

Sprawozdanie z projektu: Aplikacja do zarządzania wypożyczalnią filmów online

1. Cel projektu

Celem projektu jest stworzenie aplikacji internetowej umożliwiającej użytkownikom przeglądanie, wypożyczanie oraz zarządzanie filmami online. Aplikacja ma umożliwiać rejestrację i logowanie użytkowników, dodawanie filmów, przeszukiwanie katalogu, wypożyczanie filmów oraz dodawanie opinii.

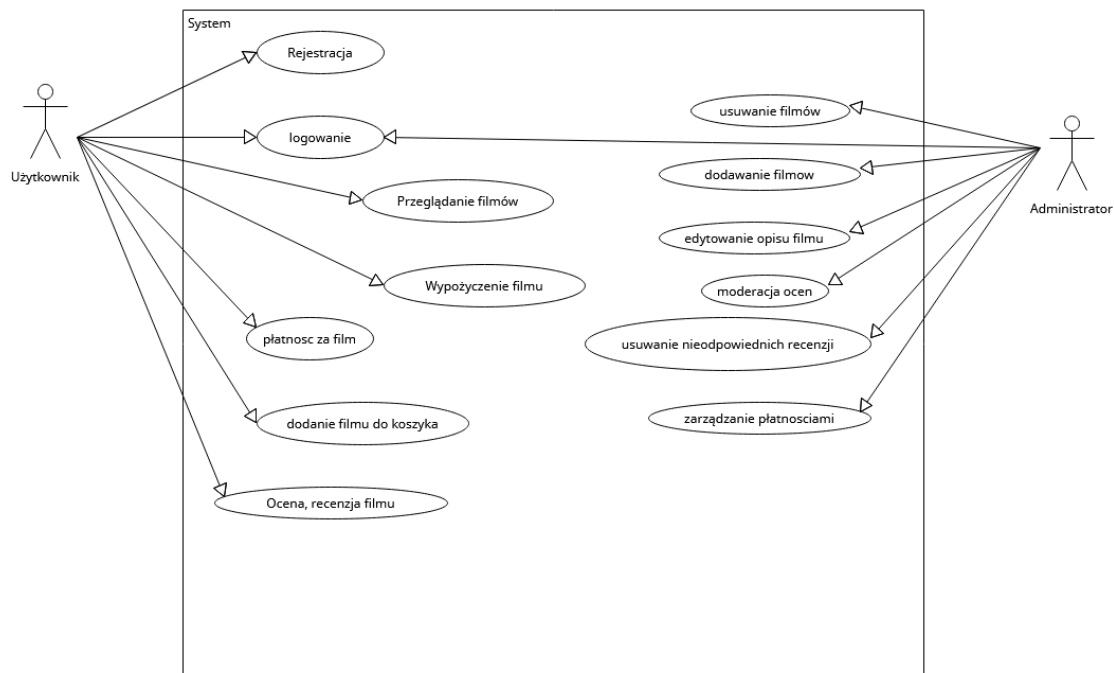
2. Główne funkcjonalności

- Rejestracja i logowanie użytkowników
- Dodawanie i edytowanie filmów
- Wyszukiwanie filmów
- Wypożyczanie filmów
- Wystawianie recenzji
- Obsługa płatności

3. Diagramy UML

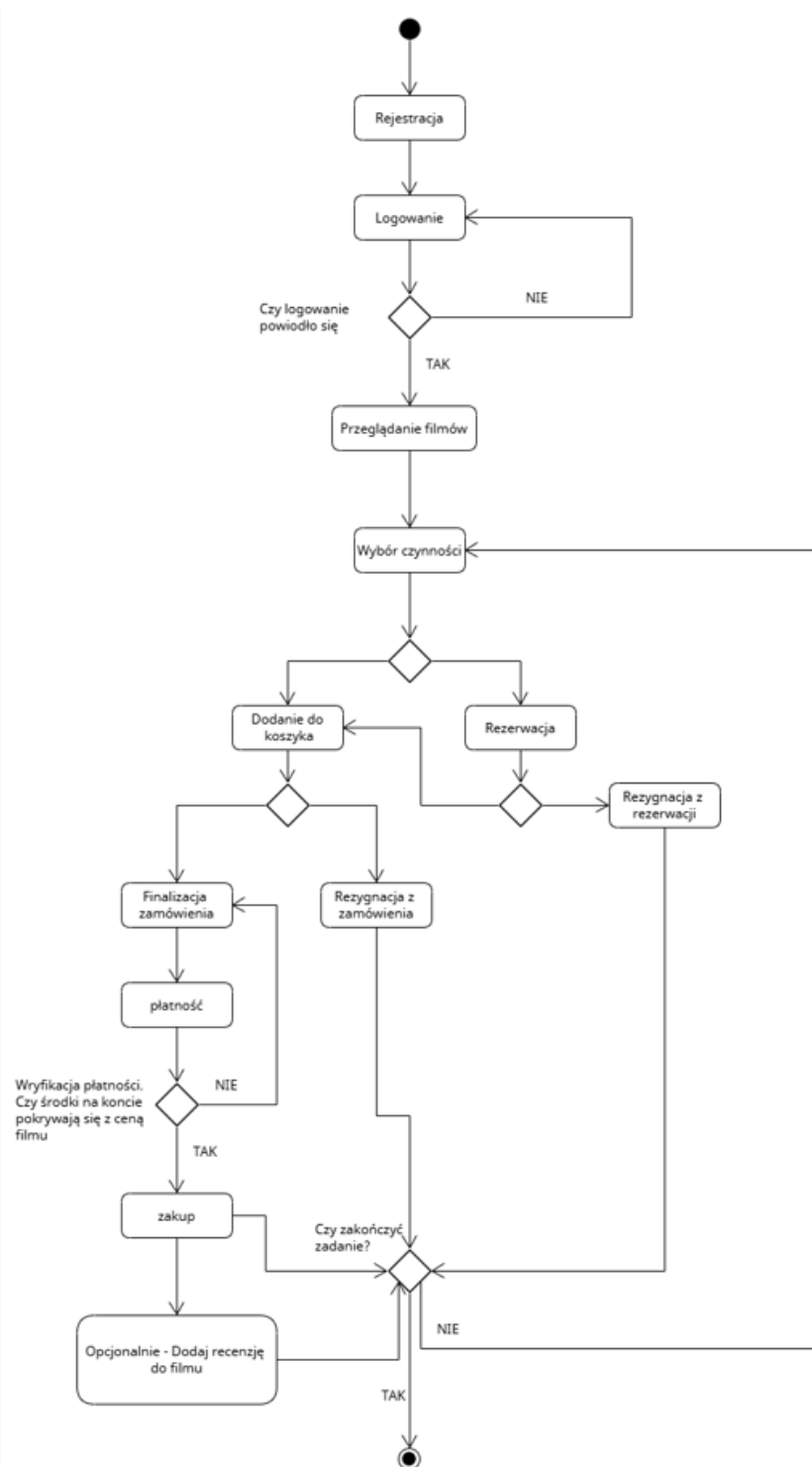
W ramach projektu wykonano następujące diagramy UML:

- Diagram przypadków użycia:



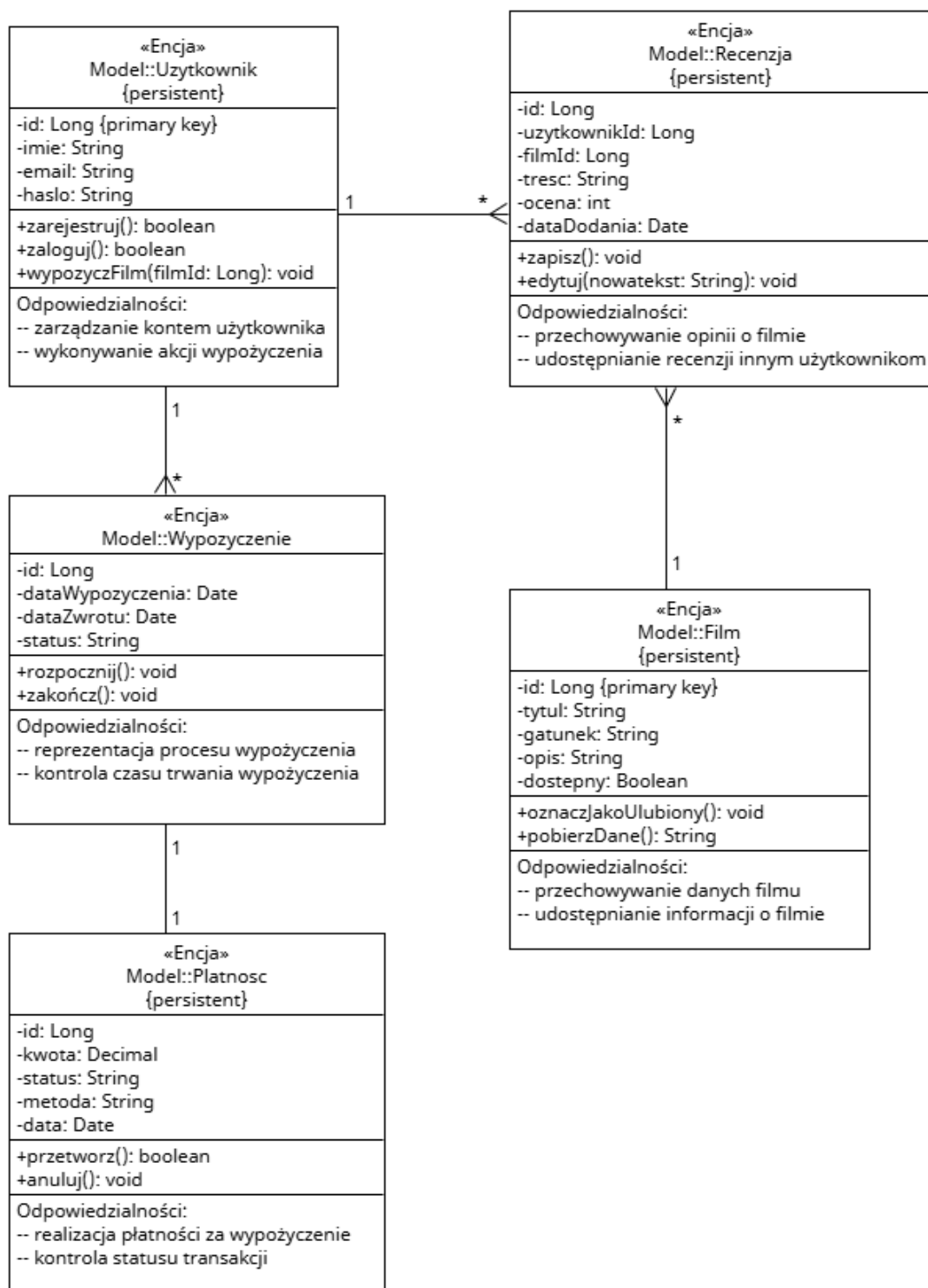
Rysunek 1 przedstawiający diagram przypadków użycia.

- Diagram czynności:



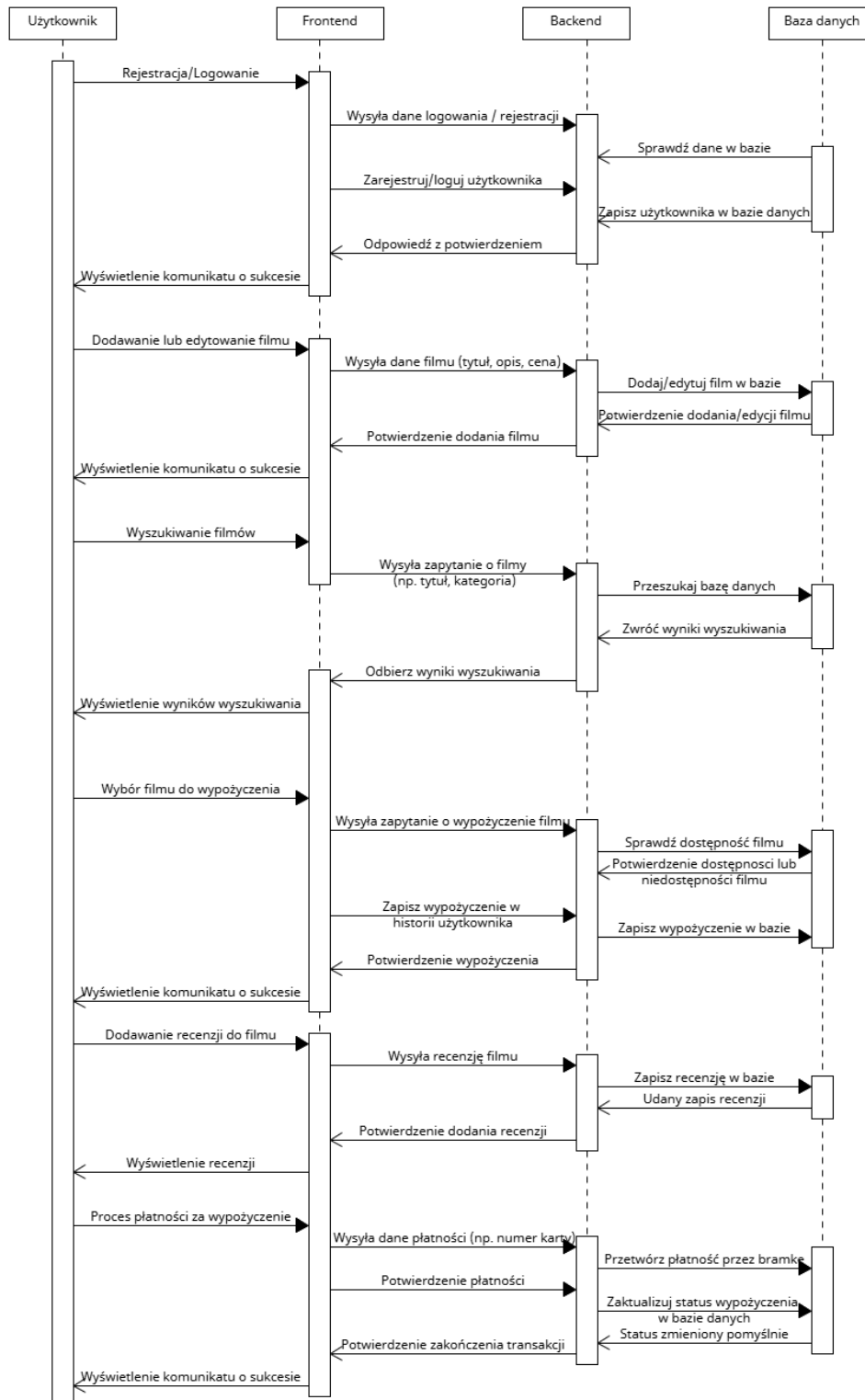
Rysunek 2 przedstawiający diagram czynności.

- Diagram klas:



Rysunek 3 przedstawiający diagram klas.

- Diagram sekwencji:



Rysunek 4 przedstawiający diagram sekwencji.

Diagramy zostały przygotowane w edytorze UMLetino i zapisane w formatach .uxf oraz .png oraz umieszczone na platformie GitHub.

4. Testy jednostkowe

Dla każdej funkcji aplikacji przygotowano co najmniej 4 testy jednostkowe z użyciem biblioteki pytest. Testowane funkcje obejmują dodawanie filmów, wyszukiwanie filmów, wypożyczanie filmów, obsługę recenzji oraz płatności.

-Dodawanie filmów

Kod źródłowy pliku test_addmovie.py:

```
import pytest

def add_movie(filmy, tytuł, gatunek, reżyser):
    film = {
        'tytuł': tytuł,
        'gatunek': gatunek,
        'reżyser': reżyser
    }
    filmy.append(film)
    return filmy

def test_add_movie_valid():
    filmy = []
    wynik = add_movie(filmy, 'Inception', 'Sci-Fi', 'Christopher Nolan')
    assert len(wynik) == 1
    assert wynik[0]['tytuł'] == 'Inception'

def test_add_movie_multiple():
    filmy = [{'tytuł': 'Inception', 'gatunek': 'Sci-Fi', 'reżyser': 'Christopher Nolan'}]
    wynik = add_movie(filmy, 'Interstellar', 'Sci-Fi', 'Christopher Nolan')
    assert len(wynik) == 2
    assert wynik[1]['tytuł'] == 'Interstellar'

def test_add_movie_invalid():
    filmy = []
    wynik = add_movie(filmy, "", 'Drama', 'Christopher Nolan')
    assert len(wynik) == 1
    assert wynik[0]['tytuł'] == ""

def test_add_movie_empty_list():
    filmy = []
    wynik = add_movie(filmy, 'Titanic', 'Romance', 'James Cameron')
    assert wynik == [{'tytuł': 'Titanic', 'gatunek': 'Romance', 'reżyser': 'James Cameron'}]

if __name__ == "__main__":
    pytest.main()
```

-Wyszukiwanie filmów

Kod źródłowy pliku test_searchmovie.py:

```
import pytest

def search_movie(filmy, tytuł):
    for film in filmy:
        if film['tytuł'].lower() == tytuł.lower():
            return film
    return None

def test_search_movie_found():
    filmy = [{'tytuł': 'Inception', 'gatunek': 'Sci-Fi', 'reżyser': 'Christopher Nolan'}]
    wynik = search_movie(filmy, 'Inception')
    assert wynik is not None
    assert wynik['tytuł'] == 'Inception'

def test_search_movie_not_found():
    filmy = [{'tytuł': 'Inception', 'gatunek': 'Sci-Fi', 'reżyser': 'Christopher Nolan'}]
    wynik = search_movie(filmy, 'Titanic')
    assert wynik is None

def test_search_movie_case_insensitive():
    filmy = [{'tytuł': 'Inception', 'gatunek': 'Sci-Fi', 'reżyser': 'Christopher Nolan'}]
    wynik = search_movie(filmy, 'inception')
    assert wynik is not None
    assert wynik['tytuł'] == 'Inception'

def test_search_movie_empty_list():
    filmy = []
    wynik = search_movie(filmy, 'Inception')
    assert wynik is None

if __name__ == "__main__":
    pytest.main()
```

- Wypożyczanie filmów

Kod źródłowy pliku test_rentmovie.py:

```
import pytest

def rent_movie(uzytkownik, film):
    if 'wypozyzione_filmy' not in uzytkownik:
        uzytkownik['wypozyzione_filmy'] = []
    uzytkownik['wypozyzione_filmy'].append(film)
```

```

    return uzytkownik

def test_rent_movie_new_user():
    uzytkownik = {}
    film = {'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan'}
    wynik = rent_movie(uzytkownik, film)
    assert 'wypozyzione_filmy' in wynik
    assert len(wynik['wypozyzione_filmy']) == 1
    assert wynik['wypozyzione_filmy'][0]['tytul'] == 'Inception'

def test_rent_movie_existing_user():
    uzytkownik = {'wypozyzione_filmy': [{'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser':
'Christopher Nolan'}]}
    film = {'tytul': 'Interstellar', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan'}
    wynik = rent_movie(uzytkownik, film)
    assert len(wynik['wypozyzione_filmy']) == 2
    assert wynik['wypozyzione_filmy'][1]['tytul'] == 'Interstellar'

def test_rent_movie_duplicate():
    uzytkownik = {'wypozyzione_filmy': [{'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser':
'Christopher Nolan'}]}
    film = {'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan'}
    wynik = rent_movie(uzytkownik, film)
    assert len(wynik['wypozyzione_filmy']) == 2
    assert wynik['wypozyzione_filmy'][1]['tytul'] == 'Inception'

def test_rent_movie_multiple_users():
    uzytkownik_1 = {'wypozyzione_filmy': [{'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser':
'Christopher Nolan'}]}
    uzytkownik_2 = {'wypozyzione_filmy': []}
    film = {'tytul': 'Interstellar', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan'}
    wynik_1 = rent_movie(uzytkownik_1, film)
    wynik_2 = rent_movie(uzytkownik_2, film)
    assert len(wynik_1['wypozyzione_filmy']) == 2
    assert len(wynik_2['wypozyzione_filmy']) == 1
    assert wynik_2['wypozyzione_filmy'][0]['tytul'] == 'Interstellar'

if __name__ == "__main__":
    pytest.main()

```

- Obsługa recenzji

Kod źródłowy pliku test_review.py:

```
import pytest
```

```

def leave_review(film, recenzja):
    if 'recenzje' not in film:
        film['recenzje'] = []
    film['recenzje'].append(recenzja)
    return film

def test_leave_review_new_review():
    film = {'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan'}
    wynik = leave_review(film, 'Amazing movie!')
    assert 'recenzje' in wynik
    assert len(wynik['recenzje']) == 1
    assert wynik['recenzje'][0] == 'Amazing movie!'

def test_leave_review_multiple_reviews():
    film = {'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan', 'recenzje':
['Great!']}
    wynik = leave_review(film, 'Amazing movie!')
    assert len(wynik['recenzje']) == 2
    assert wynik['recenzje'][1] == 'Amazing movie!'

def test_leave_review_duplicate_review():
    film = {'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan', 'recenzje':
['Great!']}
    wynik = leave_review(film, 'Great!')
    assert len(wynik['recenzje']) == 2
    assert wynik['recenzje'][1] == 'Great!'

def test_leave_review_empty_review():
    film = {'tytul': 'Inception', 'gatunek': 'Sci-Fi', 'rezyser': 'Christopher Nolan'}
    wynik = leave_review(film, '')
    assert len(wynik['recenzje']) == 1
    assert wynik['recenzje'][0] == ''

if __name__ == "__main__":
    pytest.main()

```

- Proces płatności

Kod źródłowy pliku test_processpayment.py:

```
import pytest
```

```

def process_payment(suma_platnosci):
    if suma_platnosci <= 0:

```



```
        raise ValueError("Kwota płatności musi być większa od zera.")
    return True

def test_process_payment_valid():
    wynik = process_payment(100)
    assert wynik is True

def test_process_payment_zero():
    with pytest.raises(ValueError):
        process_payment(0)

def test_process_payment_negative():
    with pytest.raises(ValueError):
        process_payment(-50)

def test_process_payment_large_amount():
    wynik = process_payment(1000000)
    assert wynik is True

if __name__ == "__main__":
    pytest.main()
```

5. Podział obowiązków

Projekt realizowany był zespołowo. Podział obowiązków przedstawia się następująco:

Konrad Borawski – funkcji programu wraz z testami do funkcji

Miłosz Dąbkowski – diagramy UML, sprawozdanie

Kamil Karwowski – diagramy UML

6. Repozytorium

Repozytorium projektu znajduje się na platformie GitHub, było ono aktualizowane po każdym zajęciach.

Link do repozytorium:

<https://github.com/Konrad-B1/Projekt-aplikacja-do-zarz-dzania-wypożyczalni-film-w-online.git>