń', '.employee-row'); *// Kliknij przycisk usuń dla tego pracownika*

        $I->**dontSee**('Jan Kowalski', '.employee-row'); *// Sprawdź, że pracownik został usunięty*

    }

*// test pobrania wszystkich pracownikow*

*public* *function* **testGetAllEmployers**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Lista pracowników');

        $I->**see**('Lista pracowników');

        $I->**seeElement**('#pracownicy'); *// Upewnij się, że lista zawiera pracowników*

    }

*// test pobrania wszystkich pojazdow*

*public* *function* **testGetAllVehicles**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Lista pojazdów');

        $I->**see**('Lista pojazdów');

        $I->**seeElement**('#pojazdy'); *// Upewnij się, że lista zawiera pojazdy*

    }

*// test dodania pojazdu*

*public* *function* **testAddVehicle**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Dodaj pojazd');

        $I->**seeInCurrentUrl**('?add\_vehicle');

        $I->**fillField**('marka', 'Toyota');

        $I->**fillField**('model', 'Yaris');

        $I->**fillField**('dane\_serwisowe', 'Wymiana opon');

        $I->**fillField**('rok\_produkcji', '2020');

        $I->**selectOption**('status', 'dostepny');

        $I->**fillField**('data\_przegladu', '2025-01-01');

        $I->**click**('Dodaj');

        $I->**amOnPage**('/views/kierownik.php?pojazdy');

        $I->**see**('Yaris'); *// Sprawdź, że pojazd został dodany*

    }

*// test usuniecia pojazdu*

*public* *function* **testDeleteVehicle**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Lista pojazdów');

        $I->**see**('Toyota Yaris', '#pojazdy'); *// Sprawdź, czy pojazd istnieje*

        $I->**click**('Usuń', '#pojazdy');

        $I->**dontSee**('Toyota Corolla', '.vehicle-row'); *// Sprawdź, że pojazd został usunięty*

    }

*// test zaakceptowania prosby o przydzial na pojazd*

*public* *function* **testAcceptRequest**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Zapytania oczekujące');

        $I->**see**('Zatwierdz przydział');

        $I->**click**('Zatwierdz przydział');

    }

*// test odrzucenia prosby o przydzial*

*public* *function* **testDeclineRequest**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Zapytania oczekujące');

        $I->**see**('Odrzuć przydział');

        $I->**click**('Odrzuć przydział');

    }

*// test usunieacia przydzialu na pojazd z floty*

*public* *function* **testDeleteAssignment**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**click**('Lista przydzielonych pojazdów');

        $I->**seeInCurrentUrl**('?przydzialy');

        $I->**see**('BMW');

        $I->**click**('Anuluj przydział');

        $I->**see**('Czy na pewno chcesz trwale usunąć przydział pojazdu');

        $I->**click**('tak');

        $I->**dontSee**('BMW'); *// Sprawdź, że przydział został usunięty*

    }

*// test generowania raportu*

*public* *function* **testGenerateReport**(AcceptanceTester $I)

    {

        $I->**amOnPage**('/views/kierownik.php');

        $I->**seeInCurrentUrl**('/views/kierownik.php');

        $I->**click**('Raport');

        $I->**seeInCurrentUrl**('?report');

        $I->**see**('Raport wypożyczeń pojazdów');

    }

}

# Testy manualne

## Testy manualne dla wszystkich (logowanie) – wynik testów

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

## Testy manualne dla wszystkich (logowanie) – zrzuty ekranu dokonywania testów

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

UC1: Logowanie do systemu – widok

Obraz zawierający tekst, oprogramowanie, Ikona komputerowa, System operacyjny

Opis wygenerowany automatycznie

UC1: Logowanie do systemu – poprawne dane

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

UC1: Logowanie do systemu – błędne dane

**W sposób w pełni analogiczny następuje logowanie dla pozostałych grup pracowniczych (t.j. admin, pracownik, serwisant), z uwzględnieniem roli. Testy i ich rezultaty są identyczne do pozostałych grup.**

## Testy manualne dla kierownika – wynik testów

## Obraz zawierający tekst, linia, Czcionka, numer Opis wygenerowany automatycznie

## Testy manualne dla kierownika – zrzuty ekranu dokonywania testów

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, linia, oprogramowanie

Opis wygenerowany automatycznie



UC2: Wprowadzenie pracownika



Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, numer, linia

Opis wygenerowany automatycznie

UC3: Usunięcie pracownika

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

UC4: Edycja danych pracownika

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie





Obraz zawierający zrzut ekranu, tekst, oprogramowanie

Opis wygenerowany automatycznie

UC5: Wprowadzenie pojazdu do floty

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

UC6: Usunięcie pojazdu z floty



Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie



UC7: Edycja danych pojazdu

Obraz zawierający tekst, Czcionka, oprogramowanie, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, oprogramowanie, zrzut ekranu, Ikona komputerowa

Opis wygenerowany automatycznie

UC8: Rozpatrzenie zapytań pracowników

Obraz zawierający tekst, oprogramowanie, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

UC9: Podgląd przydziałów pojazdów

Obraz zawierający tekst, zrzut ekranu, numer, oprogramowanie

Opis wygenerowany automatycznie

UC10: Sporządzenie raportu

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

UC11: Edycja statusu pojazdu

## Testy manualne dla serwisanta – wynik testów

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

## Testy manualne dla serwisanta – zrzuty ekranu dokonywania testów

Obraz zawierający tekst, zrzut ekranu, numer, oprogramowanie

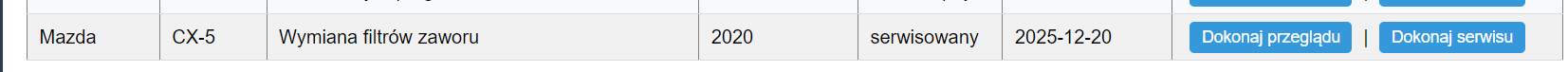
Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie



UC12: Aktualizacja danych serwisowych

## Testy manualne dla pracownika – wynik testów

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

## Testy manualne dla pracownika – zrzuty ekranu dokonywania testów

Obraz zawierający tekst, Czcionka, oprogramowanie, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

UC13: Przegląd przydziałów

Obraz zawierający tekst, numer, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie



UC14: Zgłoszenie zapytania

## Testy manualne dla administratora – wynik testów

## Obraz zawierający tekst, zrzut ekranu, Czcionka, numer Opis wygenerowany automatycznie

## Testy manualne dla administratora – zrzuty ekranu dokonywania testów

Obraz zawierający tekst, zrzut ekranu, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, oprogramowanie

Opis wygenerowany automatycznie



Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

UC15: Wprowadzanie kierownika floty

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

UC16: Usuwanie kierownika floty

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie



Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

UC17: Edycja danych kierownika floty