

Strona tytułowa

- Imię nazwisko autorów: Kacper Długosz i Bartłomiej Florek
- Numery albumów: 125111 i 125115
- Temat projektu: Muzeum czołgów
- Nazwę przedmiotu: Bazy danych
- Grupa laboratoryjna: lab 1

Spis treści

Strona tytułowa	1
Opis programu:	2
Wykorzystane technologie:	2
Opis działań	2
Diagram ERD:	3
Opis działania aplikacji	5
Funkcje języka proceduralnego	18
Źródła:	23

Opis programu:

Muzeum Czołgów to aplikacja desktopowa, która umożliwia zarządzanie kolekcją czołgów. Zalogowani użytkownicy mogą przeglądać czołgi w danych muzeach, a także wyświetlać wszystkie szczegóły czołgów takie jak: nazwa, opis, zdjęcie, amunicja, kraj produkcji itp.

Aby uruchomić program należy:

- W pgAdmin4 utworzyć nową bazę, pod nazwa „tanksdb”.
- Zimportować bazę z pliku „BAZA_GIT.sql”, która utworzy wszystkie tabele oraz wprowadzi dane.
- Pobierz kod źródłowy z repozytorium.
- Otwórz projekt w środowisku JavaFX (np. IntelliJ IDEA).
- Skonfiguruj połączenie z bazą danych w pliku konfiguracyjnym „ConnectDB”.
- Uruchom aplikację.

Wykorzystane technologie:

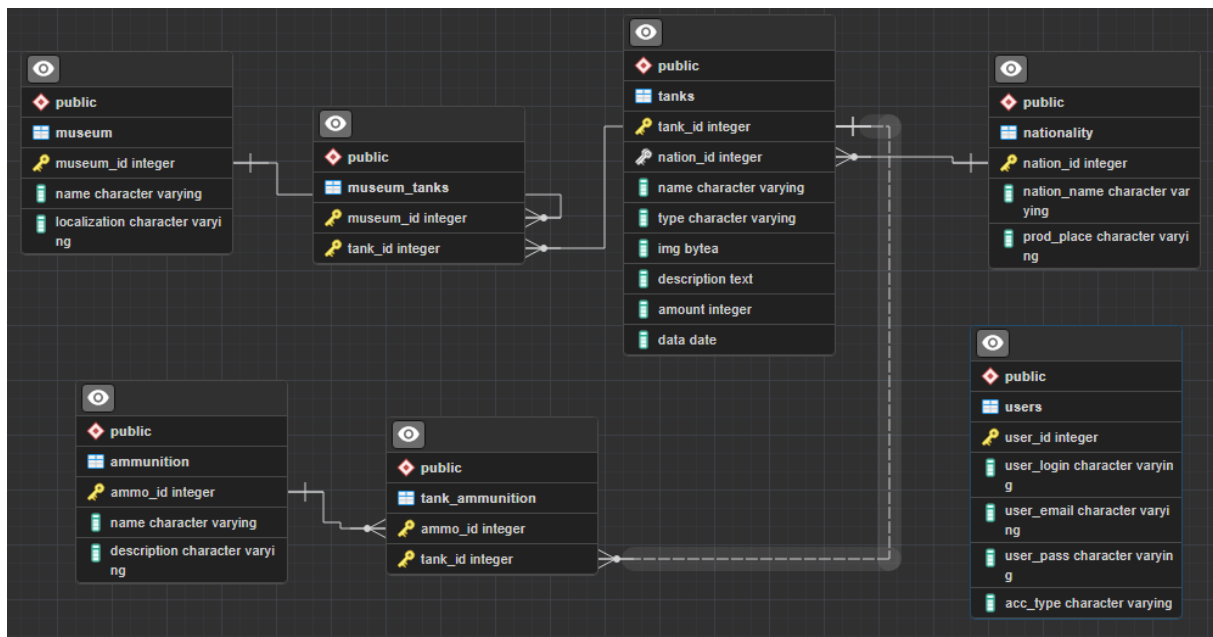
- JavaFX version 21: Do tworzenia interfejsu użytkownika.
- Java SDK version 21.0.2: Do pisania funkcjonalności
- PostgreSQL version 16: Do przechowywania danych o czołgach.
- pgAdmin4 version 7.6: Do zarządzania bazą
- <https://dbdiagram.io/d> - do przygotowania bazy danych

Opis działań

Widoki, funkcje po stronie bazy, wyświetlanie czołgów, dodawanie czołgów, sortowanie, widok dla użytkownika oraz dla admina, baza – Florek

Logowanie, tworzenie nowych użytkowników, walidacje, edycje danych, wyszukiwarki - Długosz

Diagram ERD:



Rysunek 1 - Diagram ERD

Opisy tabel:

Tabela Użytkowników (users):

Kolumny:

- user_id: Liczba całkowita stanowiąca klucz główny identyfikacji użytkownika.
- user_login: Pole typu varchar przechowujące dane logowania użytkownika.
- user_email: Pole typu varchar przechowujące adresy e-mail użytkowników.
- user_pass: Pole typu varchar przechowujące hasła użytkowników.
- acc_type: Pole typu varchar wskazujące typ konta (np. admin, zwykły użytkownik).

Tabela Czołgów (tanks):

Kolumny:

- tank_id: Liczba całkowita stanowiąca klucz główny identyfikacji czołgu.
- nation_id: Liczba całkowita odnosząca się do nation_id z tabeli nationality.
- name: Pole typu varchar zawierające nazwę czołgu.
- type: Pole typu varchar opisujące typ czołgu (np. ciężki, średni, lekki).
- img: Pole typu bytea służące do przechowywania obrazów czołgów.
- description: Pole typu text zawierające opis czołgu.
- amount: Liczba całkowita wskazująca ilość czołgów.

- data: Pole typu date przechowujące daty związane z czołgami

Tabela Nationality:

Kolumny:

- nation_id: Liczba całkowita stanowiąca klucz główny identyfikacji narodowości.
- nation_name: Pole varchar przechowujące nazwę narodu (np. Niemcy, USA).
- prod_place: Pole varchar wskazujące miejsce produkcji (np. lokalizacja fabryki).

Tabela Ammunition:

Kolumny:

- ammo_id: Pole varchar stanowiące klucz główny identyfikacji amunicji.
- name: Pole varchar przechowujące nazwę amunicji.
- description: Pole varchar opisujące amunicję.

Tabela Tank-Ammunition:

Kolumny:

- ammo_id: Liczba całkowita odnosząca się do ammo_id z tabeli ammunition.
- tank_id: Liczba całkowita odnosząca się do tank_id z tabeli tanks.

Tabela Museum Tanks:

Kolumny:

- museum_id: Liczba całkowita odnosząca się do museum_id z tabeli museum.
- tank_id: Liczba całkowita odnosząca się do tank_id z tabeli tanks.

Tabela Museum:

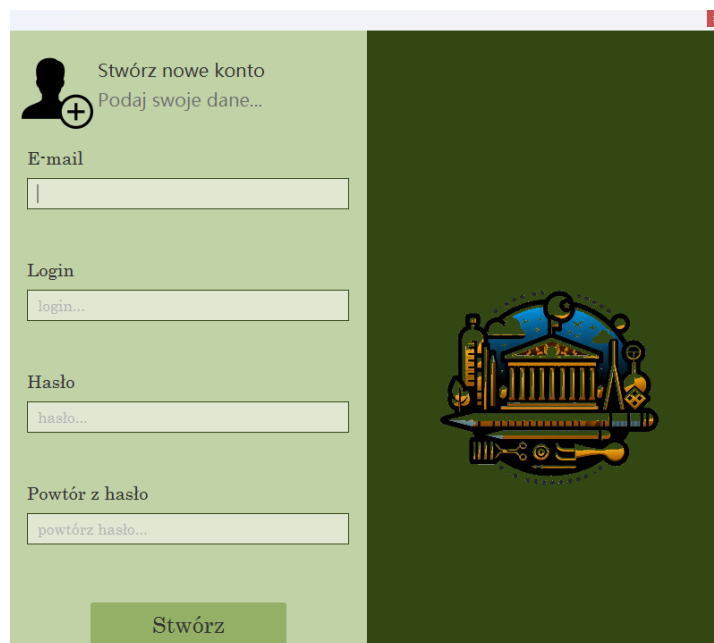
Kolumny:

- museum_id: Liczba całkowita stanowiąca klucz główny identyfikacji muzeum.
- name: Pole varchar przechowujące nazwę muzeum.
- localization: Pole varchar wskazujące lokalizację muzeum.

Opis działania aplikacji:

Rejestracja i logowanie:

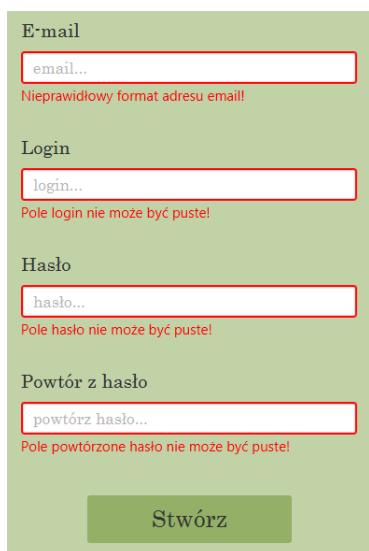
Widok rejestracji:



The registration form is displayed in a window. It has a light green background on the left and a dark green background on the right. On the left, there is a header with a user icon and a plus sign, followed by the text "Stwórz nowe konto" and "Podaj swoje dane...". Below this are four input fields: "E-mail", "Login", "Hasło", and "Powtór z hasło". Each field has a placeholder text: "email...", "login...", "hasło...", and "powtórz hasło..." respectively. At the bottom of the left panel is a green button labeled "Stwórz". On the right panel, there is a colorful illustration of a building with a clock tower and various symbols like a gear, a lightbulb, and a book.

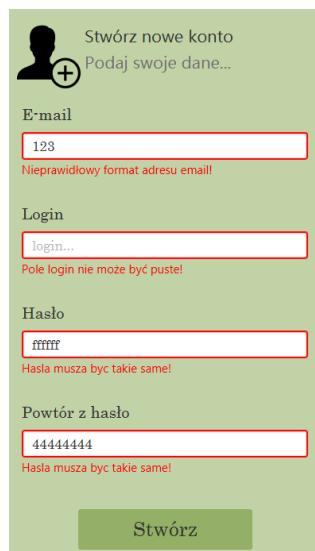
Rysunek 2 - Formularz rejestracji

Program posiada zabezpieczenia przed wprowadzeniem złych wartości i wyświetla odpowiednie komunikaty:



The registration form is shown with all input fields empty. Below each field, there is a red error message: "Nieprawidłowy format adresu email!" for E-mail, "Pole login nie może być puste!" for Login, "Pole hasło nie może być puste!" for Hasło, and "Pole powtórzone hasło nie może być puste!" for Powtór z hasło. The "Stwórz" button is at the bottom.

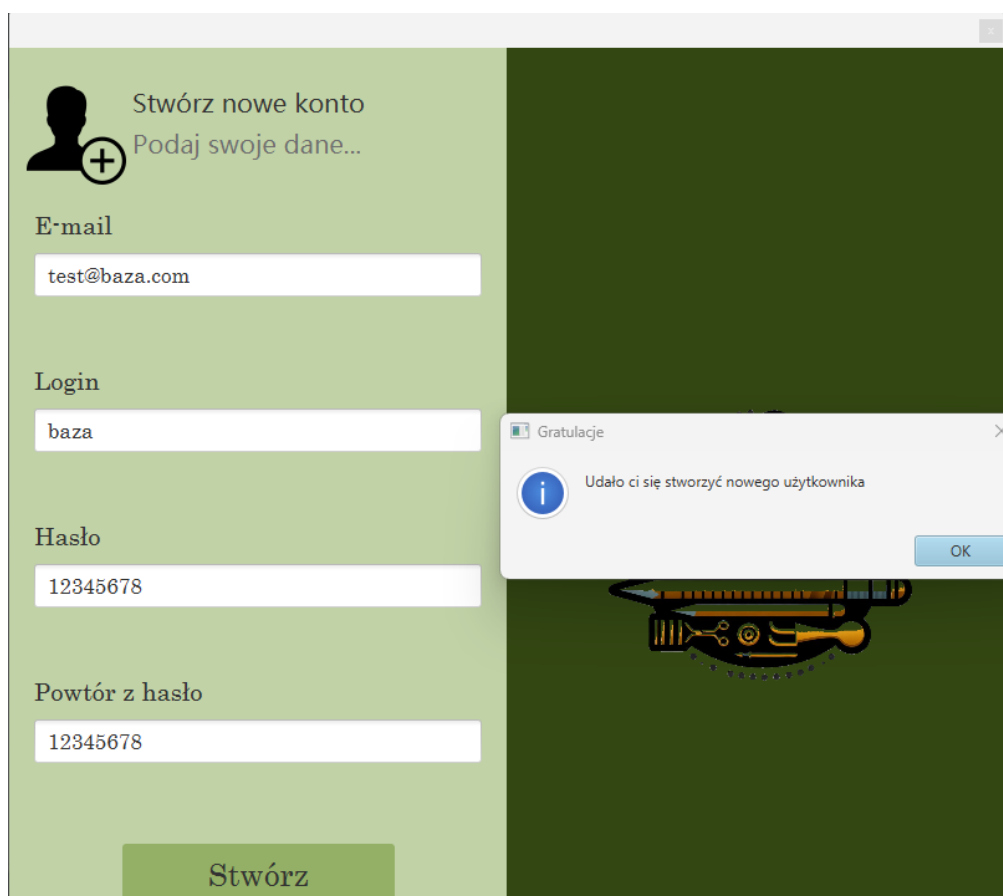
Rysunek 3 - Walidacja (puste pola)



The registration form is shown with invalid values entered in the fields. Below each field, there is a red error message: "Nieprawidłowy format adresu email!" for E-mail (value: 123), "Pole login nie może być puste!" for Login (value: login...), "Hasła muszą być takie same!" for Hasło (value: ffffff), and "Hasła muszą być takie same!" for Powtór z hasło (value: 44444444). The "Stwórz" button is at the bottom.

Rysunek 4 - Walidacja (niepoprawne wartości)

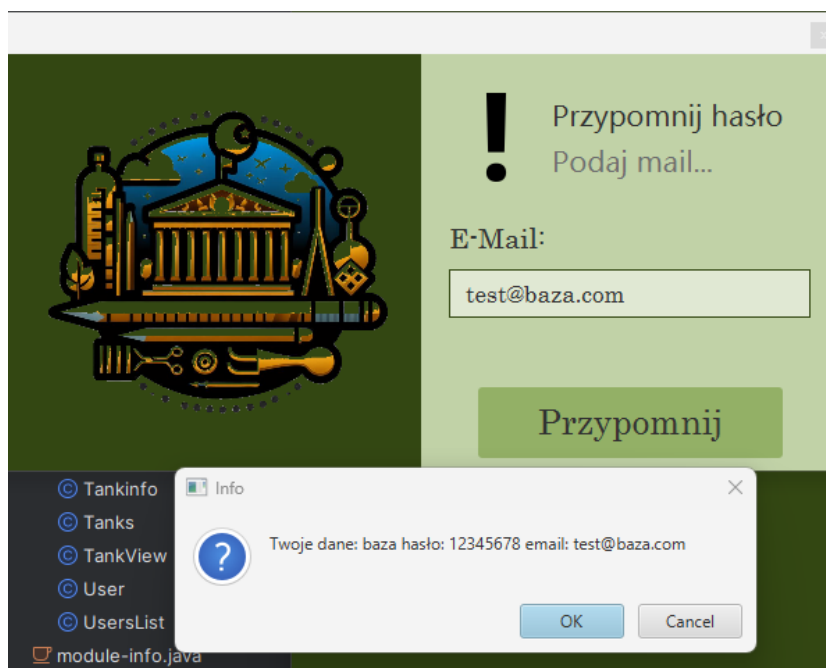
Po wprowadzeniu poprawnych danych zostanie wyświetlony komunikat:



The image shows a registration window with a light green background. On the left, there is a form titled "Stwórz nowe konto" (Create new account) with the subtitle "Podaj swoje dane..." (Provide your data...). The form contains four input fields: "E-mail" with the value "test@baza.com", "Login" with the value "baza", "Hasło" (Password) with the value "12345678", and "Powtór z hasła" (Repeat password) with the value "12345678". At the bottom of the form is a green button labeled "Stwórz". On the right side of the window, there is a dark green area with a stylized illustration of a tank. Overlaid on the right side is a small white dialog box titled "Gratulacje" (Congratulations) with an information icon. The message inside says "Udało ci się stworzyć nowego użytkownika" (You successfully created a new user) and there is an "OK" button at the bottom right of the dialog.

Rysunek 5 - Poprawne dane rejestracji

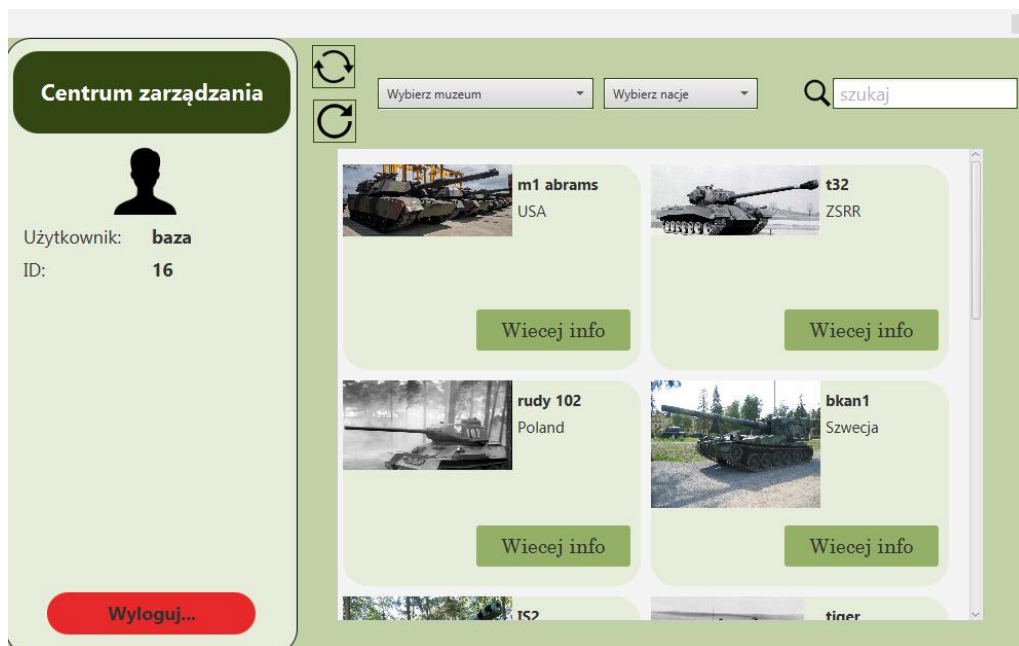
W przypadku zapomnienia hasła jest możliwość jego odzyskania przez wpisanie maila:



The image shows a password recovery window with a light green background. On the left, there is a dark green area with a stylized illustration of a tank. On the right, there is a form titled "Przypomnij hasło" (Remember password) with the subtitle "Podaj mail..." (Provide mail...). The form contains one input field labeled "E-Mail:" with the value "test@baza.com". Below the input field is a green button labeled "Przypomnij". At the bottom left of the window, there is a dark blue sidebar with a list of menu items: "Tankinfo", "Tanks", "TankView", "User", "UsersList", and "module-info.java". Overlaid on the bottom center is a small white dialog box titled "Info" with a question mark icon. The message inside says "Twoje dane: baza hasło: 12345678 email: test@baza.com" and there are "OK" and "Cancel" buttons at the bottom right of the dialog.

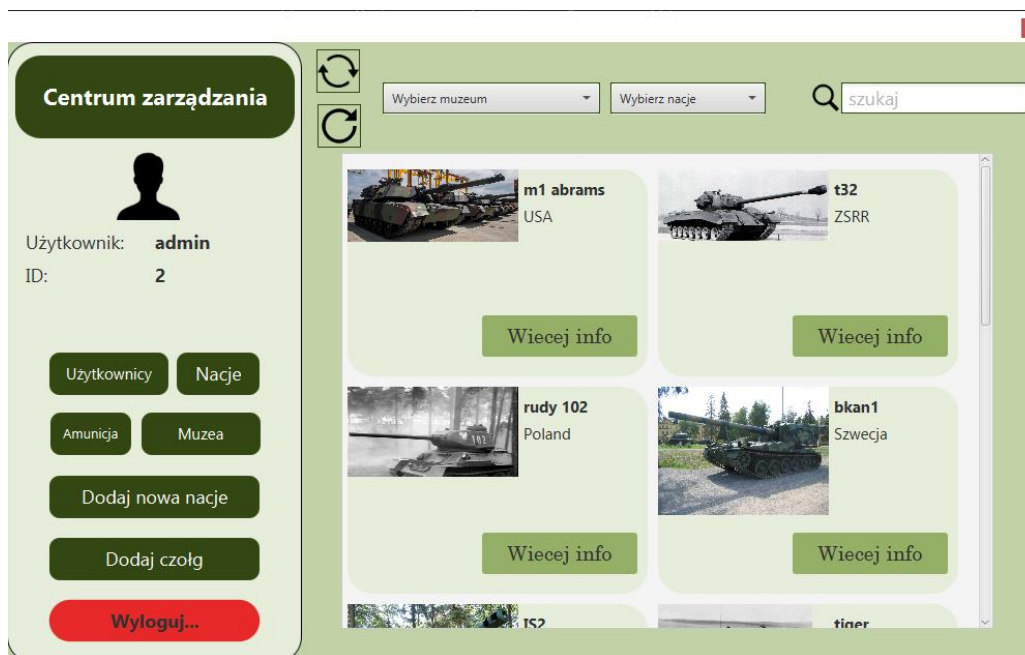
Rysunek 6 - Formularz odzyskiwania hasła

Widok zwykłego użytkownika po zalogowaniu:



Rysunek 7 - Widok użytkownika

Widok admina (admin ma dodatkowa opcje widoczne po lewej stronie panelu):



Rysunek 8 - Widok admina

Opcje po kliknięciu w wybrany czołg:

SZCZEGÓŁOWE INFORMACJE

Nazwa:
m1 abrams

Typ czołgu
ciezki

Kraj pochodzenia
USA

Data
2021-05-06

Ilość
1532


Opis
nowoczesna bestia

Amunicja
ap-1235

Opis amunicji
używana to wiel rzeczu ktore sa...

Miejsce produkcji
Chigaco

Zdjęcie



Edytuj

Usun

Rysunek 9 - Szczegóły danego czołgu

Możliwość edycji danych:

Dodawanie nowego czołgu

Nazwa

m1 abrams

Typ

ciezki

Zdjecie

Dodaj fote

Opis

nowoczesna bestia

Ilość

1532

Data

6.05.2021

Nacja


3: USA

Amunicja

7: ap-1235

Muzeum

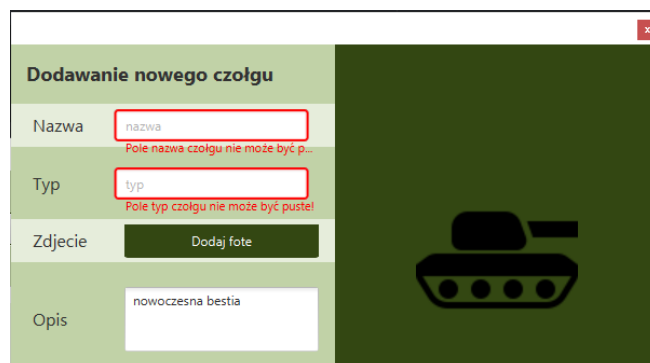
2: muzeum imienia Poniato...



Edytuj

Rysunek 10 - Formularz edycji danych o czołgu

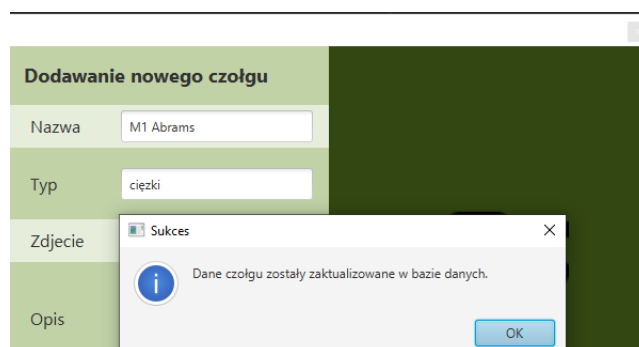
Walidacja danych przy edycji:



The screenshot shows a web form titled "Dodawanie nowego czołgu". It has five input fields: "Nazwa" (containing "nazwa"), "Typ" (containing "typ"), "Zdjecie" (with a "Dodaj foto" button), and "Opis" (containing "nowoczesna bestia"). Red boxes highlight the "Nazwa" and "Typ" fields, with red error messages below them: "Pole nazwa czołgu nie może być p..." and "Pole typ czołgu nie może być puste!". A tank icon is visible on the right side of the form.

Rysunek 11 - Walidacja danych podczas edycji

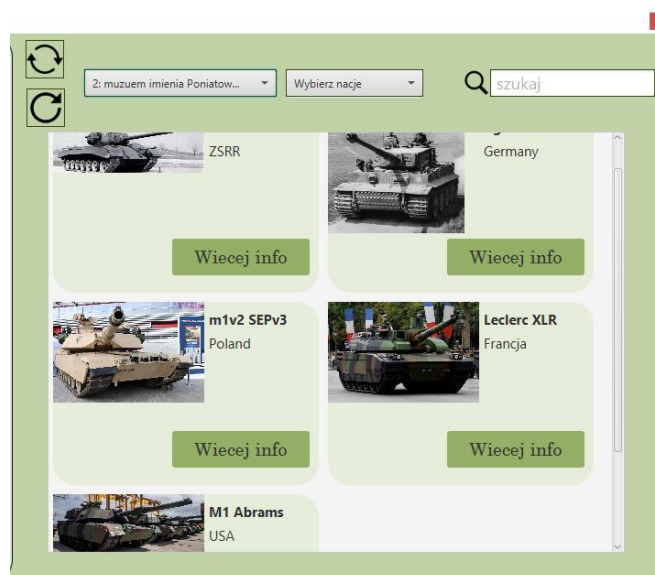
Poprawne dane przy edycji czołgu:



The screenshot shows the same "Dodawanie nowego czołgu" form, but with valid data: "Nazwa" is "M1 Abrams" and "Typ" is "ciężki". A modal dialog box titled "Sukces" is displayed in the foreground, containing an information icon and the text "Dane czołgu zostały zaktualizowane w bazie danych." with an "OK" button.

Rysunek 12 - Poprawne dane podczas edycji

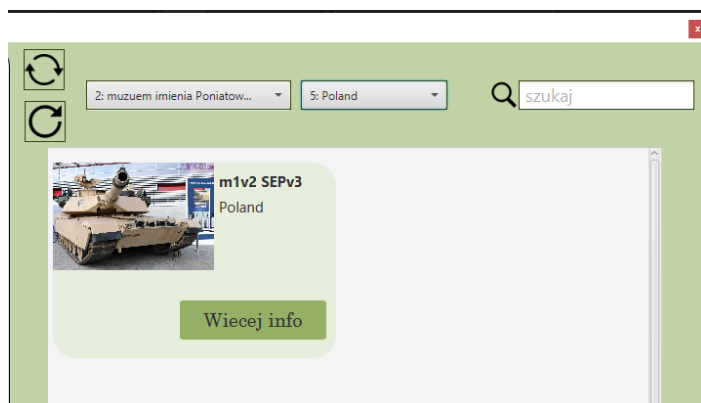
Po wybraniu muzeum wyświetlane są tylko czołgi które się w nim znajdują:



The screenshot shows a museum selection interface. At the top, there are two dropdown menus: "Z muzeum imienia Poniatow..." and "Wybierz nację", followed by a search bar with the text "szukaj". Below these are five cards, each featuring a tank image, its name, and its country of origin, with a "Wiecej info" button underneath. The cards are: "ZSRR" (image of a tank), "Germany" (image of a tank), "m1v2 SEPv3 Poland", "Leclerc XLR Francja", and "M1 Abrams USA".

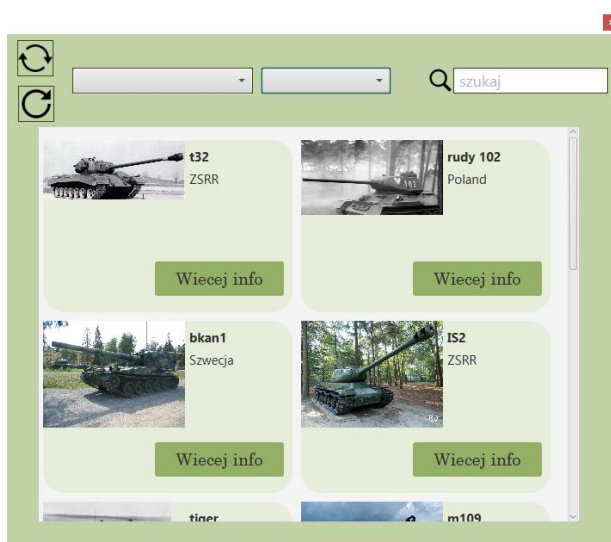
Rysunek 13 - Wybór muzeum

Możemy także filtrować czołgi po nacji:



Rysunek 14 - Filtrowanie czołgów po nacji

Przyciskiem po lewej stronie możemy resetować wcześniej wybrane filtry:



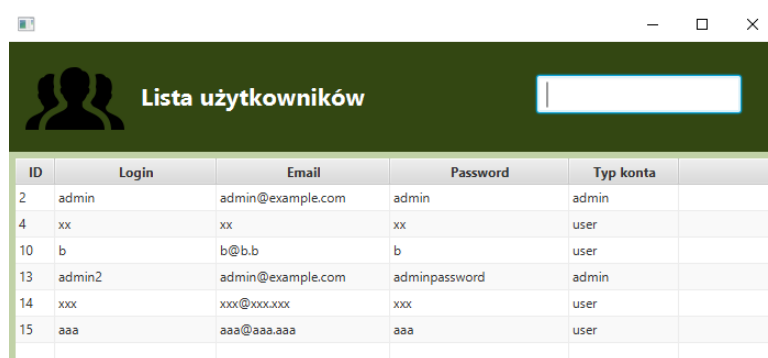
Rysunek 15 - Resetowanie filtrowania

Możemy użyć wyszukiwarki do znajdowania konkretnych czołgów w tym przypadku zawierające literę „l”



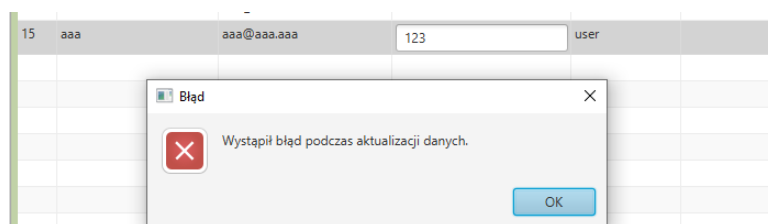
Rysunek 16 - Wyszukiwarka czołgów

Widok wszystkich użytkowników z możliwością ich edycji wraz z walidacją:

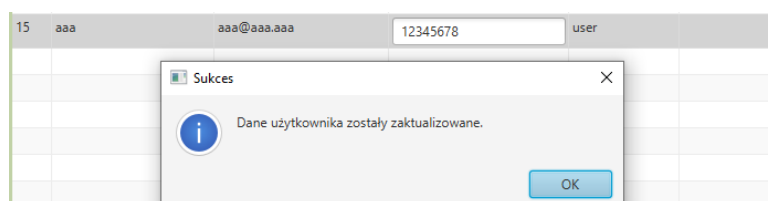


ID	Login	Email	Password	Typ konta
2	admin	admin@example.com	admin	admin
4	xx	xx	xx	user
10	b	b@b.b	b	user
13	admin2	admin@example.com	adminpassword	admin
14	xxx	xxx@xxx.xxx	xxx	user
15	aaa	aaa@aaa.aaa	aaa	user

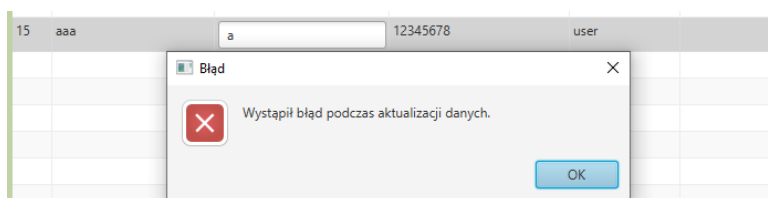
Rysunek 17 - Widok wszystkich użytkowników



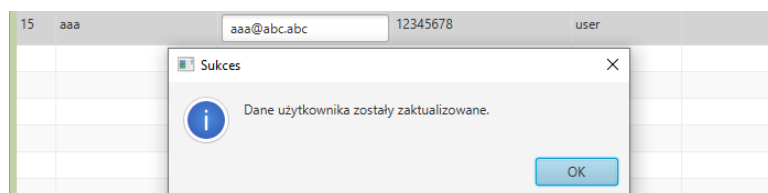
Rysunek 18 - Złe dane podczas próby edycji



Rysunek 19 - Poprawne dane podczas próby edycji

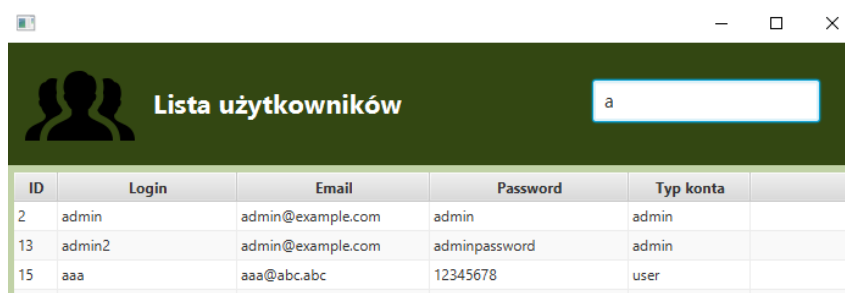


Rysunek 20 - Złe dane podczas próby edycji



Rysunek 21 - Poprawne dane podczas próby edycji

Możemy wyszukiwać użytkowników:



ID	Login	Email	Password	Typ konta
2	admin	admin@example.com	admin	admin
13	admin2	admin@example.com	adminpassword	admin
15	aaa	aaa@abc.abc	12345678	user

Rysunek 22 - Wyszukiwanie użytkowników

Widok wszystkich nacji, który działa na tej samej zasadzie co lista użytkowników, tzn. wyszukiwarka oraz edycja wraz walidacją:



ID	Nazwa nacji	Miejsce produkcji
1	ZSRR	Stalingrad
3	USA	Chigaco
4	England	London
5	Poland	Warsaw
6	Szwecja	Sztokholm
2	Germany	Berlin
7	Francja	Paryż

Rysunek 23 - Lista nacji i wyszukiwanie

Dodawanie nowego czołgu (walidacja):

The screenshot shows a web form titled "Dodawanie nowego czołgu" with the following fields and validation messages:

- Nazwa:** Input field with "nazwa". Error: "Pole nazwa czołgu nie może być p..."
- Typ:** Input field with "typ". Error: "Pole typ czołgu nie może być puste!"
- Zdjecie:** Button "Dodaj fote".
- Opis:** Input field with "opis". Error: "Pole opis czołgu nie może być puste!"
- Ilosc:** Input field with "ilosc". Error: "Pole ilość czołgów nie może być p..."
- Data:** Date picker with "11-11-1999". Error: "Pole data czołgu nie może być pus..."
- Nacja:** Dropdown menu. Error: "Pole nacja czołgu nie może być pu..."
- Amunicja:** Dropdown menu. Error: "Pole amunicja czołgu nie może być..."
- Muzuem:** Dropdown menu. Error: "Pole muzuem czołgu nie może być..."

On the right, there is a tank icon and a large green "Dodaj" button.

Rysunek 24 - Walidacja formularza przy dodawaniu czołgu

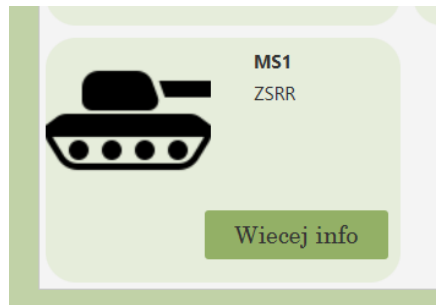
W przypadku podania prawidłowych danych: Kiedy nie wybierzemy zdjęcia, aplikacja ustawi zdjęcie domyślne:

The screenshot shows the same form as in Figure 24, but with valid data entered and a success message dialog displayed:

- Nazwa:** "nazwa"
- Typ:** "typ"
- Zdjecie:** "Dodaj fote" button
- Opis:** (empty)
- Ilosc:** (empty)
- Data:** "11-11-1999"
- Nacja:** "1: ZSRR"
- Amunicja:** "2: apcr"
- Muzuem:** "3: muzuem 3 brygady"

The success dialog box "Sukces" contains the message: "Nowy czołg został dodany do bazy danych." with an "OK" button.

Rysunek 25 - Dodanie czołgu



Rysunek 26 - Domyślne zdjęcie czołgu

Dodawanie nowej nacji wraz z walidacją danych:

 A screenshot of a web form titled 'Dodawanie nowej narodowości'. It has two input fields: 'Nazwa narodowości' with placeholder 'nazwa...' and 'Główne miejsce produkcji' with placeholder 'miejsce produkcji...'. Both fields have red borders and red error messages below them: 'Pole nazwa nacji nie może być puste!' and 'Pole nazwy produkcji nie może być puste!'. A green 'Dodaj' button is at the bottom. The right side of the form has a dark green background with a white UN logo.

Rysunek 27 - Walidacja formularza przy dodawaniu nowej nacji

W przypadku wprowadzenie poprawnych danych:

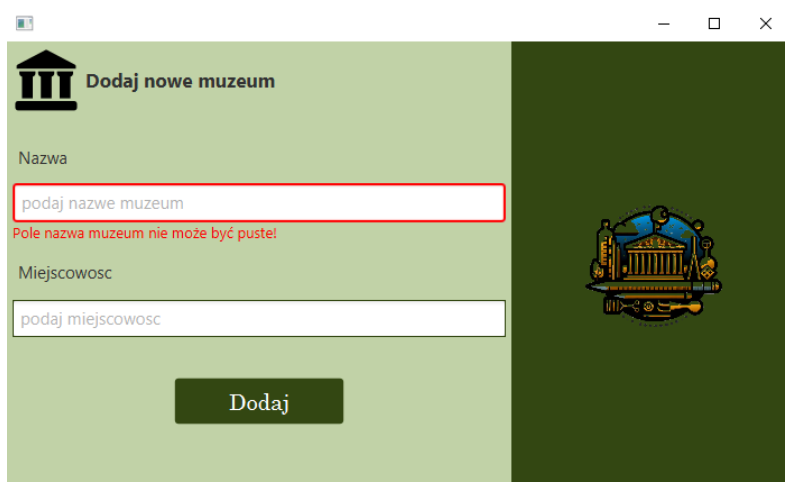
 A screenshot of the same 'Dodawanie nowej narodowości' form. A modal dialog box titled 'Sukces' is overlaid on top. It contains an information icon and the text 'Nowa nacja została dodana do bazy danych.' with an 'OK' button. The form fields behind the dialog are partially visible.

Rysunek 28 - Dodanie nowej nacji

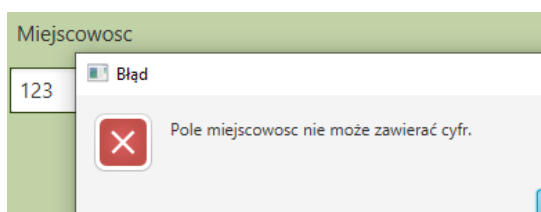
8	Indie	Taj Majal
---	-------	-----------

Rysunek 29 - Dodana wcześniej nacja w liście

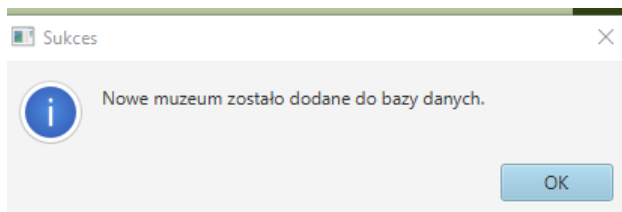
Dodawanie nowego muzeum wraz z walidacją: Miejscowość może być pusta wtedy aplikacja ustawi domyślną wartość na "Warszawę":



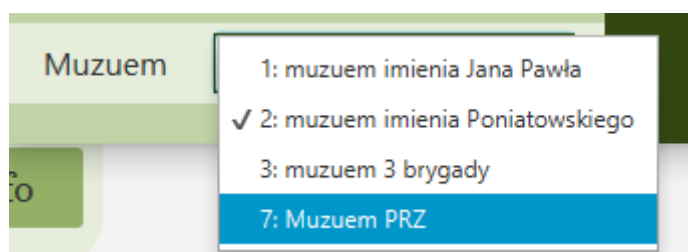
Rysunek 30 - Walidacja nazwy



Rysunek 31 – Komunikat niepoprawnych danych

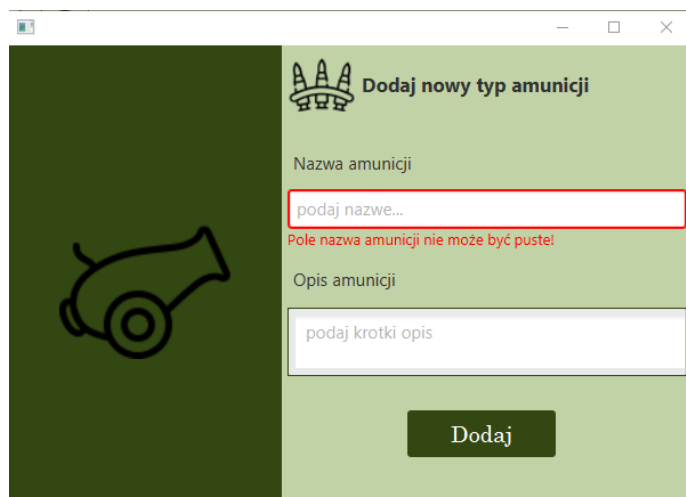


Rysunek 32 - Komunikat dodania muzeum

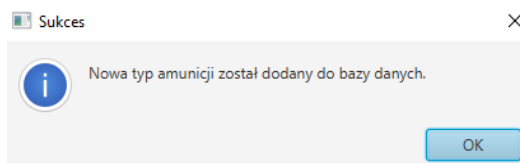


Rysunek 33 - Dowód dodania muzeum

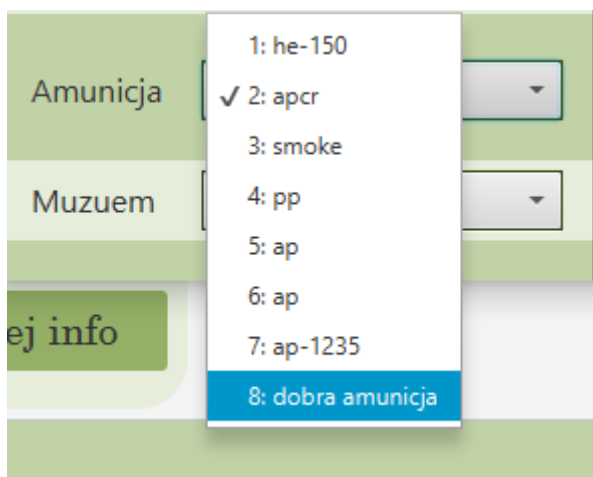
Dodawanie nowego rodzaju amunicji wraz z walidacją: Pole opis może być puste wtedy aplikacja ustawi domyślny opis



Rysunek 34 - Walidacja przy dodawaniu amunicji

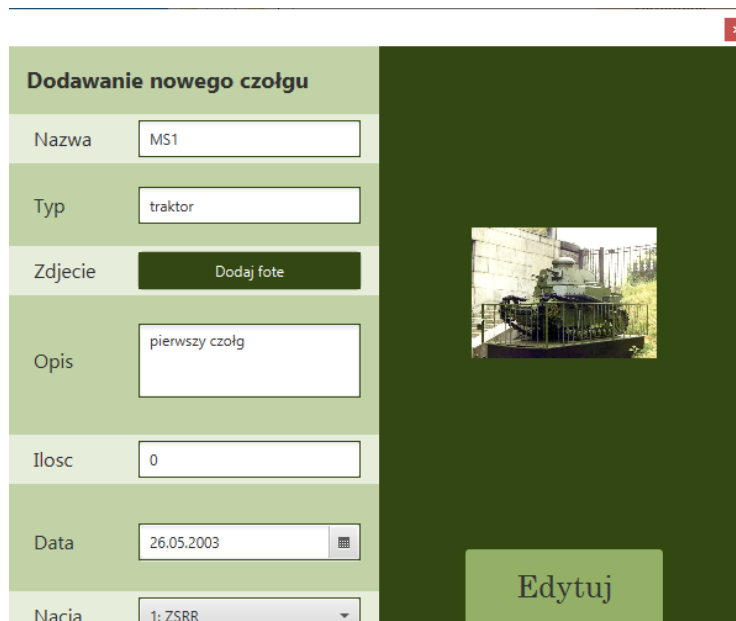


Rysunek 35 - Komunikat przy poprawnym dodaniu amunicji



Rysunek 36 – Dowód

Edycja zdjęcia czołgu:



Dodawanie nowego czołgu


Nazwa

Typ

Zdjecie

Opis

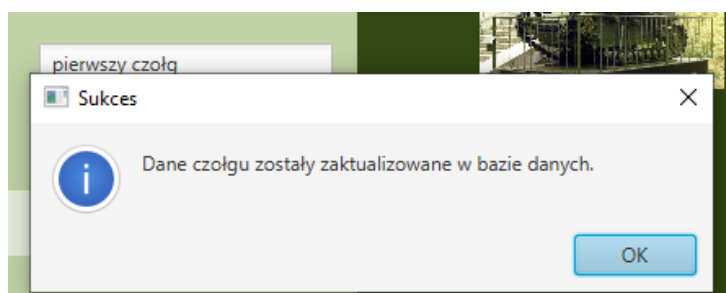
Ilosc

Data 

Nacja

Edytuj

Rysunek 37 - Formularz edycji zdjęcia



Rysunek 38 - Komunikat



Rysunek 39- Nowe zdjęcie czołgu

Funkcje języka proceduralnego:

Funkcje:

- check_no_digits:

```
CREATE OR REPLACE FUNCTION check_no_digits(input_text TEXT)
RETURNS BOOLEAN AS $$
BEGIN RETURN NOT (regexp_match(input_text, '[0-9]') IS NOT NULL);
END;
$$ LANGUAGE plpgsql;
```

Opis: Funkcja sprawdzająca, czy w podanym tekście nie znajdują się żadne cyfry. Argumenty: input_text TEXT - tekst do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE, jeśli w tekście nie ma cyfr, w przeciwnym razie FALSE.

- is_valid_email

```
CREATE OR REPLACE FUNCTION is_valid_email(email TEXT)
RETURNS BOOLEAN AS $$
BEGIN RETURN email ~* '^[A-Za-z0-9. %+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}$';
END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja sprawdzająca poprawność adresu email. Argumenty: email TEXT - adres email do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE, jeśli email jest poprawny, w przeciwnym razie FALSE

- is_valid_password

```
CREATE OR REPLACE FUNCTION is_valid_password(password TEXT)
RETURNS BOOLEAN AS $$ BEGIN RETURN LENGTH(password) >= 8;
END;
$$ LANGUAGE plpgsql;
```

Opis: Funkcja sprawdzająca, czy hasło spełnia określone wymagania (długość, obecność cyfr, małych i wielkich liter oraz znaków specjalnych). Argumenty: password TEXT - hasło do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE jeśli hasło spełnia wymagania, w przeciwnym razie FALSE

Procedury:

- add_ammunition

```
CREATE OR REPLACE PROCEDURE add_ammunition( p_name VARCHAR, p_description VARCHAR )  
LANGUAGE plpgsql AS $$  
  
BEGIN INSERT INTO ammunition(name, description) VALUES (p_name, p_description);  
  
END; $$;
```

Opis: Procedura add_ammunition służy do dodawania nowej amunicji do bazy danych. Procedura przyjmuje dwa argumenty: nazwę amunicji (p_name) oraz jej opis (p_description). Następnie wstawia te dane do tabeli ammunition. Argumenty: p_name (VARCHAR): Nazwa amunicji, która ma być dodana do bazy danych. p_description (VARCHAR): Opis amunicji, który ma być dodany do bazy danych. Działanie: Procedura wykonuje operację INSERT, aby dodać nowy rekord do tabeli ammunition z podanymi wartościami nazwy i opisu.

- add_museum

```
CREATE OR REPLACE PROCEDURE add_museum( p_name VARCHAR, p_localization VARCHAR )  
LANGUAGE plpgsql AS $$  
  
BEGIN INSERT INTO museum(name, localization) VALUES (p_name, p_localization);  
  
END; $$;
```

Opis: Procedura add_museum dodaje nowe muzeum do bazy danych. Przyjmuje dwa argumenty: nazwę muzeum i jego lokalizację, które są następnie wstawiane do tabeli museum. Argumenty: p_name VARCHAR - nazwa muzeum. p_localization VARCHAR - lokalizacja muzeum. Działanie: Procedura wstawia nowy rekord do tabeli museum z podanymi wartościami dla kolumn name i localization.

- add_nation

```
CREATE OR REPLACE PROCEDURE add_nation( p_nation_name VARCHAR, p_prod_place VARCHAR )  
LANGUAGE plpgsql AS $$
```

```
BEGIN INSERT INTO nationality(nation_name, prod_place) VALUES (p_nation_name, p_prod_place);  
END; $$;
```

Opis: Procedura add_nation służy do dodawania nowej nacji do bazy danych. Procedura przyjmuje dwa argumenty: nazwę nacji (p_nation_name) oraz miejsce produkcji (p_prod_place). Następnie wstawia te dane do tabeli nationality. Argumenty: p_nation_name (VARCHAR): Nazwa nacji, która ma być dodana do bazy danych. p_prod_place (VARCHAR): Miejsce produkcji związane z nacją, które ma być dodane do bazy danych. Działanie: Procedura wykonuje operację INSERT, aby dodać nowy rekord do tabeli nationality z podanymi wartościami nazwy nacji i miejsca produkcji.

Triggery:

- check_input_trigger_function

```
CREATE OR REPLACE FUNCTION check_input_trigger_function()  
  
RETURNS TRIGGER AS $$  
  
BEGIN IF NEW.nation_name IS NOT NULL AND NOT check_no_digits(NEW.nation_name::TEXT) THEN  
RAISE EXCEPTION 'Nazwa nacji nie może zawierać cyfr.'; END IF; IF NEW.prod_place IS NOT NULL AND  
NOT check_no_digits(NEW.prod_place::TEXT) THEN RAISE EXCEPTION 'Miejsce produkcji nie może  
zawierać cyfr.';  
  
END IF;  
  
RETURN NEW;  
  
END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja check_input_trigger_function jest funkcją wyzwalającą (trigger function), która służy do walidacji danych wprowadzanych do tabeli nationality. Jej głównym celem jest zapewnienie, że pola nation_name i prod_place nie zawierają cyfr. Jeśli którakolwiek z tych kolumn zawiera cyfry, funkcja zgłasza wyjątek i zapobiega wstawieniu lub aktualizacji danych. Działanie: Funkcja sprawdza, czy kolumna nation_name w nowym rekordzie (NEW) jest różna od NULL i czy nie zawiera cyfr, korzystając z funkcji check_no_digits. Jeśli nation_name zawiera cyfry, funkcja zgłasza wyjątek z komunikatem "Nazwa nacji nie może zawierać cyfr." Podobnie, funkcja sprawdza, czy kolumna prod_place w nowym rekordzie (NEW) jest różna od NULL i czy nie zawiera cyfr, korzystając z funkcji check_no_digits. Jeśli prod_place zawiera cyfry, funkcja zgłasza wyjątek z komunikatem "Miejsce produkcji nie może zawierać cyfr." Jeśli oba pola są prawidłowe, funkcja zwraca nowy rekord (NEW), pozwalając na wstawienie lub aktualizację danych w tabeli nationality.

- set_default_acc_type

```
CREATE OR REPLACE FUNCTION set_default_acc_type()
```

```
RETURNS TRIGGER AS $$  
  
BEGIN NEW.acc_type := 'user';  
  
RETURN NEW;  
  
END;  
  
$$ LANGUAGE plpgsql;
```

Opis: Funkcja `set_default_acc_type` jest funkcją wyzwalającą (trigger function), której celem jest automatyczne ustawienie domyślnej wartości kolumny `acc_type` na 'user' dla nowych rekordów wstawianych do tabeli. Jest używana w kontekście wyzwalacza, który uruchamia się przed operacją INSERT na określonej tabeli. Działanie: Funkcja przypisuje wartość 'user' do kolumny `acc_type` nowego rekordu (NEW). Funkcja zwraca zmodyfikowany nowy rekord (NEW), co pozwala na dalsze przetwarzanie rekordu przez operację INSERT.

- `set_default_description`

```
CREATE OR REPLACE FUNCTION set_default_description()  
  
RETURNS TRIGGER AS $$  
  
BEGIN IF NEW.description IS NULL OR NEW.description = '' THEN NEW.description := 'opis zostanie  
jeszcze dodany';  
  
END IF;  
  
RETURN NEW;  
  
END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja `set_default_description` ma na celu automatyczne ustawianie wartości domyślnej dla kolumny `description` w przypadku, gdy nowo wstawiany rekord nie zawiera wartości dla tej kolumny lub gdy wartość ta jest pusta.

- `set_default_localization`

```
CREATE OR REPLACE FUNCTION set_default_localization()  
  
RETURNS TRIGGER AS $$ BEGIN IF NEW.localization IS NULL OR NEW.localization = '' THEN  
NEW.localization := 'Warszawa';  
  
END IF;  
  
RETURN NEW;
```

```
END; $$ LANGUAGE plpgsql;
```

Opis: Funkcja `set_default_localization` ma na celu automatyczne ustawianie wartości domyślnej dla kolumny `localization` w przypadku, gdy nowo wstawiany rekord nie zawiera wartości dla tej kolumny lub gdy wartość ta jest pusta.

- `set_negative_amount_to_zero`

```
CREATE OR REPLACE FUNCTION set_negative_amount_to_zero()  
RETURNS TRIGGER AS $$ BEGIN IF NEW.amount < 0 THEN NEW.amount := 0;  
END IF;  
RETURN NEW; END; $$ LANGUAGE plpgsql;
```

Do tej funkcji utworzone są triggerzy: przed dodaniem nowego czółgu oraz podczas edycji:

```
CREATE TRIGGER set_negative_amount_to_zero_before_insert BEFORE INSERT ON tanks FOR EACH  
ROW EXECUTE FUNCTION set_negative_amount_to_zero();  
CREATE TRIGGER set_negative_amount_to_zero_before_update BEFORE UPDATE ON tanks FOR EACH  
ROW EXECUTE FUNCTION set_negative_amount_to_zero();
```

Opis: Funkcja `set_negative_amount_to_zero` ma na celu zapobieganie ustawieniu ujemnej wartości dla kolumny `amount` w tabeli `tanks`. Jeśli nowa wartość dla tej kolumny jest ujemna, zostanie automatycznie zmieniona na zero.

- `validate_user_update`

```
CREATE OR REPLACE FUNCTION validate_user_update()  
RETURNS TRIGGER AS $$  
BEGIN IF NEW.user_email IS NOT NULL AND NOT is_valid_email(NEW.user_email) THEN RAISE  
EXCEPTION 'Niepoprawny adres e-mail: %', NEW.user_email;  
END IF; IF NEW.user_pass IS NOT NULL AND NOT is_valid_password(NEW.user_pass) THEN RAISE  
EXCEPTION 'Hasło musi mieć co najmniej 8 znaków';  
END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;
```

Do tej funkcji utworzonym jest trigger, który wywoła funkcję przed dodaniem nowego użytkownika albo przed jego edycją:

```
CREATE TRIGGER check_input_trigger BEFORE INSERT OR UPDATE ON nationality FOR EACH ROW  
EXECUTE FUNCTION check_input_trigger_function();
```

Opis: Funkcja `validate_user_update` ma na celu sprawdzenie poprawności aktualizacji danych użytkownika w tabeli `users`. Sprawdza, czy nowy adres e-mail jest w prawidłowym formacie oraz czy nowe hasło spełnia minimalne wymagania dotyczące długości.

Sekwencje są użytkowane przy inkrementowaniu kluczy głównych.

```
CREATE SEQUENCE IF NOT EXISTS public.ammunition_ammo_id_seq  
  
INCREMENT 1  
  
START 1  
  
MINVALUE 1  
  
MAXVALUE 2147483647  
  
CACHE 1  
  
OWNED BY ammunition.ammo_id;
```

Źródła:

Ikony pobrane ze strony: <https://icons8.com/>

Grafika muzeum utworzona przez: <https://copilot.microsoft.com/>

Dynamiczna siatka z czołgami: <https://www.youtube.com/watch?v=L3PLDAZWU9s>

Drugi materiał dotyczący siatki: <https://www.youtube.com/watch?v=XIAzQ170kzM>

Wzór: <https://www.youtube.com/watch?v=Aliw3lNRzfc>

Dodawanie zdjęć do bazy: <https://www.tutorialspoint.com/how-to-insert-an-image-in-to-mysql-database-using-java-program>

Również dodawanie: <https://stackoverflow.com/questions/30447539/uploading-image-to-a-mysql-database-using-java>