

UNIwersYTET RZESZOWSKI
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH
INSTYTUT INFORMATYKI



Kamil Antoni Krawczyk
134931

Informatyka

Dokumentacja Systemu Zarządzania OSP

Praca projektowa

Praca wykonana pod kierunkiem
mgr inż. Ewa Żesławska

Rzeszów 2025

Spis treści

1. Opis założeń projektu	6
2. Opis Struktury Projektu	8
3. Harmonogram Realizacji Projektu	10
4. Prezentacja Warstwy Użytkowej Projektu	11
4.1. Główne Menu Aplikacji	12
4.2. Moduł Zarządzania Strażakami	13
4.3. Moduł Zarządzania Interwencjami	17
4.4. Moduł Zarządzania Pojazdami	20
4.5. Panel Użytkownika o Ograniczonych Uprawnieniach	23
5. Aspekty Techniczne i Zarządzanie Projektem	27
6. Podsumowanie	29
Bibliografia	30
Spis rysunków	31
Oświadczenie studenta o samodzielności pracy	32

1. Opis założeń projektu

Celem projektu było zaprojektowanie oraz zaimplementowanie zintegrowanego systemu informatycznego przeznaczonego do zarządzania zasobami Ochotniczej Straży Pożarnej. Stworzona aplikacja desktopowa, nazwana „System Zarządzania OSP”, ma na celu zastąpienie tradycyjnych, papierowych form ewidencji, oferując w zamian szybki dostęp do danych oraz ich łatwą modyfikację.

Projekt został zrealizowany jako aplikacja desktopowa, która łączy się z centralną bazą danych. Główne funkcjonalności systemu zostały podzielone na logiczne moduły, które obejmują kluczowe aspekty działalności jednostki OSP.

- **System Uwierzytelniania Użytkowników:** Aplikacja posiada wbudowany ekran logowania, który weryfikuje tożsamość użytkownika na podstawie danych z bazy. System obsługuje również różne role (np. „admin”), co pozwala na zróżnicowanie poziomu dostępu do poszczególnych funkcji.
- **Moduł Zarządzania Strażakami:** Umożliwia pełne zarządzanie danymi strażaków (operacje CRUD). Przechowuje informacje takie jak dane osobowe, stopień, data wstąpienia do służby czy ważność badań lekarskich.
- **Moduł Zarządzania Interwencjami:** Służy do ewidencjonowania wszystkich akcji ratowniczych. Pozwala na zapisywanie szczegółów zdarzenia, takich jak data, rodzaj, miejsce, opis działań, a także lista uczestniczących w akcji strażaków i wykorzystanych pojazdów.
- **Moduł Zarządzania Pojazdami:** Umożliwia prowadzenie ewidencji floty pojazdów jednostki, przechowując ich oznaczenia taktyczne oraz numery rejestracyjne.
- **Nawigacja:** Po zalogowaniu użytkownik ma dostęp do centralnego menu, z którego może nawigować do poszczególnych modułów systemu.

Do realizacji projektu wykorzystano sprawdzone i stabilne technologie, które zapewniły odpowiednią funkcjonalność oraz wydajność aplikacji.

- **Język Programowania:** Cała logika aplikacji została zaimplementowana w języku **Java**, co gwarantuje jej wieloplatformowość.
- **Interfejs Graficzny Użytkownika (GUI):** Interfejs został zbudowany przy użyciu biblioteki **Java Swing**, która jest standardowym narzędziem do tworzenia aplikacji okienkowych w Javie.
- **Baza Danych:** System opiera się na relacyjnej bazie danych **MySQL**. Nazwa schematu bazy to **szosp**.
- **Komunikacja z Bazą Danych:** Do połączenia aplikacji z bazą danych wykorzystano sterownik **JDBC (Java Database Connectivity)**, który jest standardowym API Javy do wykonywania zapytań SQL.

Wymagania niefunkcjonalne

Poniżej przedstawiono wymagania niefunkcjonalne dla projektu „System Zarządzania OSP”:

- **Bezpieczeństwo:**

- System powinien weryfikuje tożsamość użytkownika przy pomocy loginu i hasła.
- Różne poziomy dostępu (np. administrator, użytkownik) ograniczają dostęp do wybranych funkcjonalności.

- **Dostępność i niezawodność:**

- W przypadku problemu z połączeniem do bazy danych, użytkownik otrzymuje stosowny komunikat błędu.

- **Użyteczność:**

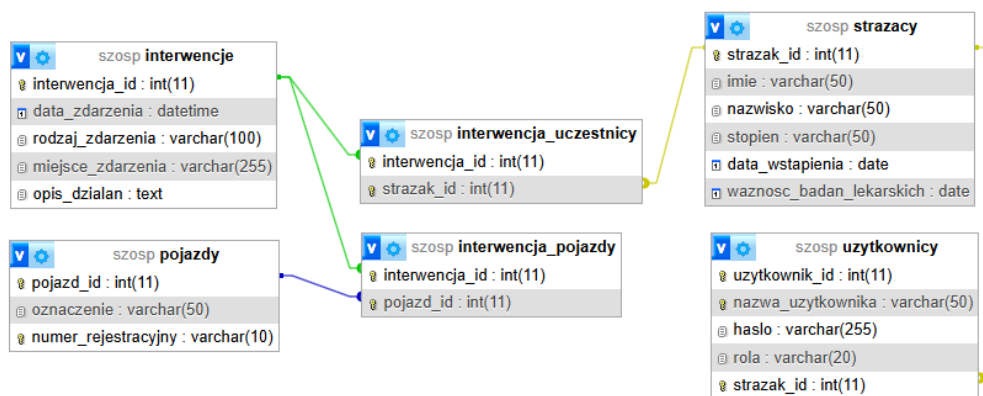
- Interfejs graficzny jest intuicyjny i przyjazny dla użytkowników nietechnicznych.
- Wszystkie komunikaty i formularze są w języku polskim.

- **Odporność na błędy i backup:**

- Wszystkie krytyczne operacje na danych są wykonywane w ramach transakcji.

2. Opis Struktury Projektu

Fundamentem aplikacji jest relacyjna baza danych o nazwie `szosp`, która została zaprojektowana w celu przechowywania i zarządzania wszystkimi informacjami dotyczącymi strażaków, pojazdów, interwencji oraz użytkowników systemu. Baza danych składa się z czterech głównych tabel oraz dwóch tabel łączących, które realizują relacje wiele-do-wielu. Schemat bazy danych został przedstawiony na diagramie encji-związków (Rys. 2.1).¹



Rys. 2.1. Diagram encji-związków (ERD) dla bazy danych `szosp`.

Tabela `strazacy`

Tabela ta przechowuje podstawowe dane osobowe oraz służbowe każdego strażaka.

- `strazak_id`: Klucz główny, unikalny identyfikator strażaka (INT).
- `imie`: Imię strażaka (VARCHAR(50)).
- `nazwisko`: Nazwisko strażaka (VARCHAR(50)).
- `stopien`: Stopień służbowy (VARCHAR(50)).
- `data_wstapienia`: Data wstąpienia do służby (DATE).
- `waznosc_badan_lekarskich`: Data ważności badań lekarskich (DATE).

Tabela `uzytkownicy`

Tabela odpowiedzialna za przechowywanie danych logowania i uprawnień użytkowników systemu.

- `uzytkownik_id`: Klucz główny, unikalny identyfikator użytkownika (INT).
- `nazwa_uzytkownika`: Login użytkownika (VARCHAR(50)).

¹ JavaStart: JDBC – Podstawy pracy z bazą danych. <https://javastart.pl/baza-wiedzy/java-ee/jdbc-podstawy-pracy-z-baza-danych>
[Dostęp: 14.06.2025]

- **haslo:** Hasło użytkownika, przechowywane jako skrót (VARCHAR(255)).
- **rola:** Rola użytkownika w systemie, np. admin (VARCHAR(20)).
- **strazak_id:** Klucz obcy wskazujący na powiązanego strażaka (INT).

Tabela pojazdy

Ewidencja pojazdów będących na wyposażeniu jednostki.

- **pojazd_id:** Klucz główny, unikalny identyfikator pojazdu (INT).
- **oznaczenie:** Oznaczenie taktyczne pojazdu (VARCHAR(50)).
- **numer_rejestracyjny:** Numer rejestracyjny pojazdu (VARCHAR(10)).

Tabela interwencje

Główna tabela przechowująca informacje o przeprowadzonych interwencjach.

- **interwencja_id:** Klucz główny, unikalny identyfikator interwencji (INT).
- **data_zdarzenia:** Dokładna data i czas zdarzenia (DATETIME).
- **rodzaj_zdarzenia:** Kategoria zdarzenia (VARCHAR(100)).
- **miejsce_zdarzenia:** Adres/lokalizacja zdarzenia (VARCHAR(255)).
- **opis_dzialan:** Szczegółowy opis podjętych działań (TEXT).

Tabele Łączące

W celu zrealizowania relacji wiele-do-wielu, w projekcie wykorzystano dwie tabele pośredniczące.

- **interwencja_uczestnicy:** Tabela łącząca interwencje ze strażakami, którzy brali w nich udział. Składa się z dwóch kluczy obcych: `interwencja_id` oraz `strazak_id`.
- **interwencja_pojazdy:** Tabela łącząca interwencje z pojazdami, które były do nich zadysponowane. Składa się z dwóch kluczy obcych: `interwencja_id` oraz `pojazd_id`.

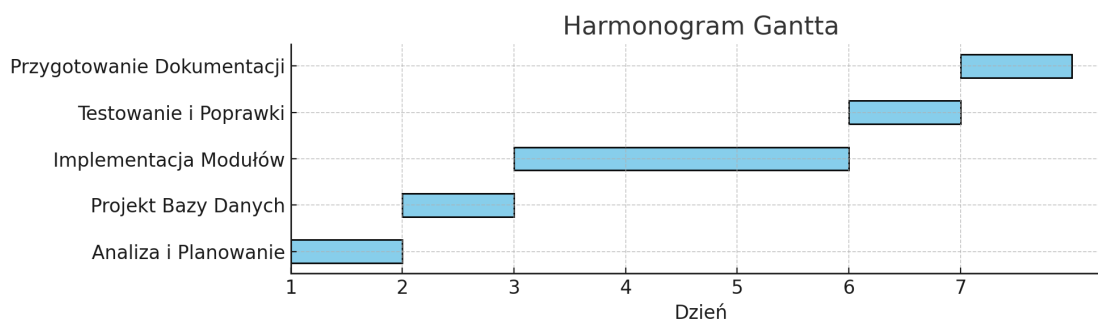
Relacje Między Tabelami

Diagram ERD (Rys. 2.1) ilustruje kluczowe powiązania między tabelami.

- **Relacja jeden-do-jednego** występuje między tabelami `strazacy` a `uzytkownicy`. Każdy strażak może mieć dokładnie jedno konto użytkownika w systemie, a każde konto jest przypisane do jednego strażaka. Relacja ta jest realizowana poprzez klucz obcy `strazak_id` w tabeli `uzytkownicy`.
- **Relacja wiele-do-wielu** łączy tabelę `interwencje` z tabelą `strazacy` za pośrednictwem tabeli `interwencja_uczestnicy`. Pozwala to na przypisanie wielu strażaków do jednej interwencji oraz na udział jednego strażaka w wielu różnych interwencjach.
- **Relacja wiele-do-wielu** łączy również tabelę `interwencje` z tabelą `pojazdy` za pomocą tabeli `interwencja_pojazdy`. Dzięki temu do jednej interwencji można przypisać wiele pojazdów, a jeden pojazd może być wykorzystywany w wielu akcjach.

3. Harmonogram Realizacji Projektu

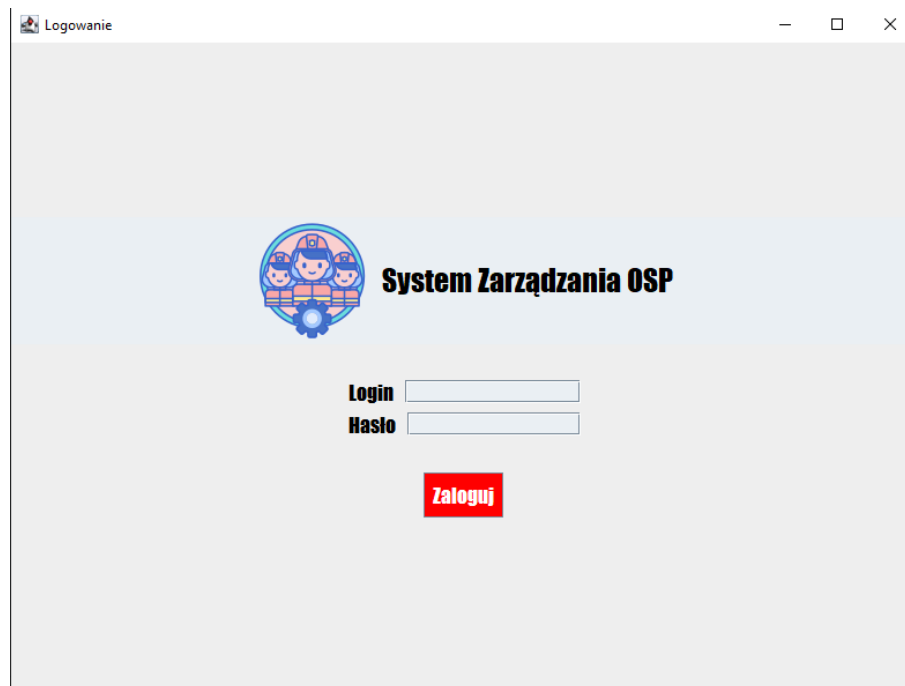
Prace rozpoczęto od fazy analizy i planowania, która zajęła pierwszy dzień. Drugiego dnia zrealizowano projekt bazy danych. Najdłuższy, trwający od trzeciego do piątego dnia, był etap implementacji kluczowych modułów systemu. Szósty dzień w całości poświęcono na testowanie funkcjonalności i wprowadzanie niezbędnych poprawek. Ostatni, siódmy dzień, przeznaczono na przygotowanie kompletnej dokumentacji technicznej i użytkowej projektu (Rys. 3.1).



Rys. 3.1. Diagram Gantta przedstawiający tygodniowy harmonogram prac nad projektem.

4. Prezentacja Warstwy Użytkowej Projektu

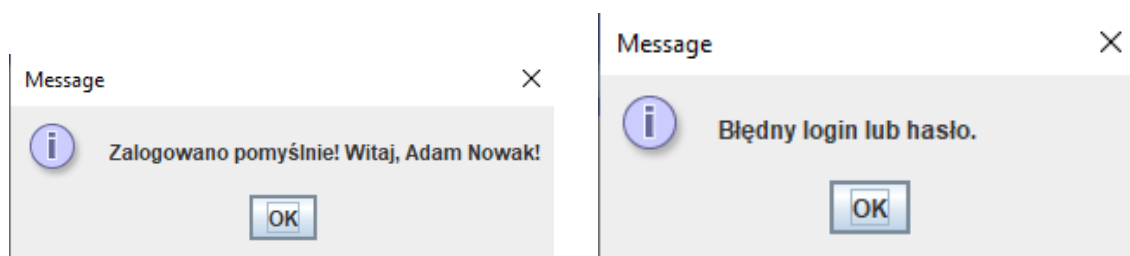
Pierwszym elementem, z którym użytkownik ma styczność, jest okno logowania. Zostało ono zaprojektowane w celu weryfikacji tożsamości użytkownika i nadania mu odpowiednich uprawnień w systemie. Interfejs okna logowania (Rys. 4.1) składa się z pól tekstowych do wprowadzenia nazwy użytkownika i hasła oraz przycisku "Zaloguj". Pola te są obsługiwane przez komponenty 'JTextField' dla loginu i 'JPasswordField' dla hasła, co zapewnia maskowanie wpisywanych znaków.



Rys. 4.1. Główne okno logowania do systemu.

Po kliknięciu przycisku "Zaloguj", uruchamiany jest 'ActionListener'. Proces weryfikacji przebiega następująco:

1. Pobierane są dane z pól 'LoginInput' i 'PasswordInput'.
2. Następuje walidacja, czy pola nie są puste. Jeśli tak, wyświetlany jest komunikat z prośbą o uzupełnienie danych.
3. Konstruowane jest zapytanie SQL, które ma na celu odnalezienie użytkownika o podanym loginie i hasle w bazie danych. Zapytanie łączy tabele 'uzytkownicy' i 'strazacy', aby pobrać rolę, imię oraz nazwisko strażaka.
4. Na podstawie pobranej z bazy roli, system decyduje, które okno aplikacji otworzyć – 'Start' dla administratora lub 'UzytkownikStart' dla zwykłego użytkownika.
5. W przypadku błędnych danych, użytkownik widzi stosowny komunikat (Rys. 4.2b). System jest również przygotowany na obsługę błędów połączenia z bazą danych.



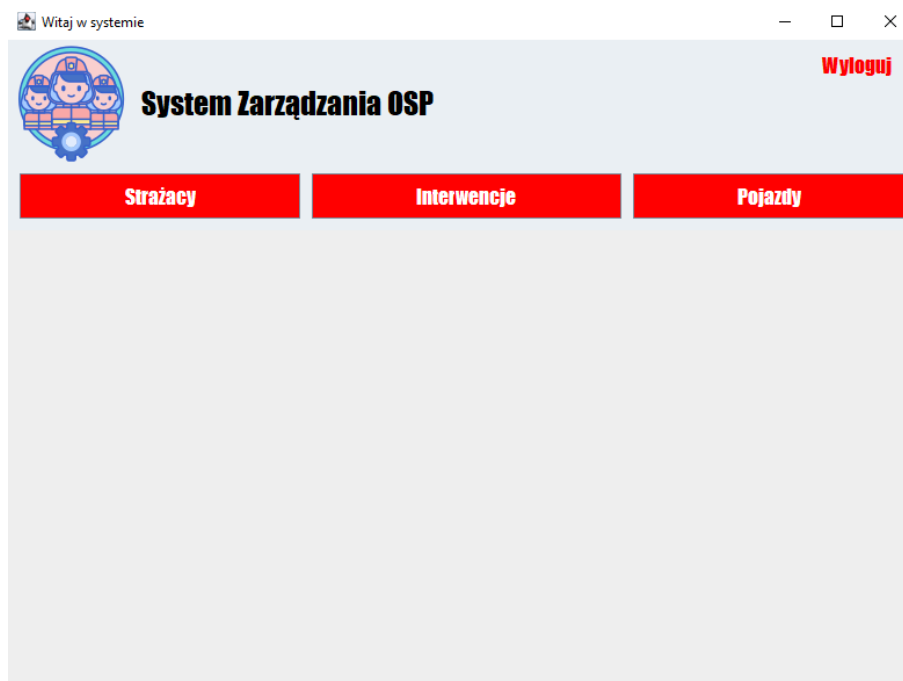
(a) Komunikat o pomyślnym zalogowaniu.

(b) Komunikat o błędnych danych logowania.

Rys. 4.2. Komunikaty zwrotne w procesie logowania.

4.1. Główne Menu Aplikacji

Po pomyślnym zalogowaniu użytkownik zostaje przekierowany do głównego menu, które stanowi centralny punkt nawigacyjny aplikacji. Interfejs menu (Rys. 4.3) zawiera przyciski nawigacyjne: "Strażacy", "Interwencje", "Pojazdy" oraz przycisk "Wyloguj". Każdy z przycisków posiada zaimplementowany 'ActionListener', który po kliknięciu zamyka aktualne okno ('dispose()') i otwiera nowe, odpowiadające wybranej funkcjonalności. Na przykład, kliknięcie "Strażacy" tworzy i wyświetla obiekt klasy 'Strażacy'.

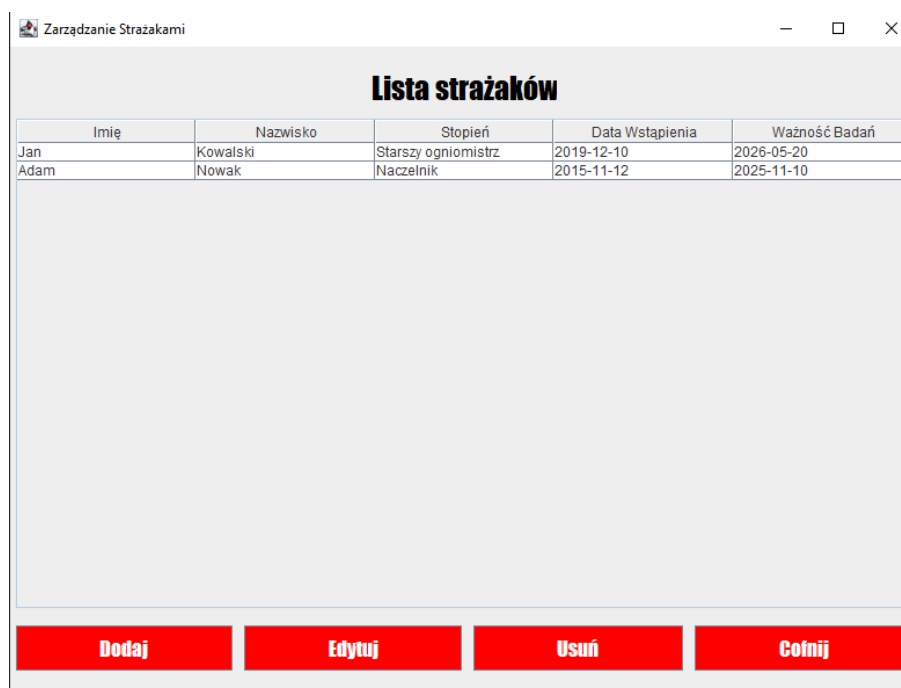
**Rys. 4.3.** Główne menu aplikacji.

4.2. Moduł Zarządzania Strażakami

Moduł zarządzania strażakami jest kluczową funkcjonalnością systemu, umożliwiającą administratorowi wykonywanie pełnego zakresu operacji CRUD (Create, Read, Update, Delete) na danych pracowników. Każda operacja została zaimplementowana w dedykowanej klasie, co zapewnia wysoką organizację kodu i łatwość w zarządzaniu.

Widok Główny i Wczytywanie Danych

Główny widok modułu (Rys. 4.4) prezentuje listę wszystkich strażaków w formie tabeli. Jest to podstawowy interfejs, z którego użytkownik może nawigować do bardziej szczegółowych operacji. Poniżej tabeli znajdują się przyciski akcji: "Dodaj", "Edytuj", "Usuń" oraz "Cofnij", pozwalający na powrót do menu głównego.



The screenshot shows a window titled 'Zarządzanie Strażakami'. Inside, there is a table titled 'Lista strażaków'. The table has five columns: 'Imię', 'Nazwisko', 'Stopień', 'Data Wstąpienia', and 'Ważność Badań'. Below the table, there are four red buttons: 'Dodaj', 'Edytuj', 'Usuń', and 'Cofnij'.

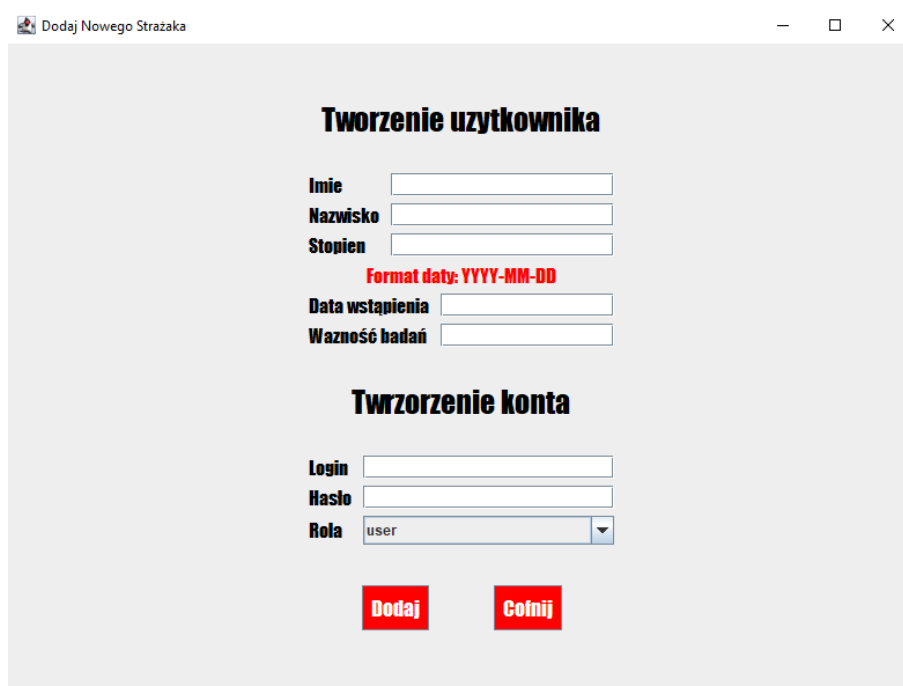
Imię	Nazwisko	Stopień	Data Wstąpienia	Ważność Badań
Jan	Kowalski	Starszy ogniomistrz	2019-12-10	2026-05-20
Adam	Nowak	Naczelnik	2015-11-12	2025-11-10

Rys. 4.4. Główny widok modułu zarządzania strażakami.

Dane do tabeli są dynamicznie ładowane przy tworzeniu okna za pomocą metody 'ZaladujStrazacy()'. Metoda ta wykonuje zapytanie SQL w celu pobrania z bazy danych imienia, nazwiska, stopnia, daty wstąpienia oraz ważności badań lekarskich każdego strażaka. Pobrane dane są następnie dodawane jako wiersze do obiektu 'DefaultTableModel', który stanowi model dla komponentu 'JTable'. W tym widoku, w celu ochrony danych, komórki tabeli są domyślnie nieedytowalne, co jest zdefiniowane przez nadpisanie metody 'isCellEditable'.

Dodawanie Nowego Strażaka

Po naciśnięciu przycisku "Dodaj", użytkownik jest przenoszony do formularza tworzenia nowego strażaka i powiązanego z nim konta użytkownika (Rys. 4.5). Formularz jest podzielony na sekcje dotyczące danych osobowych oraz danych logowania.



Rys. 4.5. Formularz dodawania nowego strażaka.

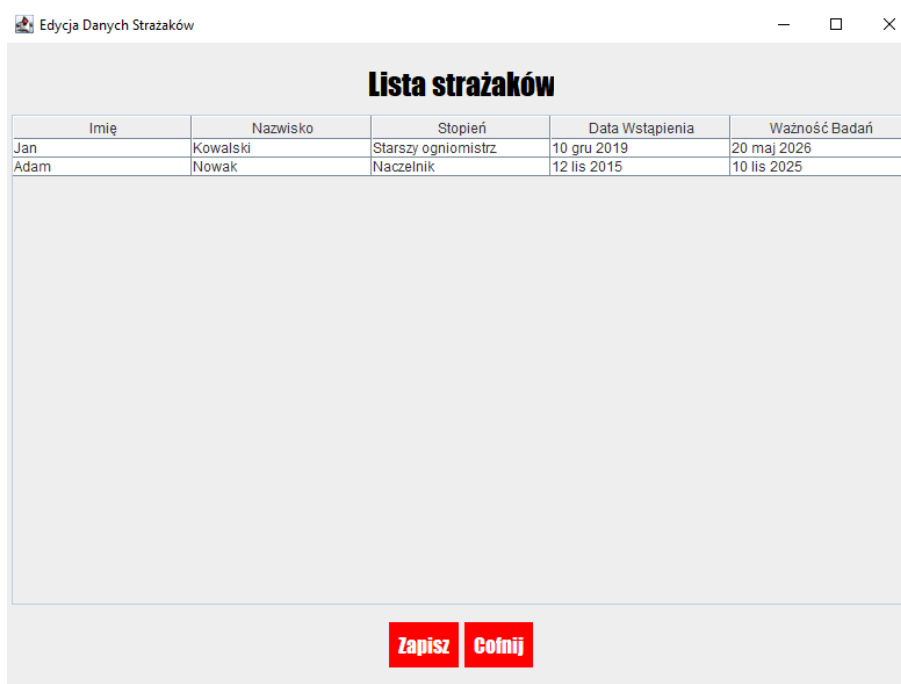
Proces zapisu nowego strażaka jest operacją krytyczną, dlatego został zaimplementowany w ramach transakcji bazodanowej, aby zapewnić integralność i spójność danych.

1. Po odczytaniu danych z formularza, system weryfikuje, czy kluczowe pola (imię, nazwisko, login, hasło) nie są puste.
2. Nawiązywane jest połączenie z bazą, a tryb automatycznego zatwierdzania transakcji (`AutoCommit`) zostaje wyłączony (`conn.setAutoCommit(false)`).
3. Wykonywane jest pierwsze zapytanie `INSERT` do tabeli `strazacy`. Użycie flagi `Statement.RETURN_GENERATED_KEYS` pozwala na natychmiastowe pobranie automatycznie wygenerowanego przez bazę klucza głównego (`strazak_id`) dla nowo utworzonego rekordu.
4. Pobrany identyfikator (`strazakId`) jest następnie używany w drugim zapytaniu `INSERT`, które tworzy konto w tabeli `uzytkownicy` i wiąże je z nowo dodanym strażakiem poprzez klucz obcy.
5. Jeśli obie operacje zapisu powiodą się, transakcja jest finalizowana za pomocą `conn.commit()`. W przypadku wystąpienia błędu na dowolnym etapie, wszystkie zmiany wprowadzone w ramach bieżącej transakcji są wycofywane przez wywołanie `conn.rollback()`.
6. System jest również przygotowany na obsługę wyjątków związanych z nieprawidłowym formatem daty, informując o tym użytkownika w czytelny sposób.

Edycja Danych Strażaka

Okno edycji (Rys. 4.6), dostępne po kliknięciu przycisku "Edytuj", dziedziczy po głównym oknie 'Strażacy', ale wykorzystuje specjalnie przygotowany model tabeli – 'EditableStrazacyTableModel'. Ten model, w przeciwieństwie do domyślnego, pozwala na bezpośrednią modyfikację danych w komórkach, dzięki

zwracaniu wartości 'true' w metodzie 'isCellEditable'. Zmiany są zapisywane w bazie danych po kliknięciu przycisku "Zapisz". Metoda 'zapiszZmiany()' iteruje przez wszystkie wiersze tabeli i dla każdego z nich przygotowuje zapytanie 'UPDATE'. W celu optymalizacji komunikacji z bazą danych, wszystkie zapytania 'UPDATE' są grupowane w jedną partię ('batch') za pomocą 'stmt.addBatch()' i wykonywane jednocześnie przez 'stmt.executeBatch()'. Zaimplementowano również pomocniczą metodę 'convertToSqlDate' do bezpiecznej konwersji obiektów daty, co uniezależnia system od formatu wprowadzanego przez użytkownika.



The screenshot shows a window titled "Edycja Danych Strażaków" with a close button (X) in the top right corner. The main content area is titled "Lista strażaków" and contains a table with the following data:

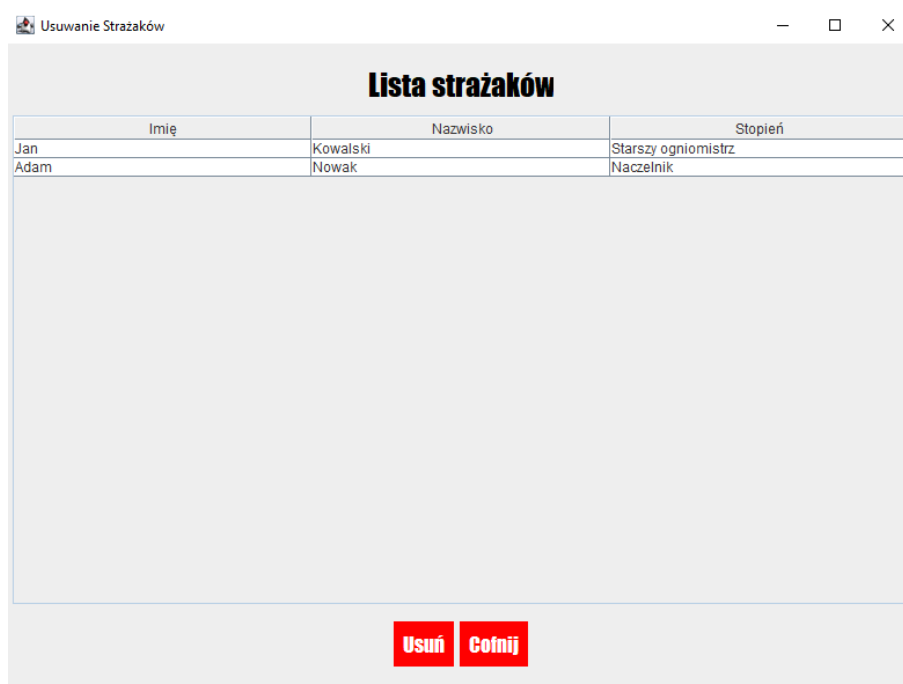
Imię	Nazwisko	Stopień	Data Wstąpienia	Ważność Badań
Jan	Kowalski	Starszy ogniomistrz	10 gru 2019	20 maj 2026
Adam	Nowak	Naczelnik	12 lis 2015	10 lis 2025

Below the table, there are two red buttons: "Zapisz" (Save) and "Cofnij" (Cancel).

Rys. 4.6. Widok edycji danych strażaków.

Usuwanie Strażaka

Okno usuwania (Rys. 4.7) również dziedziczy po klasie 'Strazacy', ale modyfikuje panel z przyciskami oraz logikę ładowania tabeli, aby dostosować ją do operacji usuwania.



Rys. 4.7. Widok usuwania strażaków.

Proces usuwania przebiega następująco:

1. Użytkownik musi najpierw zaznaczyć wiersz w tabeli, który ma zostać usunięty. Jeśli żaden wiersz nie jest zaznaczony, system wyświetla stosowne ostrzeżenie.
2. W celu uniknięcia przypadkowego usunięcia danych, wyświetlane jest okno dialogowe z prośbą o potwierdzenie operacji (`JOptionPane.showConfirmDialog`).
3. Do identyfikacji rekordu do usunięcia używany jest dedykowany model tabeli, `DeletableStrazacyTableModel`. Przechowuje on wewnętrzną listę identyfikatorów (`strazak_id`) odpowiadających każdemu wierszowi, co pozwala jednoznacznie zidentyfikować rekord w bazie danych, nawet jeśli tabela nie wyświetla kolumny ID.
4. Wykonywane jest zapytanie `DELETE FROM strazacy WHERE strazak_id = ?`.
5. Po pomyślnym usunięciu rekordu z bazy, odpowiedni wiersz jest również usuwany z modelu tabeli (`tableModel.removeRow(selectedRow)`), co natychmiastowo odświeża widok interfejsu. System obsługuje również błędy związane z naruszeniem więzów integralności (np. gdy próbuje się usunąć strażaka, który jest przypisany do istniejącej interwencji), informując o tym użytkownika.

4.3. Moduł Zarządzania Interwencjami

Moduł zarządzania interwencjami jest kolejnym kluczowym elementem systemu, umożliwiającym ewidencjonowanie i zarządzanie akcjami ratowniczo-gaśniczymi. Dostęp do modułu jest możliwy z poziomu menu głównej aplikacji, poprzez dedykowany przycisk „Interwencje”. Struktura modułu, podobnie jak w przypadku zarządzania strażakami, opiera się na operacjach CRUD (Create, Read, Update, Delete).

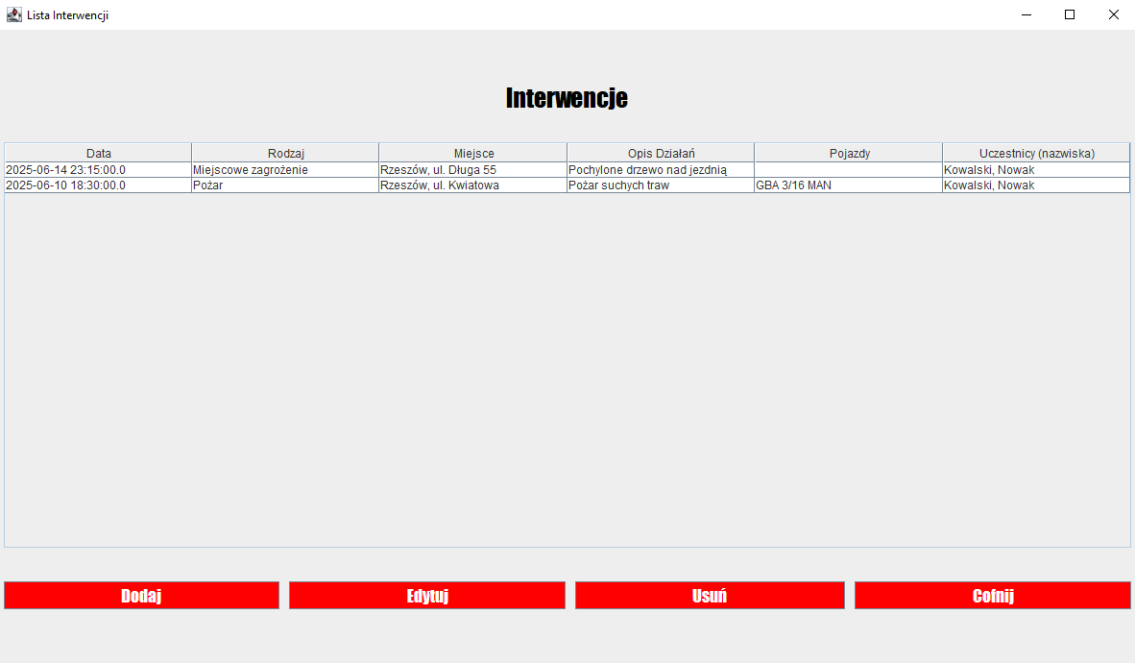
Widok Główny Modułu

Główny widok modułu (Rys. 4.8) prezentuje w formie tabeli listę wszystkich zarejestrowanych interwencji. Umożliwia to szybki przegląd kluczowych informacji o zdarzeniach.

Tabela zawiera następujące kolumny:

- **Data:** Dokładna data i czas rozpoczęcia interwencji.
- **Rodzaj:** Kategoria zdarzenia, np. „Miejscowe zagrożenie” lub „Pożar”.
- **Miejsce:** Adres lub lokalizacja, w której miała miejsce interwencja.
- **Opis Działań:** Zwięzły opis podjętych czynności ratowniczych.
- **Pojazdy:** Lista pojazdów zadysponowanych do zdarzenia.
- **Uczestnicy (nazwiska):** Lista strażaków biorących udział w akcji.

Na dole okna znajdują się przyciski nawigacyjne: „Dodaj”, „Edytuj”, „Usuń” oraz „Cofnij”, które pozwalają na wykonanie odpowiednich operacji lub powrót do menu głównego.



Rys. 4.8. Główny widok modułu zarządzania interwencjami.

Dodawanie Nowej Interwencji

Po wybraniu opcji „Dodaj”, użytkownik jest kierowany do formularza (Rys. 4.9), który umożliwia wprowadzenie danych nowego zdarzenia. Formularz zawiera pola tekstowe odpowiadające kolumnom z

widoku głównego. Zwraca uwagę wyraźna informacja dla użytkownika o wymaganym formacie daty i godziny: YYYY-MM-DD HH-MM-SS. Logika działania formularza, analogicznie do modułu strażaków, obejmuje walidację wprowadzonych danych (sprawdzenie, czy kluczowe pola nie są puste) oraz wykonanie zapytania INSERT do bazy danych po naciśnięciu przycisku „Dodaj”.



Dodaj nową interwencję

Dodaj interwencje

Format daty i godziny: YYYY-MM-DD HH-MM-SS

Data i godzina

Rodzaj

Miejsce

Opis

Pojazdy

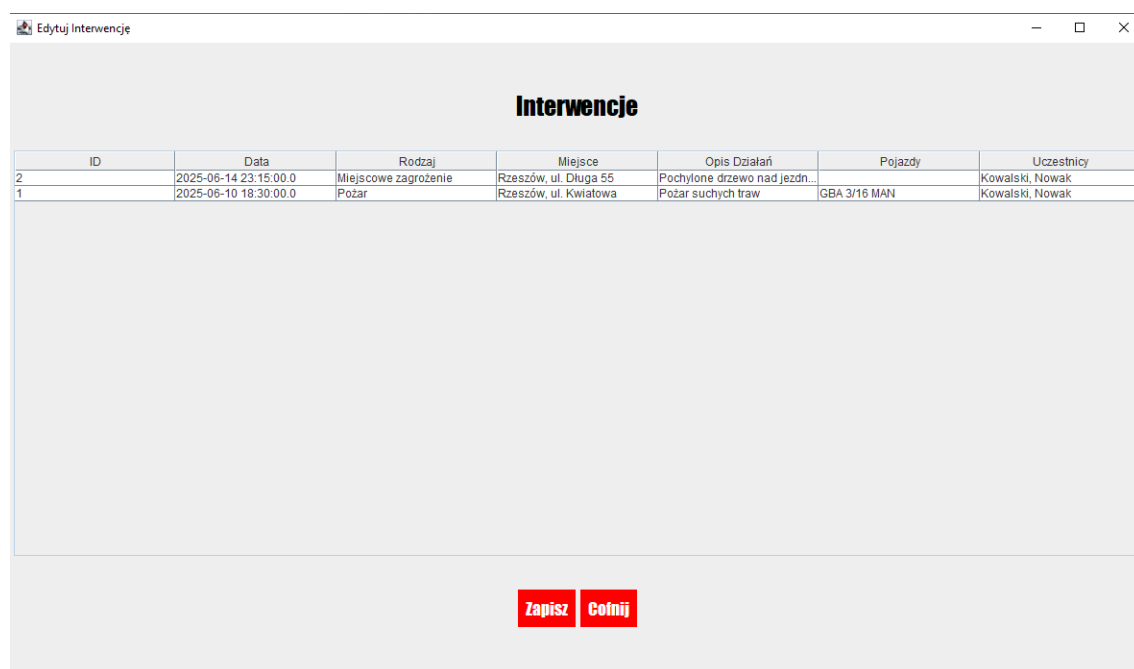
Uczestnicy

Dodaj **Cofnij**

Rys. 4.9. Formularz dodawania nowej interwencji.

Edycja Interwencji

Funkcjonalność edycji pozwala na modyfikację istniejących wpisów. Interfejs (Rys. 4.10) prezentuje dane w tabeli, która pozwala na bezpośrednią zmianę wartości w komórkach. W widoku tym pojawia się dodatkowa kolumna „ID”, która jednoznacznie identyfikuje rekord w bazie danych. Po dokonaniu pożądanych zmian, użytkownik zatwierdza je przyciskiem „Zapisz”. Działanie to inicjuje wykonanie serii zapytań UPDATE w bazie danych w celu uaktualnienia zmodyfikowanych rekordów.



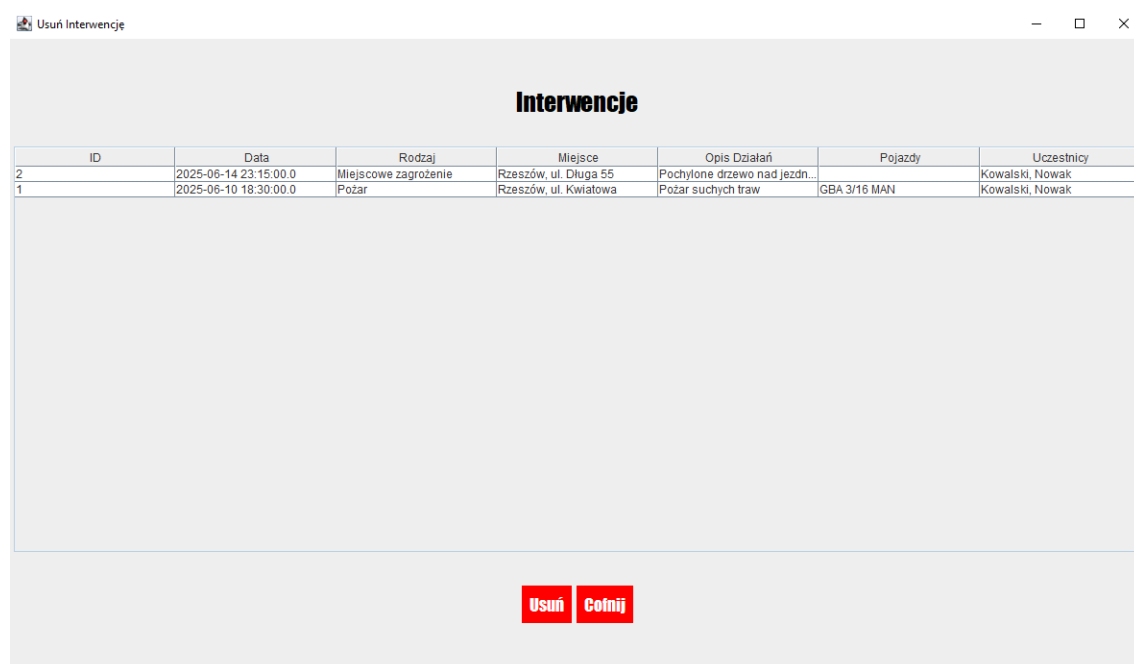
ID	Data	Rodzaj	Miejsce	Opis Działania	Pojazdy	Uczestnicy
2	2025-06-14 23:15:00.0	Miejscowe zagrożenie	Rzeszów, ul. Długa 55	Pochylone drzewo nad jezdni...		Kowalski, Nowak
1	2025-06-10 18:30:00.0	Pożar	Rzeszów, ul. Kwiatowa	Pożar suchych traw	GBA 3/16 MAN	Kowalski, Nowak

Zapisz **Cofnij**

Rys. 4.10. Widok edycji interwencji.

Usuwanie Interwencji

System umożliwia również trwałe usuwanie wpisów o interwencjach. W dedykowanym do tego widoku (Rys. 4.11) użytkownik musi najpierw zaznaczyć wiersz w tabeli, który chce usunąć, a następnie kliknąć przycisk „Usuń”. Analogicznie do pozostałych modułów, operacja ta jest zabezpieczona oknem dialogowym z prośbą o potwierdzenie, aby zapobiec przypadkowemu usunięciu danych. Po potwierdzeniu, wykonywane jest zapytanie DELETE na rekordzie o odpowiednim identyfikatorze (ID).



ID	Data	Rodzaj	Miejsce	Opis Działania	Pojazdy	Uczestnicy
2	2025-06-14 23:15:00.0	Miejscowe zagrożenie	Rzeszów, ul. Długa 55	Pochylone drzewo nad jezdni...		Kowalski, Nowak
1	2025-06-10 18:30:00.0	Pożar	Rzeszów, ul. Kwiatowa	Pożar suchych traw	GBA 3/16 MAN	Kowalski, Nowak

Usuń **Cofnij**

Rys. 4.11. Widok usuwania interwencji.

4.4. Moduł Zarządzania Pojazdami

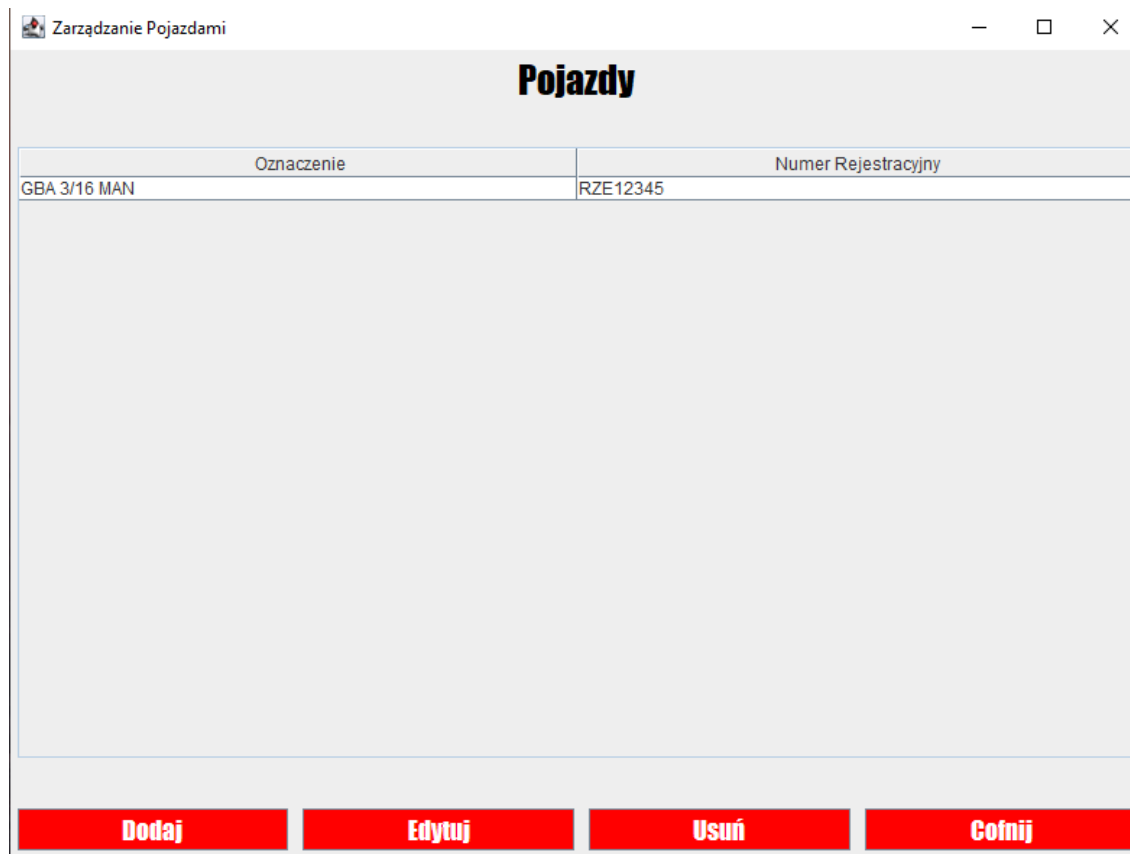
Ostatnim z kluczowych modułów aplikacji jest system ewidencji pojazdów. Umożliwia on katalogowanie oraz zarządzanie flotą pojazdów dostępnych w jednostce. Dostęp do tego modułu, podobnie jak do pozostałych, jest realizowany z poziomu menu głównego, gdzie znajduje się przycisk „Pojazdy”. Jego kliknięcie powoduje zamknięcie okna menu i otwarcie interfejsu zarządzania pojazdami.

Widok Główny Modułu

Główny widok modułu (Rys. 4.12) przedstawia listę wszystkich pojazdów wprowadzonych do systemu. Dane prezentowane są w tabeli, która zawiera dwie kluczowe kolumny:

- **Oznaczenie:** Taktyczne lub zwyczajowe oznaczenie pojazdu (np. GBA 3/16 MAN).
- **Numer Rejestracyjny:** Oficjalny numer rejestracyjny pojazdu.

Interfejs ten stanowi centralny punkt do przeglądania floty oraz inicjowania operacji dodawania, edycji lub usuwania pojazdów za pomocą przycisków umieszczonych na dole okna.



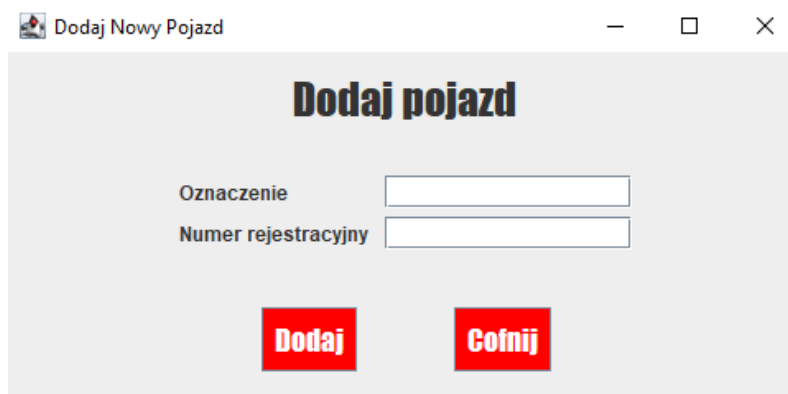
The screenshot shows a window titled "Zarządzanie Pojazdami" with a close button (X) in the top right corner. The main content area is titled "Pojazdy" and contains a table with two columns: "Oznaczenie" and "Numer Rejestracyjny". The table has one row with the values "GBA 3/16 MAN" and "RZE12345". Below the table, there are four red buttons: "Dodaj", "Edytuj", "Usuń", and "Cofnij".

Oznaczenie	Numer Rejestracyjny
GBA 3/16 MAN	RZE12345

Rys. 4.12. Główny widok modułu zarządzania pojazdami.

Dodawanie Nowego Pojazdu

Dodawanie nowego pojazdu do ewidencji odbywa się za pomocą prostego formularza (Rys. 4.13). Formularz zawiera dwa pola tekstowe: „Oznaczenie” i „Numer rejestracyjny”. Po ich uzupełnieniu, naciśnięcie przycisku „Dodaj” powoduje zapisanie nowego rekordu w bazie danych. Proces ten, podobnie jak w przypadku dodawania strażaka, jest poprzedzony walidacją sprawdzającą, czy pola nie są puste.



Dodaj pojazd

Oznaczenie

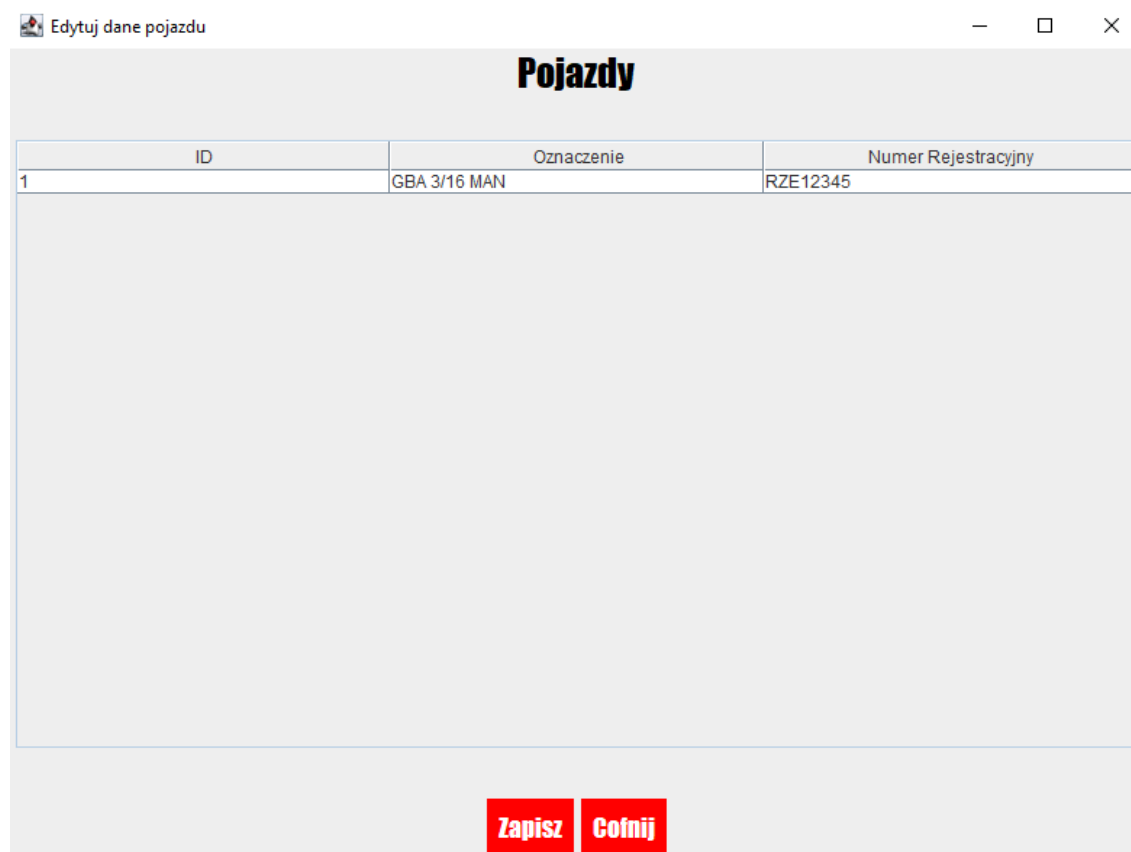
Numer rejestracyjny

Dodaj **Cofnij**

Rys. 4.13. Formularz dodawania nowego pojazdu.

Edycja Danych Pojazdu

Modyfikacja istniejących danych o pojeździe jest możliwa w dedykowanym oknie edycji (Rys. 4.14). Interfejs ten prezentuje listę pojazdów w tabeli, w której można bezpośrednio edytować dane w komórkach. W celu jednoznacznej identyfikacji modyfikowanego rekordu, w widoku tym widoczna jest dodatkowa kolumna „ID”. Po wprowadzeniu zmian, użytkownik zatwierdza je przyciskiem „Zapisz”.



Pojazdy

ID	Oznaczenie	Numer Rejestracyjny
1	GBA 3/16 MAN	RZE12345

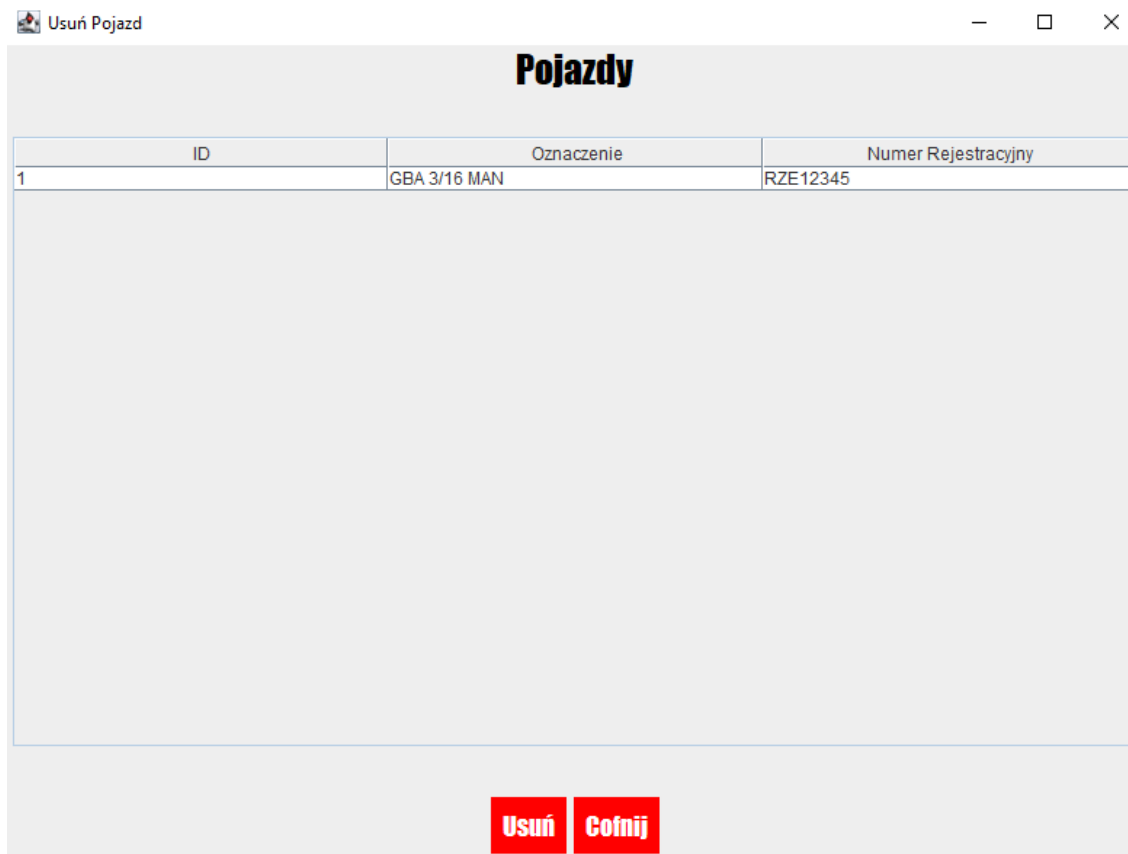
Zapisz **Cofnij**

Rys. 4.14. Widok edycji danych pojazdu.

Usuwanie Pojazdu

System pozwala również na usuwanie pojazdów z ewidencji. W tym celu użytkownik musi przejść do okna usuwania (Rys. 4.15), zaznaczyć wiersz odpowiadający pojazdowi przeznaczonemu do usunięcia, a

następnie kliknąć przycisk „Usuń”. Zgodnie z przyjętym w aplikacji standardem, operacja ta jest chroniona oknem dialogowym z prośbą o potwierdzenie, aby uniknąć omyłkowego skasowania danych. Usunięcie może być zablokowane, jeśli dany pojazd jest przypisany do istniejącej interwencji, co chroni integralność danych w systemie.



ID	Oznaczenie	Numer Rejestracyjny
1	GBA 3/16 MAN	RZE12345

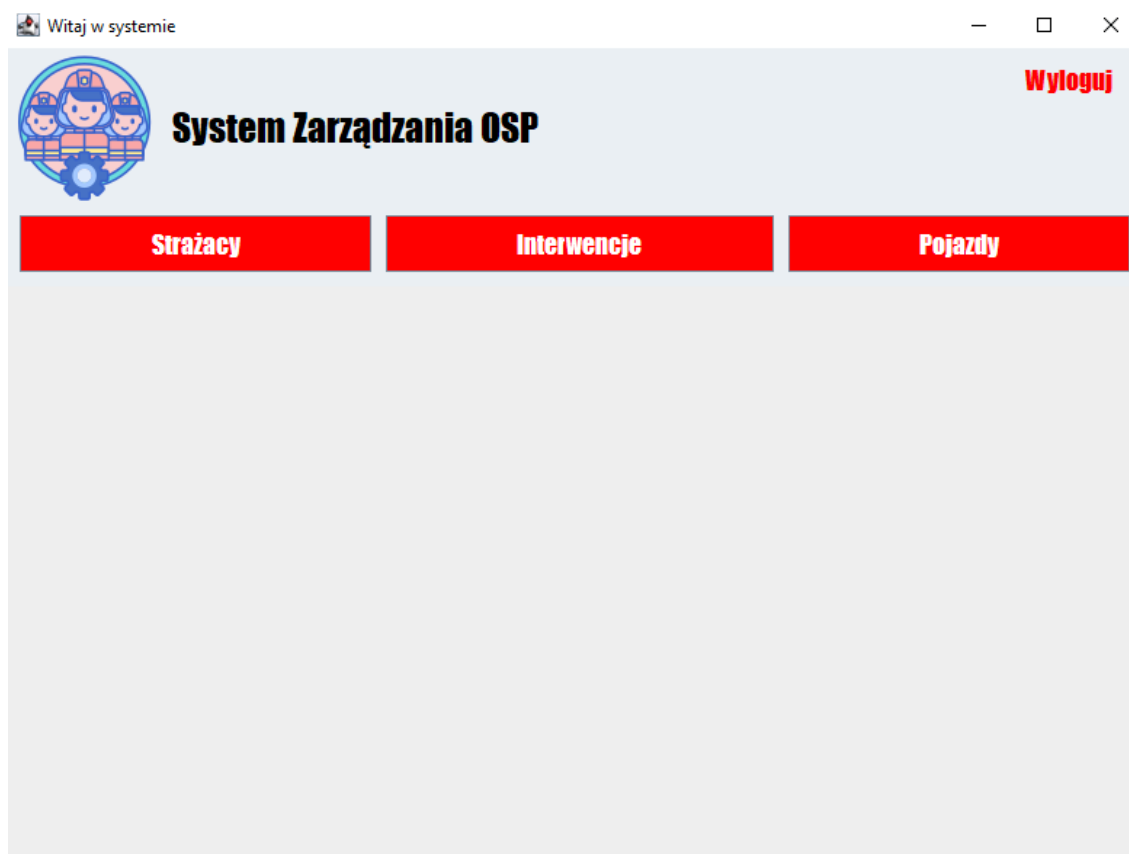
Rys. 4.15. Widok usuwania pojazdu z ewidencji.

4.5. Panel Użytkownika o Ograniczonych Uprawnieniach

System Zarządzania OSP został zaprojektowany z uwzględnieniem dwóch podstawowych ról użytkowników: administratora oraz użytkownika standardowego. Rola jest przypisywana w momencie tworzenia konta i weryfikowana podczas procesu logowania. W przeciwieństwie do administratora, który posiada pełne uprawnienia do zarządzania danymi (CRUD), użytkownik standardowy ma dostęp do uproszczonej wersji aplikacji, skoncentrowanej na przeglądaniu informacji. Ten rozdział opisuje wygląd i funkcjonalność panelu przeznaczonego dla tej właśnie roli.

Menu Główne Użytkownika

Po pomyślnym zalogowaniu na konto bez uprawnień administratora, użytkownik jest kierowany do menu głównego, przedstawionego na (Rys. 4.16). Interfejs ten jest wizualnie tożsamy z panelem administratora i również oferuje nawigację do modułów „Strażacy”, „Interwencje” oraz „Pojazdy”. Kluczowa różnica polega na tym, że przyciski nawigacyjne w tym panelu kierują do specjalnych, uproszczonych wersji okien, które mają charakter wyłącznie informacyjny. Przycisk „Wyloguj” pozwala na bezpieczne zamknięcie sesji i powrót do ekranu logowania.



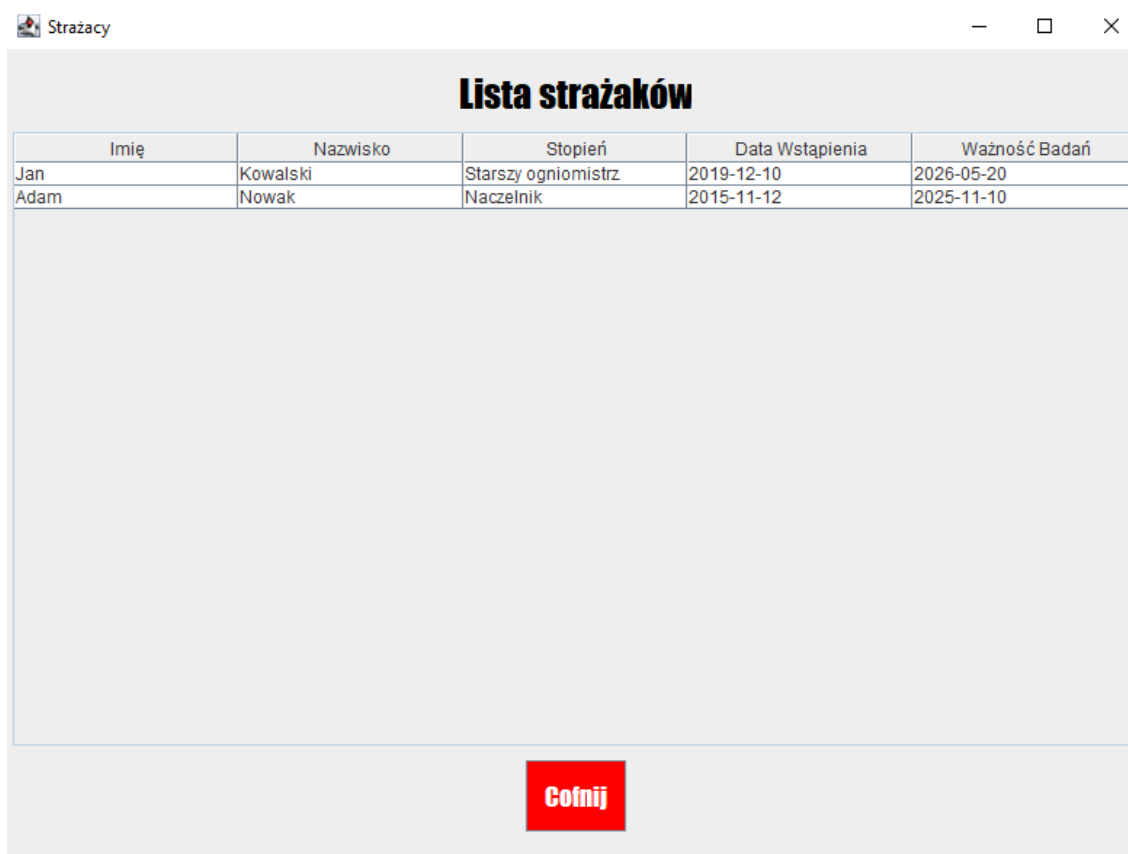
Rys. 4.16. Główne menu w panelu użytkownika standardowego.

Widoki Danych w Trybie Tylko do Odczytu

Głównym ograniczeniem panelu użytkownika jest brak możliwości modyfikacji, dodawania lub usuwania jakichkolwiek danych. Wszystkie moduły prezentują informacje w trybie „tylko do odczytu”, a jedyną dostępną akcją jest powrót do menu głównego za pomocą przycisku „Cofnij”.

Lista Strażaków

Widok listy strażaków (Rys. 4.17) pozwala na przeglądanie podstawowych danych o wszystkich członkach jednostki. Dane są pobierane z bazy za pomocą prostego zapytania `SELECT` i wyświetlane w tabeli. Zgodnie z założeniem o braku możliwości edycji, komórki tabeli są zablokowane, co zostało zaimplementowane poprzez nadpisanie metody `isCellEditable` w modelu tabeli.



Imię	Nazwisko	Stopień	Data Wstąpienia	Ważność Badań
Jan	Kowalski	Starszy ogniomistrz	2019-12-10	2026-05-20
Adam	Nowak	Naczelnik	2015-11-12	2025-11-10

Cofnij

Rys. 4.17. Widok listy strażaków w panelu użytkownika.

Ewidencja Pojazdów

Analogicznie, moduł pojazdów (Rys. 4.18) udostępnia jedynie listę pojazdów wraz z ich oznaczeniem i numerem rejestracyjnym. Tabela jest nieedytowalna, a jedynym interaktywnym elementem jest przycisk „Cofnij”, który zamyka bieżące okno i otwiera menu główne.

Pojazdy

Pojazdy

Oznaczenie	Numer Rejestracyjny
GBA 3/16 MAN	RZE12345

Cofnij

Rys. 4.18. Widok ewidencji pojazdów w panelu użytkownika.

Historia Interwencji

Widok historii interwencji (Rys. 4.19) prezentuje on kompleksowe podsumowanie każdej akcji, łącząc dane z wielu tabel.

Interwencje

Interwencje

Data	Rodzaj	Miejsce	Opis Działania	Pojazdy	Uczestnicy (nazwiska)
2025-06-14 23:15:00.0	Miejscowe zagrożenie	Rzeszów, ul. Długa 55	Pochylone drzewo nad jezdnią		Kowalski, Nowak
2025-06-10 18:30:00.0	Pożar	Rzeszów, ul. Kwiatowa	Pożar suchych traw	GBA 3/16 MAN	Kowalski, Nowak

Cofnij

Rys. 4.19. Widok historii interwencji w panelu użytkownika.

Do przygotowania tych danych wykorzystywane jest rozbudowane zapytanie SQL, które łączy dane z tabel `interwencje`, `strazacy` oraz `pojazdy` za pośrednictwem tabel łączących. W celu zwięzłego przedstawienia listy uczestników i pojazdów w pojedynczych komórkach tabeli, zapytanie wykorzystuje funkcję agregującą `GROUP_CONCAT`.

Listing 4.1. Fragment zapytania SQL agregującego dane dla widoku interwencji.

```
1 SELECT
2     i.data_zdarzenia,
3     i.rodzaj_zdarzenia,
4     i.miejsce_zdarzenia,
5     i.opis_dzialan,
6     GROUP_CONCAT(DISTINCT p.oznaczenie SEPARATOR ', ') AS pojazdy,
7     GROUP_CONCAT(DISTINCT s.nazwisko SEPARATOR ', ') AS strazacy
8 FROM
9     interwencje i
10 LEFT JOIN ...
```

Podobnie jak w pozostałych modułach, dane są prezentowane w trybie tylko do odczytu, a jedyną możliwą akcją jest powrót do menu. Takie podejście zapewnia integralność danych, jednocześnie udostępniając wszystkim użytkownikom informacje operacyjne.

5. Aspekty Techniczne i Zarządzanie Projektem

Problemy Napotkane Podczas Realizacji

Zapewnienie Spójności Danych przy Zapisie Złożonym

Problem Jednym z pierwszych wyzwań było dodawanie nowego strażaka do systemu. Operacja ta wymagała wykonania dwóch oddzielnych zapytań `INSERT`: jednego do tabeli `strazacy` i drugiego do tabeli `uzytkownicy`. Istniało ryzyko, że po pomyślnym dodaniu rekordu do pierwszej tabeli, zapis do drugiej mógłby się nie udać (np. z powodu błędu połączenia). Pozostawiłoby to bazę danych w niespójnym stanie — istniałby strażak bez powiązanego z nim konta użytkownika.

Rozwiązanie Aby zagwarantować atomowość tej operacji, zaimplementowano mechanizm **transakcji bazodanowych**. Przed wykonaniem zapytań, tryb automatycznego zatwierdzania (`AutoCommit`) jest wyłączany. Oba zapytania `INSERT` są wykonywane w ramach jednej transakcji. Jeśli oba zakończą się sukcesem, zmiany są trwale zapisywane w bazie za pomocą komendy `commit()`. W przypadku wystąpienia błędu na dowolnym etapie, wszystkie dotychczasowe zmiany są wycofywane za pomocą `rollback()`, przywracając bazę do stanu sprzed rozpoczęcia operacji.

Agregacja Danych z Wielu Tabel w Jednym Widoku

Problem Wyświetlenie listy interwencji w czytelny sposób wymagało zebrania danych z wielu tabel. Pojedyncza interwencja jest powiązana z wieloma strażakami i wieloma pojazdami poprzez tabele łączące (`interwencja_uczestnicy` i `interwencja_pojazdy`). Wyzwaniem było przedstawienie listy nazwisk uczestników oraz oznaczeń pojazdów w pojedynczych komórkach głównej tabeli interwencji.

Rozwiązanie Problem ten rozwiązano na poziomie zapytania SQL, wykorzystując funkcję agregującą `GROUP_CONCAT` dostępną w MySQL. Zapytanie zostało skonstruowane w taki sposób, aby grupowało wyniki po identyfikatorze interwencji, a następnie łączyło nazwiska strażaków oraz oznaczenia pojazdów w pojedyncze ciągi znaków, oddzielone przecinkami. Dzięki temu cała logika agregacji danych odbywa się po stronie bazy danych, a aplikacja otrzymuje gotowe do wyświetlenia, sformatowane informacje.

Obsługa Edycji Dat w Tabeli

Problem W module edycji danych strażaka umożliwiono bezpośrednią modyfikację dat (np. ważności badań) w komórkach komponentu `JTable`. Powstał problem z obsługą formatu danych — po zakończeniu edycji, wartość w komórce mogła być obiektem typu `String` lub `java.util.Date`, podczas gdy sterownik JDBC wymagał obiektu typu `java.sql.Date` do poprawnego zapisu w bazie. Bezpośrednia konwersja groziła błędem `IllegalArgumentException`.

Rozwiązanie W klasie `StrazacyEdytuj` zaimplementowano dedykowaną, prywatną metodę pomocniczą `convertToSqlDate(Object dateObj)`. Metoda ta w bezpieczny sposób konwertuje otrzymany obiekt na format `java.sql.Date`. Sprawdza ona typ obiektu wejściowego i obsługuje różne przypadki: jeśli obiekt jest już w poprawnym formacie, jest zwracany bez zmian; jeśli jest to `java.util.Date`, jest konwertowany; jeśli jest to `String`, jest parsowany zgodnie z oczekiwanym

formatem "yyyy-MM-dd". Takie podejście znacząco zwiększyło odporność aplikacji na błędy związane z wprowadzaniem danych przez użytkownika.

GitHub

Jako platforma do hostingu zdalnego repozytorium kodu został wybrany serwis **GitHub**. Pełni on nie tylko rolę kopii zapasowej, ale również stanowi publiczne portfolio projektu. Całość kodu źródłowego aplikacji jest dostępna publicznie pod adresem:

<https://github.com/Kamil0507/ProjektPO>

6. Podsumowanie

Celem pracy było zaprojektowanie i zaimplementowanie w pełni funkcjonalnego, desktopowego Systemu Zarządzania dla jednostki Ochotniczej Straży Pożarnej. Głównym założeniem było stworzenie narzędzia, które usprawni ewidencję kluczowych zasobów i działań, zastępując tradycyjne metody cyfrowym, scentralizowanym rozwiązaniem. Zaimplementowany system stanowi podstawę, którą można w przyszłości rozbudowywać o nowe funkcjonalności. Poniżej przedstawiono kilka potencjalnych kierunków dalszego rozwoju:

- **Moduł finansowy i magazynowy:** Rozszerzenie systemu o możliwość zarządzania budżetem, ewidencjonowania wydatków oraz śledzenia stanu magazynowego sprzętu i umundurowania.
- **Generowanie raportów i statystyk:** Implementacja narzędzi do automatycznego generowania raportów w formacie PDF (np. rocznych sprawozdań z działalności, raportów z interwencji) oraz tworzenia statystyk, które mogłyby wspomóc analizę działań jednostki.
- **Wersja webowa lub mobilna:** Przeniesienie aplikacji na platformę webową lub stworzenie dedykowanej aplikacji mobilnej, co zwiększyłoby dostępność systemu i pozwoliłoby na wprowadzanie danych bezpośrednio w terenie.
- **Kalendarz i harmonogram:** Dodanie modułu kalendarza do planowania szkoleń, zbiórek, przeglądów technicznych sprzętu oraz innych ważnych wydarzeń.
- **System powiadomień:** Wprowadzenie automatycznych powiadomień, np. o zbliżającym się terminie ważności badań lekarskich strażaków lub przeglądów technicznych pojazdów.

Realizacja powyższych propozycji mogłaby uczynić system jeszcze bardziej kompleksowym i użytecznym narzędziem, wspierającym OSP w ich codziennej służbie.

Bibliografia

- **JavaStart: JDBC – Podstawy pracy z bazą danych.** Przewodnik wprowadzający interfejs JDBC – sterowniki, połączenia i operacje na MySQL.
<https://javastart.pl/baza-wiedzy/java-ee/jdbc-podstawy-pracy-z-baza-danych> [Dostęp: 14.06.2025]
- **JavaStart: Darmowy kurs Java.** Kurs omawiający instalację JDK, podstawy języka, pętle i programowanie obiektowe.
<https://javastart.pl/baza-wiedzy/darmowy-kurs-java> [Dostęp: 14.06.2025]

Spis rysunków

2.1	Diagram encji-związków (ERD) dla bazy danych szosp.	8
3.1	Diagram Gantta przedstawiający tygodniowy harmonogram prac nad projektem.	10
4.1	Główne okno logowania do systemu.	11
4.2	Komunikaty zwrotne w procesie logowania.	12
4.3	Główne menu aplikacji.	12
4.4	Główny widok modułu zarządzania strażakami.	13
4.5	Formularz dodawania nowego strażaka.	14
4.6	Widok edycji danych strażaków.	15
4.7	Widok usuwania strażaków.	16
4.8	Główny widok modułu zarządzania interwencjami.	17
4.9	Formularz dodawania nowej interwencji.	18
4.10	Widok edycji interwencji.	19
4.11	Widok usuwania interwencji.	19
4.12	Główny widok modułu zarządzania pojazdami.	20
4.13	Formularz dodawania nowego pojazdu.	21
4.14	Widok edycji danych pojazdu.	21
4.15	Widok usuwania pojazdu z ewidencji.	22
4.16	Główne menu w panelu użytkownika standardowego.	23
4.17	Widok listy strażaków w panelu użytkownika.	24
4.18	Widok ewidencji pojazdów w panelu użytkownika.	25
4.19	Widok historii interwencji w panelu użytkownika.	25
6.1	Oświadczenie studenta o samodzielności pracy	32

Oświadczenie studenta o samodzielności pracy

Załącznik nr 2 do Zarządzenia nr 228/2021 Rektora Uniwersytetu Rzeszowskiego z dnia 1 grudnia 2021 roku w sprawie ustalenia procedury antyplagiatowej w Uniwersytecie Rzeszowskim

OŚWIADCZENIE STUDENTA O SAMODZIELNOŚCI PRACY

..... Kamil Antoni Krawczyk
Imię (imiona) i nazwisko studenta

Wydział Nauk Ścisłych i Technicznych

..... Informatyka
Nazwa kierunku

..... 134931
Numer albumu

1. Oświadczam, że moja praca projektowa pt.: Dokumentacja Systemu Zarządzania OSP

- 1) została przygotowana przeze mnie samodzielnie*,
- 2) nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
- 3) nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,
- 4) nie była podstawą otrzymania oceny z innego przedmiotu na uczelni wyższej ani mnie, ani innej osobie.

2. Jednocześnie wyrażam zgodę/nie wyrażam zgody** na udostępnienie mojej pracy projektowej do celów naukowo-badawczych z poszanowaniem przepisów ustawy o prawie autorskim i prawach pokrewnych.

Dawid 14.06.2023 Kamil Krawczyk
(miejscowość, data) (czytelny podpis studenta)

* Uwzględniając merytoryczny wkład prowadzącego przedmiot
** – niepotrzebne skreślić

Rys. 6.1. Oświadczenie studenta o samodzielności pracy