****

**Programowanie obiektowe**

*Baza danych pojazdów*

Prowadzący: Autor:

mgr inż. Ewa Żesławska *Arkadiusz Haznar*

nr albumu: 125120

Kierunek: Informatyka, grupa lab 2

Rzeszów, 04.02.2024

Spis treści

[**1.** **Opis założeń projektu** 3](#_Toc157874634)

[**2.** **Specyfikacja wymagań** 3](#_Toc157874635)

[**3.** **Diagram przypadków użycia** 4](#_Toc157874636)

[**4.** **Harmonogram realizacji projektu (diagram Gantta)** 5](#_Toc157874637)

[**5.** **Opis techniczny projektu** 5](#_Toc157874638)

[**6.** **Prezentacja warstwy użytkowej projektu** 6](#_Toc157874639)

[**7.** **System kontroli wersji** 10](#_Toc157874640)

[**8.** **Literatura** 11](#_Toc157874641)

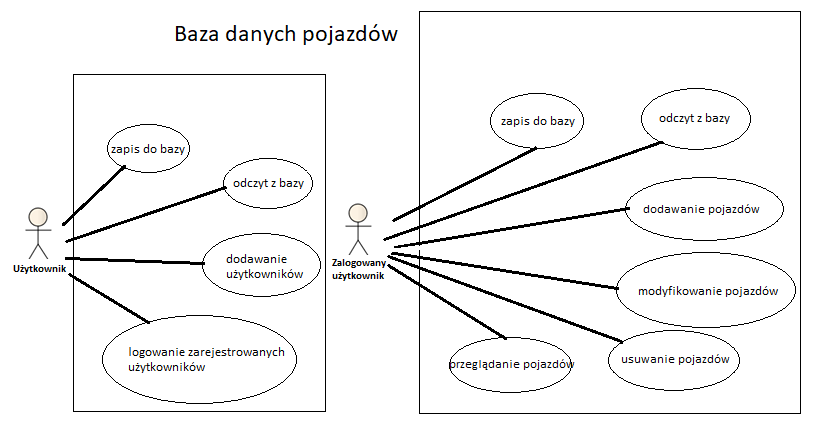
1. **Opis założeń projektu**

Użytkownik może rejestrować (dodawać) nowych użytkowników do bazy. Każdy użytkownik reprezentowany jest przez następujące dane: imię, nazwisko, email, login, hasło. Po rejestracji użytkownik posiadający konto może się zalogować do głównego menu aplikacji przy pomocy loginu i hasła podanego przy rejestracji. Zalogowany użytkownik może dodawać do bazy nowe pojazdy. Każdy pojazd jest reprezentowany przez następujące dane: marka, model, przebieg i kolor. Użytkownik może dodawać dane dotyczące różnych pojazdów do bazy (najpierw musi uzupełnić wszystkie wymagane pola, później nacisnąć odpowiedni przycisk). Może również usuwać dane z bazy na przykład w przypadku błędnego dodania, chęci usunięcia danego rekordu (najpierw zaznacza rekord, który chce usunąć w tabeli i naciska odpowiedni przycisk). Ma możliwość modyfikowania rekordów, które znajdują się w bazie (uzupełnia wymagane pola, następnie zaznacza rekord, który chce zmodyfikować w tabeli i naciska przycisk). W każdej chwili może zobaczyć dane, które są zapisane w bazie. Ma możliwość wyczyszczenia danych w polach do wprowadzania danych i również w tabeli wyświetlającej dane w menu. W dowolnym momencie może wyświetlić dane z bazy w tabeli i dowolnie je dodawać, usuwać i modyfikować. Każdy pojazd zapisuje się w tej samej bazie w tabeli pojazdy, co upraszcza odfiltrowanie pojazdów ze względu na poszczególne pola, na przykład porównania przebiegów poszczególnych pojazdów. Baza danych może posłużyć jako bardzo ważne dane w celu identyfikacji różnych pojazdów. Zaleca się język Java.

1. **Specyfikacja wymagań**
   1. **Wymagania funkcjonalne**

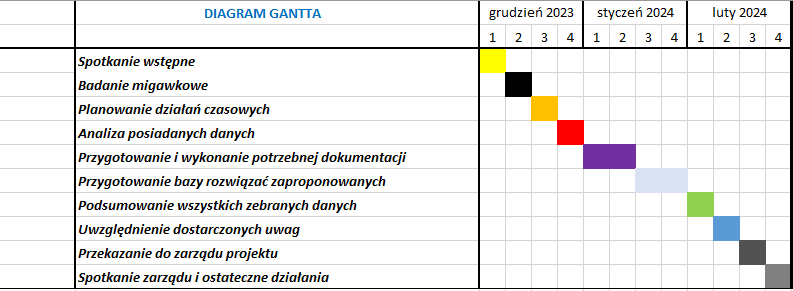
* Użytkownik może dodawać nowych użytkowników do bazy.
* Zarejestrowani użytkownicy mogą logować się do głównego menu aplikacji.
* Zalogowany użytkownik może dodawać pojazdy do bazy.
* Zalogowany użytkownik może usuwać pojazdy z bazy.
* Zalogowany użytkownik może modyfikować pojazdy z bazy.
* Zalogowany użytkownik może wyczyścić zawartość pól do wprowadzania danych jak i tabeli wyświetlającej dane w menu.
* Zalogowany użytkownik ma dostęp do danych wprowadzanych jak i do zapisanych w bazie.
* Zalogowany użytkownik może dodawać, usuwać i edytować dowolną liczbę pojazdów (przeglądając ich zawartość).
* Zalogowany użytkownik posiada uprawnienia do przeglądania wszystkich danych w bazie pojazdów jak i usuwania, modyfikacji dodawanych danych, zawartość bazy nie kasuje się po wyjściu z aplikacji i ponownym otworzeniu aplikacji w celu posiadania kompletnej bazy bez utraty danych.
* Nie można wprowadzać danych, które są niezgodne ze wzorcem.
  1. **Wymagania niefunkcjonalne**
* Możliwość rejestracji (dodawania) nowych użytkowników do bazy i późniejszego logowania do aplikacji zarejestrowanych użytkowników (wszystko z poziomu użytkownika).
* Możliwość dodawania rekordów do bazy, oraz usuwania, edycji i wyświetlania rekordów z bazy (wszystko z poziomu zalogowanego użytkownika).
* Aplikacja jest przyjazna dla klienta i jego rodziny, oraz jest bardzo prosta w użyciu.
* Aplikacja działa na systemach, które posiadają środowisko programistyczne, np. IntelliJ IDEA.
* Aplikacja tworzona jest w języku Java.
* Aplikacja zapisuje i odczytuje dane do/z bazy danych.

1. **Diagram przypadków użycia**



Rysunek 1. Diagram przypadków użycia

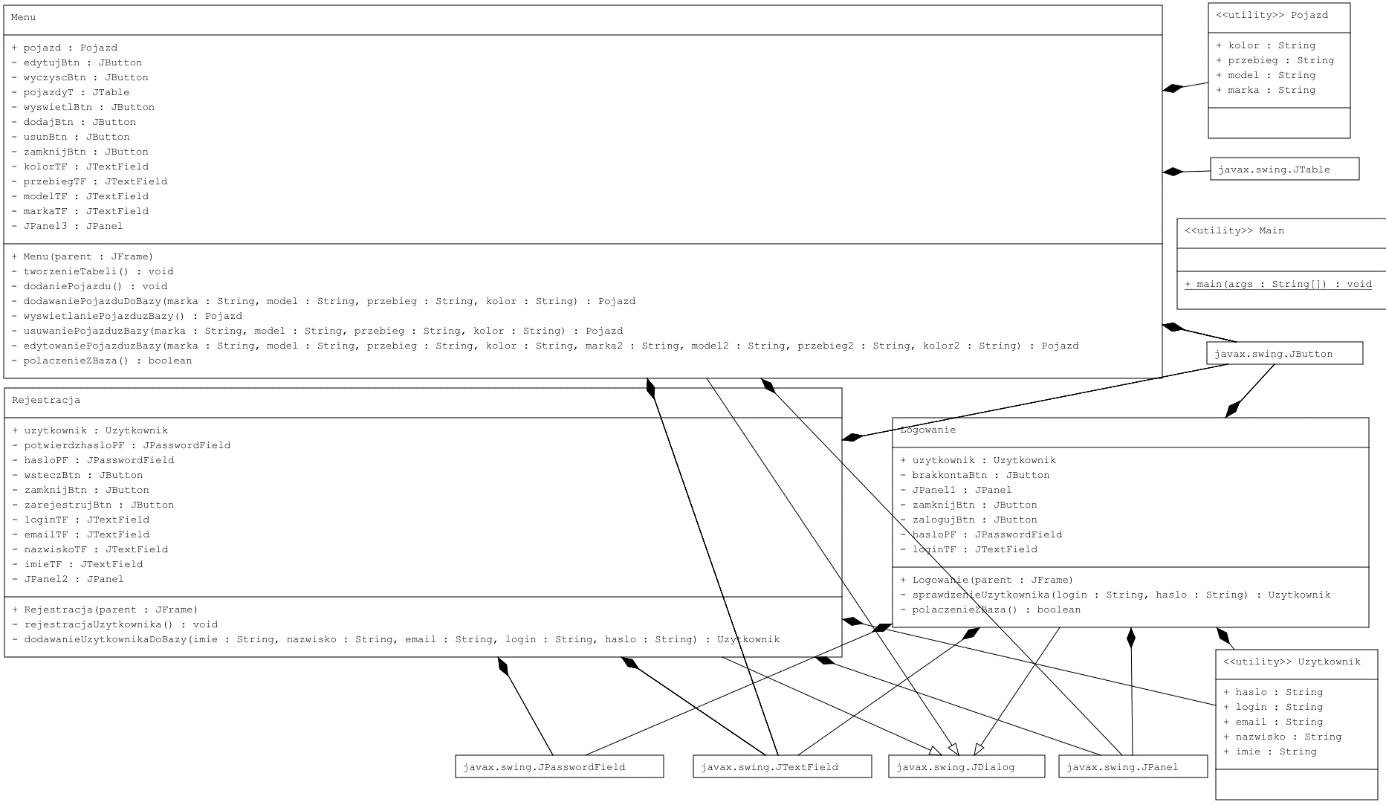
1. **Harmonogram realizacji projektu (diagram Gantta)**



Rysunek 2. Diagram Gantta

1. **Opis techniczny projektu**

* Środowisko programistyczne Javy: Java JDK Kit 7u79
* Dołączona biblioteka graficzna: Swing.iml
* Dołączony sterownik do połączenia się z bazą danych: mysql-connector-j-8.2.0.jar
* Ze względów na liczbę szczegółów aplikacja była projektowana na urządzenia typu laptop ‘Lenovo G570’.



Rysunek 3. Diagram klas projektowanej aplikacji

1. **Prezentacja warstwy użytkowej projektu**

Na rysunku 4 przedstawiono pierwsze okno aplikacji. Użytkownikowi po uruchomieniu aplikacji ukazuje się ekran startowy na którym może:

* Zalogować się do aplikacji (jeśli posiada konto, w tym celu wprowadza odpowiednie dane i naciska przycisk „Zaloguj”, aplikacja przejdzie automatycznie do głównego menu pojazdów, jeśli wprowadzone dane będą prawidłowe (w przeciwnym wypadku wyświetli odpowiedni komunikat)).
* Zarejestrować się do aplikacji (naciskając przycisk „Nie mam konta!”), następuje przeniesienie do drugiego okna aplikacji.
* Zamknąć aplikację (naciskając przycisk „Zamknij”).



Rysunek 4. Okno logowania do aplikacji

Na rysunku 5 przedstawiono drugie okno aplikacji. Użytkownikowi po naciśnięciu przycisku „Nie mam konta” ukazuje się ekran rejestracji na którym może:

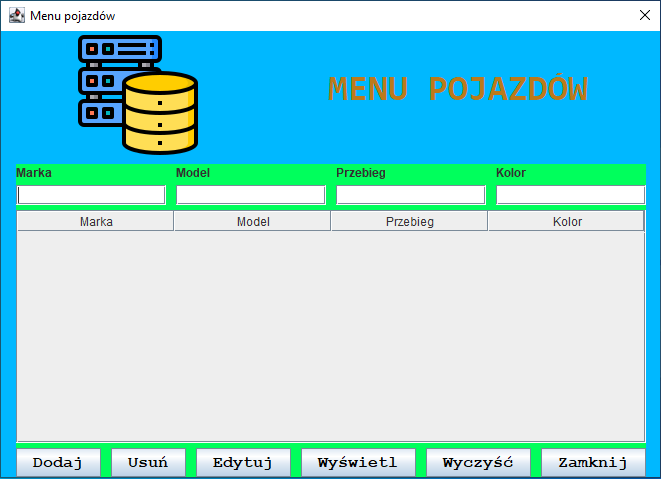
* Założyć sobie konto do aplikacji (należy podać wszystkie potrzebne dane i zatwierdzić przyciskiem „Zarejestruj się”, aplikacja przejdzie automatycznie do okna logowania, jeśli wprowadzone dane będą prawidłowe (w przeciwnym wypadku wyświetli odpowiedni komunikat)).
* Powrócić do ekranu logowania (pierwszego okna) aplikacji (naciskając przycisk „Wstecz”).
* Zamknąć aplikację (naciskając przycisk „Zamknij”).



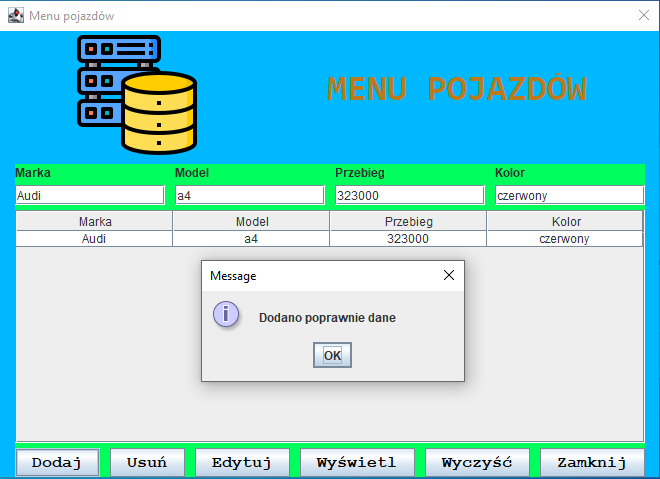
Rysunek 5. Okno rejestracji do aplikacji

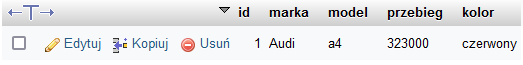
Na rysunku 6 przedstawiono główne menu aplikacji. Użytkownikowi po poprawnym zalogowaniu ukazuje się menu pojazdów na którym może:

* Dodać dowolny pojazd (należy podać wszystkie wymagane dane, a następnie nacisnąć przycisk „Dodaj”, jeśli wprowadzone dane są prawidłowe pojazd wyświetli się w tabeli, doda się do bazy i otrzymamy stosowny komunikat).
* Usunąć dowolny pojazd (należy zaznaczyć rekord, który chcemy usunąć w tabeli i nacisnąć przycisk „Usuń”, jeśli pojazd zostanie usunięty z tabeli i bazy otrzymamy stosowny komunikat).
* Edytować dowolny pojazd (należy podać wszystkie wymagane dane i wybrać z tabeli rekord, który chcemy zmodyfikować, a następnie nacisnąć przycisk „Edytuj”, jeśli wprowadzone dane są prawidłowe pojazd zmieni się w tabeli, zmodyfikuje się w bazie i otrzymamy stosowny komunikat).
* Wyświetlić dane z bazy (należy nacisnąć przycisk „Wyświetl”, dane zostaną pobrane z bazy i pojawią się w tabeli).
* Wyczyścić pola i tabelę (należy nacisnąć przycisk „Wyczyść”, zostaną wyczyszczone pola do wprowadzania danych i tabela wyświetlająca).
* Zamknąć aplikację (naciskając przycisk „Zamknij”).

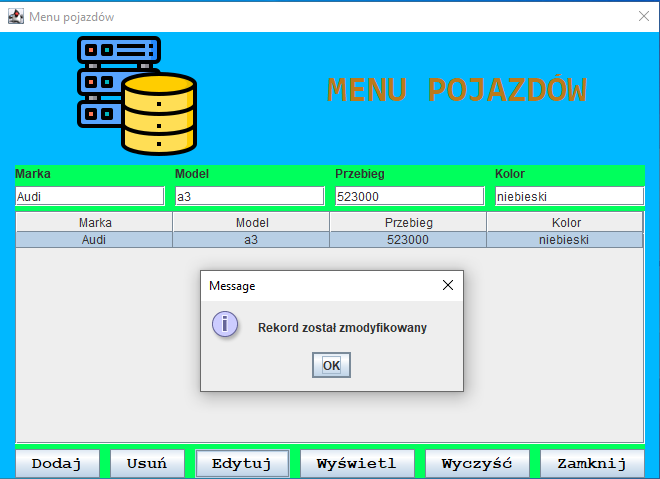


Rysunek 6. Główne okno aplikacji





Rysunek 7. Przykład dodania pojazdu do bazy

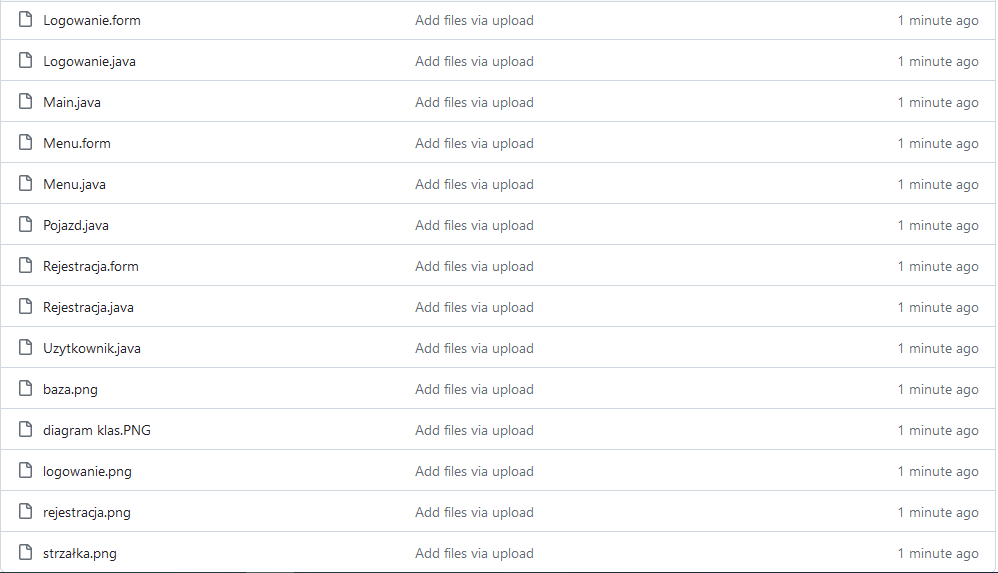




Rysunek 8. Przykład modyfikacji pojazdu z bazy

1. **System kontroli wersji**

Projekt realizowany był z wykorzystaniem systemu kontroli wersji Github, wszystkie pliki źródłowe projektu znajdują się pod adres: **https://github.com/arek123456/Programowanie-obiektowe2.git**. Na rysunku 9 przedstawiono zrzut ekranu pokazujący zawartość folderu projekt.



Rysunek 9. Zawartość folderu projekt

1. **Literatura**
2. <https://www.samouczekprogramisty.pl>
3. <https://javappa.com/kurs-java>
4. <https://javastart.pl/baza-wiedzy/java-podstawy-jezyka>
5. <https://docs.oracle.com/javase/tutorial/uiswing/>
6. <https://javastart.pl/baza-wiedzy/java-ee/jdbc-podstawy-pracy-z-baza-danych>
7. <https://developeronthego.pl/java-jdbc-polaczenie-z-baza-danych/>