****

**Praca projektowa programowanie obiektowe**

*System do zarządzania bazą danych pojazdów*

Prowadzący: Autor:

mgr inż. Ewa Żesławska *Dawid Mucha*

nr albumu:125146

Kierunek: Informatyka, grupa lab 2

Rzeszów 20XX

Spis treści

[**1.** **Opis założeń projektu** 3](#_Toc157959707)

[**2.** **Specyfikacja wymagań** 4](#_Toc157959708)

[**3.** **Opis struktury projektu** 5](#_Toc157959709)

[**4.** **Harmonogram realizacji projektu** 8](#_Toc157959710)

[**5.** **Prezentacja warstwy użytkowej projektu** 10](#_Toc157959711)

[**6.** **Podsumowanie** 15](#_Toc157959712)

[**7.** **Literatura** 16](#_Toc157959713)

1. **Opis założeń projektu**

W dzisiejszych biurach, gdzie skomplikowane procesy zarządzania pojazdami są nieuniknione, drobne niedociągnięcia mogą znacząco wpływać na efektywność codziennych działań. Takie sytuacje przypominają nam o potrzebie wprowadzenia systemów ułatwiających zarządzanie flotą pojazdów. Analogicznie do organizacji w domu, gdzie odpowiednie dbanie o detale wpływa na komfort życia, w firmie należy zadbać o skuteczne zarządzanie informacjami dotyczącymi pojazdów. Wprowadzenie systemu bazodanowego to krok w kierunku zwiększenia efektywności zarządzania flotą. Dzięki takiemu narzędziu, pracownicy biura stają się bardziej zorganizowani, a dostęp do kompleksowych informacji o każdym pojeździe staje się łatwiejszy niż kiedykolwiek wcześniej. W bazie danych można przechowywać nie tylko podstawowe informacje, takie jak dane techniczne czy daty produkcji.

**Cele i założenia projektu**

Celem projektu było stworzenie efektywnej bazy danych pojazdów, umożliwiającej kompleksowe zarządzanie flotą i usprawnienie procesów związanych z monitorowaniem pojazdów w firmie, co miało przyczynić się do zwiększenia efektywności operacyjnej. Głównym problemem, który został rozwiązany, był brak skutecznej organizacji. Niedostateczne zarządzanie flotą miało negatywny wpływ na efektywność operacyjną, a dowodami na istnienie problemu były m.in. brak aktualnych informacji o pojazdach. Aby skutecznie rozwiązać ten problem, zdecydowano się na stworzenie zaawansowanej bazy danych, przechowującej kompleksowe informacje o każdym pojeździe, oraz opracowanie intuicyjnego interfejsu, umożliwiającego łatwe wprowadzanie, usuwanie danych oraz monitorowanie stanu floty. Projekt został zrealizowany krok po kroku, rozpoczynając od analizy potrzeb firmy i struktury bazy danych. Następnie przystąpiono do implementacji, uwzględniając aspekty zabezpieczeń danych oraz łatwość użytkowania. Kolejnym etapem było zintegrowanie systemu z monitorowaniem technicznym pojazdów. Efektem prac powstała baza danych, umożliwiająca zarządzanie flotą oraz śledzenie pojazdów. W rezultacie uzyskano dostęp do zaktualizowanych danych, co pozwoliło na skuteczniejsze zarządzanie flotą oraz poprawę ogólnej efektywności operacyjnej. Ostatecznym rezultatem projektu jest kompleksowa aplikacja do zarządzania pojazdami, dostosowana do indywidualnych potrzeb firmy, stworzona przy użyciu języka programowania Java, biblioteki Java Swing oraz lokalnego serwera baz danych MySQL.

1. **Specyfikacja wymagań**

**Wymagania funkcjonalne**

* Rejestracja użytkowników.
* Logowanie użytkowników.
* Walidacja formularzy rejestracji i logowania.
* Bezpieczne uwierzytelnianie i autoryzacja użytkowników.
* Pulpit administracyjny do zarządzania flotą pojazdów z bazy danych.
* Dodawanie pojazdów do bazy danych.
* Walidacja dodawanych pojazdów.
* Wybór rodzaju pojazdu spośród dwóch opcji.
* Wyświetlanie pojazdów z bazy danych.
* Usuwanie pojazdów z bazy danych.
* Walidacja usuwanych pojazdów.
* Export bazy danych do pliku csv.
* Import bazy danych z pliku SQL.

**Wymagania niefunkcjonalne**

* Użyteczność aplikacji i dostępność, w tym intuicyjny design.
* Wydajność aplikacji i skalowalność, obejmujące szybkie czasy ładowania.
* Niezawodność aplikacji, działanie na lokalnym serwerze baz danych.
* Konserwowalność, łatwe utrzymanie aplikacji, szybka aktualizacja, sprawna naprawa błędów.
* Integralność danych, poprawność zapisów danych w bazie.
* Bezpieczeństwo, lokalne działanie aplikacji oraz walidacja wprowadzonych danych.
* Interpolacyjność, działanie aplikacji na systemie Windows.

1. **Opis struktury projektu**

W projekcie zaimplementowano hierarchię klas, które reprezentują różne typy pojazdów oraz użytkowników systemu. Poniżej przedstawiono opis struktury projektu:

Klasa Car:

* Reprezentuje ogólne informacje o pojeździe.
* Zawiera pola takie jak nazwa, marka, model, rodzaj paliwa, przebieg, rok produkcji, oraz rodzaj pojazdu (ustawiony na "Osobowy").
* Służy jako podstawowa klasa dla innych rodzajów pojazdów.

Klasa DeletedCars:

* Dziedziczy po klasie Car, co oznacza, że zawiera te same pola informacyjne.
* Dodatkowo posiada pola marka i model, służące do przechowywania informacji o usuniętych pojazdach.

Klasa Motor:

* Dziedziczy po klasie Car, również zawiera te same pola informacyjne.
* Dodatkowo posiada rodzaj pojazdu (ustawiony na "Motocykl").
* Reprezentuje informacje specyficzne dla motocykli.

Klasa User:

* Reprezentuje informacje o użytkowniku systemu.
* Posiada pola login i password, które przechowują dane niezbędne do uwierzytelniania użytkownika.

Projekt tworzono oraz testowano na urządzeniu o wymaganiach systemowych:

* Środowisko programistyczne Javy: Java openjdk-20
* Środowisko programistyczne: IntelliJ IDEA 2023.1 version 17.0.6+10-b829.5
* Urządzenie z systemem operacyjnym Windows 11 ver.23H2

Obraz zawierający tekst, diagram, zrzut ekranu, Równolegle

Opis wygenerowany automatycznie

Rysunek 1. Diagram klas projektowanej aplikacji

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

Rysunek 2.Diagram klas projektowanej aplikacji

1. **Harmonogram realizacji projektu**

Podczas realizacji projektu napotkano pewne wyzwania, głównie związane z dostosowywaniem interfejsu oraz z połączeniem aplikacji z bazą danych. Problemy te zostały skutecznie rozwiązane poprzez intensywną analizę i testowanie aplikacji, co przyczyniło się do uzyskania ostatecznego sukcesu w dostarczeniu efektywnego systemu zarządzania bazą danych pojazdów.

Rysunek 3. Diagram Gantta

Projekt realizowany był z wykorzystaniem systemu kontroli wersji Git, wszystkie pliki źródłowe projektu znajdują się pod adres: <https://github.com/Nostalgiox/Baza_Danych_Pojazdow> i będą dostępne do 04.02.2025. Na rysunku 4 przedstawiono zrzut ekranu pokazujący historię commit’ów.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 4. Lista commit'ów

1. **Prezentacja warstwy użytkowej projektu**

Na rysunku 5 przedstawiono okno logowania do aplikacji. Użytkownik po uruchomieniu aplikacji otrzymuje ekran startowy na którym może:

* Utworzyć własne konto za pomocą którego będzie w stanie zalogować się do aplikacji.
* Zalogować się do aplikacji.
* Wyłączyć aplikację klikając na przycisk „Wyjdź”.

Obraz zawierający tekst, Pojazd lądowy, pojazd, koło

Opis wygenerowany automatycznie

Rysunek 5. Okno logowania

Na rysunku 6 przedstawiono okno rejestracyjne. Użytkownik po kliknięciu przycisku „Zarejestruj się” otrzymuje panel do utworzenia konta na którym jest w stanie:

* Wprowadzić swoje dane do przyszłego logowania.
* Powrócić do poprzedniego okna.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 6. Okno rejestracji

Na rysunku 7 zaprezentowano główne okno aplikacji. Użytkownik po zalogowaniu się do aplikacji ma do wyboru:

* Przyciski „Dodaj pojazd”, „Usuń pojazd”, „Pokaż bazę pojazdów”, „Exportuj do pliku CSV”, „Importuj bazę z pliku SQL”.
* W prawym górnym rogu jest pokazana nazwa za pomocą której użytkownik się logował.
* Na środku okna znajduje się duże zdjęcie pojazdu.

Obraz zawierający tekst, pojazd, zrzut ekranu, samochód

Opis wygenerowany automatycznie

Rysunek 7. Główne okno aplikacji

Na kolejnych rysunkach przedstawiono okna aplikacji dla przycisków:

* Przycisk „Dodaj pojazd” umożliwia:
  + Wprowadzanie danych dla pojazdu.
  + Zaktualizowanie bazy danych, dodając nowy pojazd.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Rysunek 8. Okno dodania pojazdu

* Przycisk „Usuń pojazd” pozwala na:
  + Wprowadzenie danych pojazdu.
  + Zaktualizowanie bazy danych, usuwając wskazany pojazd.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Rysunek 9. Okno usuwania pojazdu

* Przycisk „Pokaż bazę pojazdów” ukazuje:
  + Listę samochodów będących w bazie danych.

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 10. Okno listy pojazdów

1. **Podsumowanie**

W ramach projektu zrealizowano różnorodne prace mające na celu stworzenie kompleksowej aplikacji do zarządzania flotą pojazdów. W pierwszej kolejności skoncentrowano się na implementacji funkcji rejestracji i logowania użytkowników, przy czym szczególną uwagę zwrócono na skuteczną walidację formularzy rejestracji i logowania w celu zapewnienia bezpieczeństwa danych. W dalszych etapach projektu skoncentrowano się na aspektach bezpiecznego uwierzytelniania i autoryzacji użytkowników, co zaowocowało stworzeniem pulpitu administracyjnego umożliwiającego efektywne zarządzanie flotą pojazdów z bazy danych. Dodatkowo, wprowadzono funkcjonalność dodawania pojazdów do bazy danych, implementując równocześnie mechanizmy walidacji dodawanych informacji oraz umożliwiając wybór rodzaju pojazdu spośród dwóch dostępnych opcji. Aplikacja umożliwia także wygodne przeglądanie pojazdów z bazy danych oraz skuteczne usuwanie ich, z odpowiednią walidacją procesu usuwania. Ponadto, wprowadzono możliwość eksportu bazy danych do pliku csv oraz importu danych z pliku SQL, co dodatkowo poszerza zakres funkcjonalności aplikacji.

Należy również wspomnieć o planach dalszego rozwoju projektu. Potencjalne obszary do dalszych prac mogą obejmować implementację dodatkowych funkcji, optymalizację wydajności systemu, rozszerzenie interfejsu użytkownika oraz dostosowanie aplikacji do ewentualnych zmian w wymaganiach lub nowych funkcjonalnościach oczekiwanych przez użytkowników.

1. **Literatura**
2. <https://www.rawpixel.com>
3. <https://docs.oracle.com/javase%2Ftutorial%2Fuiswing%2F%2F/layout/visual.html>
4. <https://github.com/JFormDesigner/FlatLaf>