Projekt zespołowy - projekt  
Mateusz Sierputowski gr. 31D

# Dokumentacja techniczna

Mobile Post

Spis treści

[1. Specyfikacja wymagań 4](#_Toc400017921)

[1.1. Definicja produktu 4](#_Toc400017922)

[1.2. Technologia 4](#_Toc400017923)

[1.3. Wymagania funkcjonalne 4](#_Toc400017924)

[2. Projekt techniczny 5](#_Toc400017925)

[2.1. Projekt interfejsu klienta 5](#_Toc400017926)

[Tabela 1. Główne okno aplikacji 6](#_Toc400017927)

[Rysunek 2. Kolejny widok 6](#_Toc400017928)

[Tabela 2. Pasek menu – menu „Plik” 6](#_Toc400017929)

[Rysunek 3. Pasek menu – menu „Import/eksport” 7](#_Toc400017930)

[Tabela 3. Pasek menu – menu „Import/eksport” 7](#_Toc400017931)

[Rysunek 6. Okno „Import” 8](#_Toc400017932)

[Tabela 6. Okno „Import” 8](#_Toc400017933)

[Rysunek 8. Okno "Szukaj wg albumu" 8](#_Toc400017934)

[Tabela 8. Okno "Szukaj wg albumu" 8](#_Toc400017935)

[Rysunek 10. Okno "O aplikacji...” 9](#_Toc400017936)

[Tabela 10. Okno "O aplikacji...” 9](#_Toc400017937)

[Rysunek 13. Komunikat o błędzie 9](#_Toc400017938)

[Tabela 13. Komunikat o błędzie 9](#_Toc400017939)

[2.2. Opisy funkcjonalności 10](#_Toc400017940)

[Funkcjonalność otwierania plików do edycji 10](#_Toc400017941)

[Funkcjonalność wyświetlania tagów 10](#_Toc400017942)

[Funkcjonalność edycji tagów załadowanych plików 10](#_Toc400017943)

[Funkcjonalność edycji okładki załadowanych plików 10](#_Toc400017944)

[Funkcjonalność importowania tagów z pliku tekstowego 11](#_Toc400017945)

[2.3 Diagramy UML 11](#_Toc400017946)

[2.3.1. Diagram przypadków użycia 11](#_Toc400017947)

[2.3.2. Diagramy aktywności 12](#_Toc400017948)

[2.3.4. Diagramy sekwencji 13](#_Toc400017949)

[2.3.5. Diagram obiektów 14](#_Toc400017950)

[2.3.6. Diagram klas 14](#_Toc400017951)

[3. Scenariusze testowe 15](#_Toc400017952)

[4. Plan testów jednostkowych 16](#_Toc400017953)

# 1. Specyfikacja wymagań

## 1.1. Definicja produktu

Głównym zadaniem aplikacji jest dostarczanie zamówień od klienta do firmy bądź osoby zajmującej się dostarczaniem przesyłek. W Firmie są dwie zmiany od 6: 00 do 13: 00 kurier zajmuje się odbieraniem paczek do klientów. Na kolejnej zmianie od 13: 00 do 20: 00 kurier loguje się do systemu, pobiera zadania z systemu i rozwozi paczki.

## 1.2. Technologia

Kod aplikacji będzie oparty o framework PHP symfony3.

## 1.3. Wymagania funkcjonalne

Jako klient możemy zalogować się do systemu.

Jako klient możemy zalogować się jako gość .

Jako klient możemy podać dane związane z wysyłką paczki

Jako klient możemy edytować dane podane wcześniej przez tego samego klienta.

Jako klient możemy podać dodatkowe informacje dotyczące wysyłki.

Jako klient możemy zarejestrować się i posiadać własne konto.

Jako klient możemy używać funkcjonalności związanej ze śledzeniem paczki.

Jako klient możemy pobrać unikalny kod każdej wysłanej przez niego paczki.

Jako administrator możemy zalogować się do systemu.

Jako administrator możemy zalogować się do systemu jako administrator.

Jako administrator możemy zarejestrować nowego listonosza.

Jako administrator możemy edytować dane listonoszy.

Jako administrator możemy zarządzać klientami.

Jako administrator możemy zarządzać listonoszami.

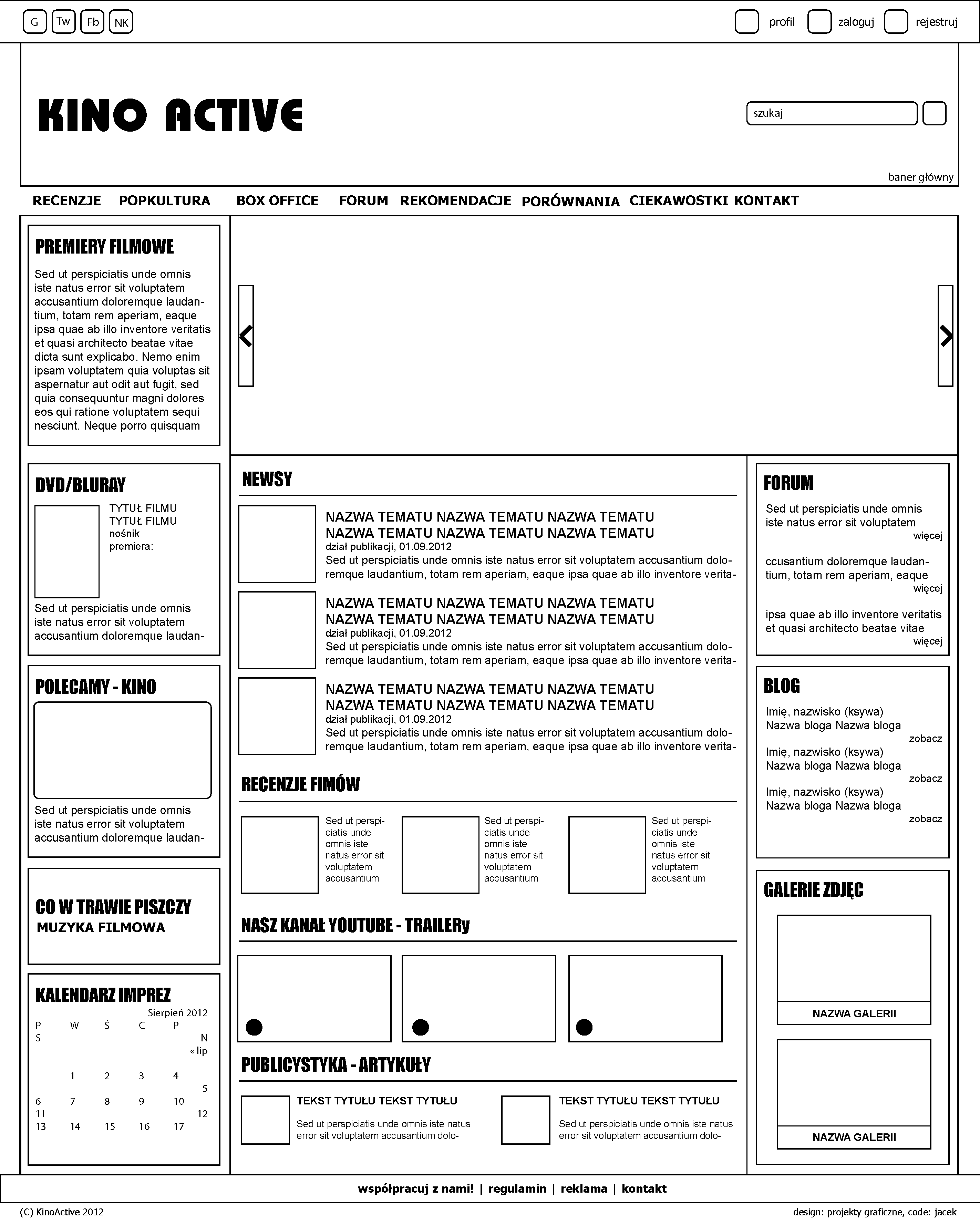
Jako administrator możemy zarządzać paczkami.

Jako administrator możemy modyfikować ustawienia systemu

Jako administrator możemy zmieniać parametry takie jak płatność,rodzaj dostawy.

# 2. Projekt techniczny

## 2.1. Projekt interfejsu klienta

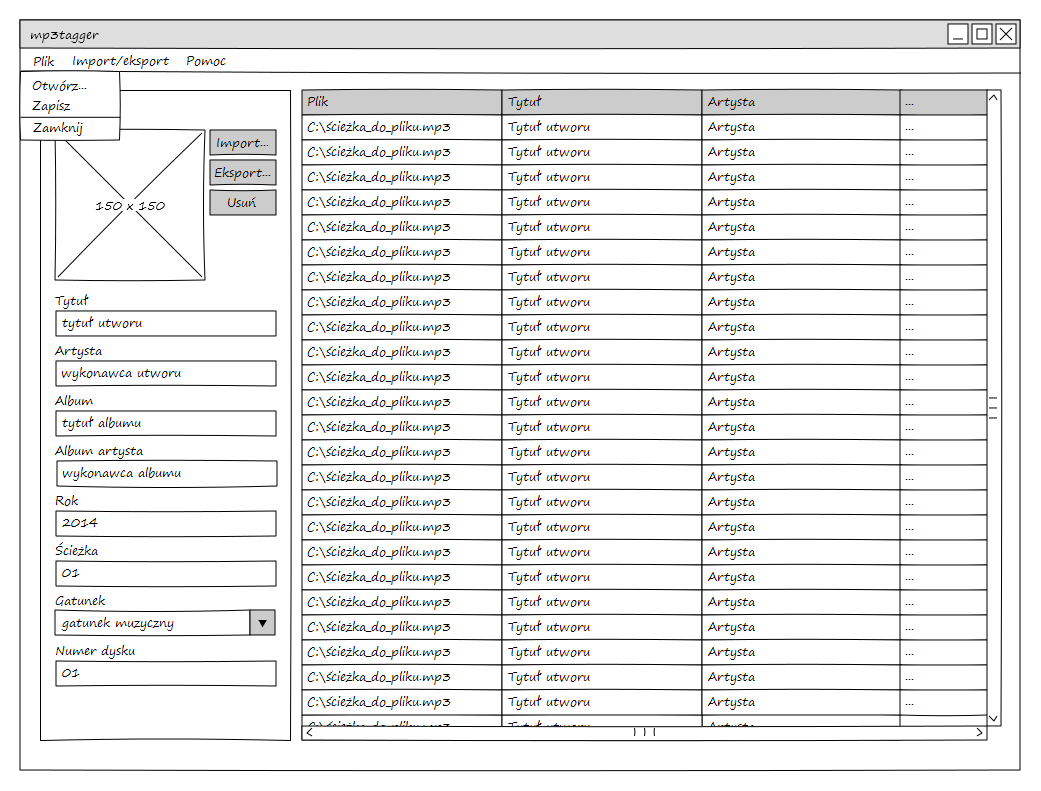


Rysunek 1. Główne okno aplikacji.

### Tabela 1. Główne okno aplikacji

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *1* | Przycisk „profil” | Opis funkcji przycisku |
| *2* | Przycisk „zaloguj” | Opis funkcji przycisku |
| *3* | Przycisk „rejestruj” | Opis funkcji przycisku |
| *4* | Sekcja „PREMIERY FILMOWE” | Opis sekcji |
| *5* | Moduł „Newsy” | Opis modułu |
| *6* | … | … |

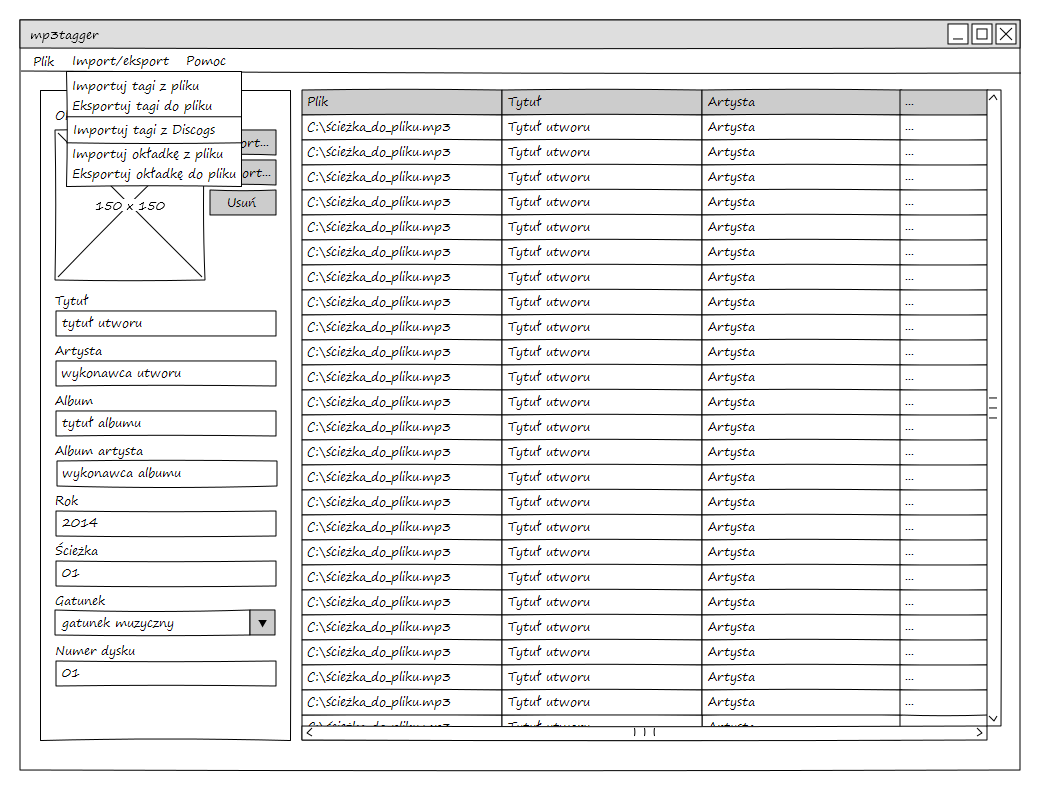
### Rysunek 2. Kolejny widok



### Tabela 2. Pasek menu – menu „Plik”

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *15* | Komenda „Otwórz” | Otwiera okno pozwalające otworzyć pliki, których tagi chcemy edytować. |
| *16* | Komenda „Zapisz” | Pozwala zapisać dokonane zmiany |
| *17* | Komenda „Zamknij” | Zamyka program, jeśli istnieją jakieś niezapisane zmiany wyświetla okno z pytaniem o zapisanie i zamknięcie, porzucenie zmian i zamknięcie lub anulowanie i powrót do programu. |

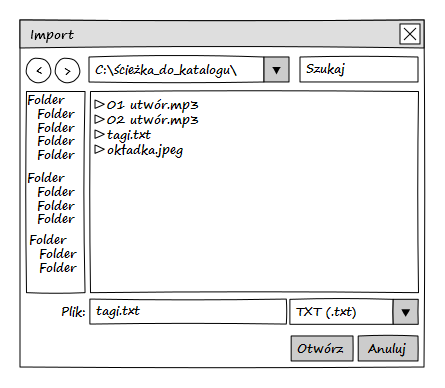
### Rysunek 3. Pasek menu – menu „Import/eksport”



### Tabela 3. Pasek menu – menu „Import/eksport”

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *18* | Komenda „Importuj tagi z pliku” | Otwiera okno pozwalające wybrać ścieżkę do pliku tekstowego oraz wyświetlające krótki poradnik, jak przygotować plik tekstowy do importu tagów. |
| *19* | Komenda „Eksportuj tagi do pliku” | Otwiera okno pozwalające zapisać tagi wybranych plików jako plik tekstowy. |
| *20* | Komenda „Importuj tagi z Discogs” | Otwiera okno „Szukaj wg albumu”. |
| *21* | Komenda „Importuj okładkę z pliku” | Otwiera okno pozwalające wybrać plik graficzny, z którego chcemy zaimportować okładkę. |
| *22* | Komenda „Eksportuj okładkę do pliku” | Otwiera okno pozwalające zapisać aktualnie ustawioną okładkę jako plik graficzny. |

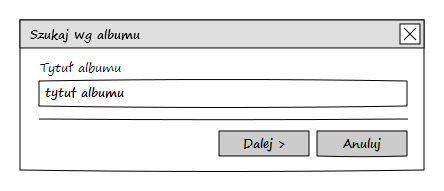
### Rysunek 6. Okno „Import”



### Tabela 6. Okno „Import”

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *27* | Przycisk „Otwórz” | Otwiera plik |
| *28* | Przycisk „Anuluj” | Zamyka okno. |
| *29* | Pola wyboru i edycyjne „Folder” i „Plik” | Umożliwiają wprowadzenie ścieżki do katalogu lub do pliku, który klient chce otworzyć. |

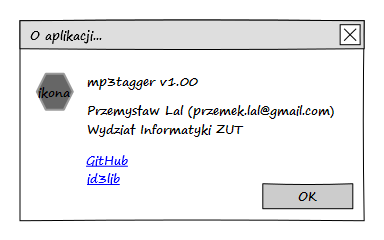
### Rysunek 8. Okno "Szukaj wg albumu"



### Tabela 8. Okno "Szukaj wg albumu"

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *34* | Pole edycyjne „Tytuł” | Umożliwia wprowadzenie nazwy albumu, dla którego chcemy znaleźć tagi w bazie Discogs. |
| *35* | Przycisk „Dalej” | Wyszukuje w bazie danych Discogs podanego albumu. |
| *36* | Przycisk „Anuluj” | Przerywa działanie. |

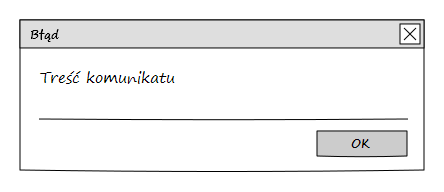
### Rysunek 10. Okno "O aplikacji...”



### Tabela 10. Okno "O aplikacji...”

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *48* | Łącze „GitHub” | Łącze do projektu w serwisie GitHub. |
| *49* | Łącze „id3lib” | Łącze do strony domowej biblioteki id3lib. |
| *50* | Przycisk „OK” | Zamyka okno. |

### Rysunek 13. Komunikat o błędzie



### Tabela 13. Komunikat o błędzie

|  |  |  |
| --- | --- | --- |
| ***ID*** | **Nazwa** | **Opis** |
| *55* | Przycisk „OK” | Zamyka okno komunikatu. |

## 2.2. Opisy funkcjonalności

### Funkcjonalność otwierania plików do edycji

#### Scenariusz główny

1. Klient w oknie głównym z paska menu wybiera „Plik”. (Rys. 1)
2. Klient wybiera opcję „Otwórz” (Rys. 2).
3. Klient w wyświetlonym oknie wybiera pliki, których tagi chce edytować. (Rys. 5).
4. Klient klika przycisk „OK” (Rys. 5).
5. Program dodaje pliki do listy plików w oknie głównym (Rys. 1).

### Funkcjonalność wyświetlania tagów

#### Scenariusz główny

1. Tagi wszystkich załadowanych plików wyświetlają się na liście plików w oknie głównym  
   (Rys. 1).

#### Scenariusz alternatywny – klient zaznacza jeden plik na liście

2a. Klient zaznacza jeden plik na liście (Rys. 1).  
3a. Wszystkie tagi pliku zostają wyświetlone w polach w panelu bocznym (Rys. 1).

#### Scenariusz alternatywny – klient zaznacza więcej niż jeden plik na liście

2b. Klient zaznacza więcej niż jeden plik na liście (Rys. 1).  
3b. Wspólne wartości tagów zostają wyświetlone w polach w panelu bocznym, dla różniących się w polu zostaje umieszczona informacja „(wiele)”.

### Funkcjonalność edycji tagów załadowanych plików

#### Scenariusz główny

1. Klient zaznacza wybrany plik/wybrane pliki na liście plików w oknie głównym (Rys. 1).
2. Klient wprowadza wybrane wartości tagów w polach edycji (ID 4-ID 11) w lewym panelu okna głównego (Rys. 1). (Wprowadzone tagi dotyczą wszystkich zaznaczonych plików, np. edycja pola ARTYSTA przy więcej niż jednym zaznaczonym pliku spowoduje ustawienie takiej wartości pola ARTSYTA dla wszystkich zaznaczonych plików.)

### Funkcjonalność edycji okładki załadowanych plików

#### Scenariusz główny – import okładki z pliku

1. Klient zaznacza wybrany plik/wybrane pliki na liście plików w oknie głównym (Rys. 1).
2. Klient klika przycisk „Import…” (Rys. 1).
3. Zostaje uruchomiona funkcjonalność importowania okładki.

#### Scenariusz alternatywny – eksport okładki do pliku

2a. Klient klika przycisk „Eksport…” (Rys. 1).  
3a. Zostaje uruchomiona funkcjonalność eksportowania okładki.

#### Scenariusz alternatywny – wyczyszczenie pola okładka

2b. Klient kilka przycisk „Usuń” (Rys. 1).  
3b. Pole tagu OKŁADKA zostaje wyczyszczone.

### Funkcjonalność importowania tagów z pliku tekstowego

#### Scenariusz główny

1. Klient zaznacza wybrane pliki na liście plików w oknie głównym (Rys. 1).
2. Klient w oknie głównym z paska menu wybiera opcję „Import/eksport” (Rys. 1).
3. Klient wybiera opcję „Importuj tagi z pliku” (Rys. 3).
4. Wyświetlone zostaje okno z informacją jak powinien wyglądać plik, aby został poprawnie sparsowany (Rys. 11).
5. Wyświetlone zostaje okno „Import” (Rys. 6).
6. Klient w oknie wybiera plik, z którego chce zaimportować tagi.
7. Klient klika przycisk „Otwórz” (Rys. 6).
8. Wartości tagów zostają ustawione dla zaznaczonych plików.

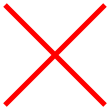
#### Scenariusz alternatywny - błąd odczytu

8a. W przypadku błędu zostaje wyświetlony stosowny komunikat.

## 2.3 Diagramy UML

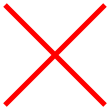
### 2.3.1. Diagram przypadków użycia

### Diagramy aktywności

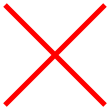


### 2.3.4. Diagramy sekwencji

### 2.3.5. Diagram obiektów



### 2.3.6. Diagram klas



# Scenariusze testowe

|  |  |
| --- | --- |
| Nazwa | Otwieranie pliku do edycji |
| Warunki początkowe | Aplikacja jest otwarta. |
| Przebieg testu | 1. Klient w oknie głównym z paska menu wybiera „Plik”. 2. Klient wybiera opcję „Otwórz”. 3. Klient w wyświetlonym oknie wybiera plik we wspieranym formacie (mp3, aiff, wma, mp4 lub m4a). 4. Klient klika przycisk „OK”. |
| Warunki końcowe | Program dodaje pliki do listy plików w oknie głównym (Rys. 1). |

|  |  |
| --- | --- |
| Nazwa | Otwieranie pliku do edycji |
| Warunki początkowe | Aplikacja jest otwarta. |
| Przebieg testu | 1. Klient w oknie głównym z paska menu wybiera „Plik”. 2. Klient wybiera opcję „Otwórz”. 3. Klient w wyświetlonym oknie wybiera plik w niewspieranym formacie (txt). 4. Klient klika przycisk „OK”. |
| Warunki końcowe | Program dodaje pliki do listy plików w oknie głównym. |

# 4. Plan testów jednostkowych

Testy wykonywane przy użyciu „Microsoft Unit Testing Framework for C++”

|  |  |
| --- | --- |
| Testowana metoda | int Cmp3taggerApp::openFile(CString path, CString ext) |
| Dane wejściowe | path = L”testfiles/song.mp3”;  ext = L”mp3”; |
| Oczekiwana wartość | 0 |

|  |  |
| --- | --- |
| Testowana metoda | int Cmp3taggerApp::openFile(CString path, CString ext) |
| Dane wejściowe | path = L”testfiles/song.wma”;  ext = L”wma”; |
| Oczekiwana wartość | 0 |